ਕੰਪਿਊਟਰ ਸਾਇੰਸ

ਗਿਆਰਵੀਂ ਸ਼੍ਰੇਣੀ ਲਈ

ਇਹ ਪੁਸਤਕ ਪੰਜਾਬ ਸਰਕਾਰ ਦੁਆਰਾ ਮੁਫ਼ਤ
ਦਿੱਤੀ ਜਾਣੀ ਹੈ ਅਤੇ ਵਿਕਰੀ ਲਈ ਨਹੀਂ ਹੈ।

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ

ਸਾਹਿਬਜ਼ਾਦਾ ਅਜੀਤ ਸਿੰਘ ਨਗਰ

ਪਹਿਲਾ ਐਡੀਸ਼ਨ 2021-22

ਰਿਵਾਇਜ਼ਡ ਐਡੀਸ਼ਨ 2025-26 ..... 1,57,606 ਕਾਪੀਆਂ

ਚਿਤਾਵਨੀ

1. ਕੋਈ ਵੀ ਏਜੰਸੀ-ਹੋਲਡਰ ਵਾਧੂ ਪੈਸੇ ਵਸੂਲਣ ਦੇ ਮੰਤਵ ਨਾਲ਼ ਪਾਠ-ਪੁਸਤਕਾਂ 'ਤੇ ਜਿਲਦ-ਸਾਜ਼ੀ ਨਹੀਂ ਕਰ ਸਕਦਾ। (ਏਜੰਸੀ-ਹੋਲਡਰਾਂ ਨਾਲ਼ ਹੋਏ ਸਮਝੌਤੇ ਦੀ ਧਾਰਾ ਨੰ. 7 ਅਨੁਸਾਰ)
2. ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਦੁਆਰਾ ਛਪਾਈਆਂ ਅਤੇ ਪ੍ਰਕਾਸ਼ਿਤ ਪਾਠ-ਪੁਸਤਕਾਂ ਦੇ ਜਾਲ਼ੀ/ਨਕਲੀ ਪ੍ਰਕਾਸ਼ਨਾਂ (ਪਾਠ-ਪੁਸਤਕਾਂ) ਦੀ ਛਪਾਈ, ਸਟਾਕ ਕਰਨਾ, ਜਮ੍ਹਾਖ਼ੋਰੀ ਜਾਂ ਵਿਕਰੀ ਆਦਿ ਕਰਨਾ ਭਾਰਤੀ ਦੰਡ-ਪ੍ਰਣਾਲੀ ਦੇ ਅੰਤਰਗਤ ਫ਼ੌਜਦਾਰੀ ਜੁਰਮ ਹੈ।

   (ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਦੀਆਂ ਪਾਠ-ਪੁਸਤਕਾਂ ਬੋਰਡ ਦੇ 'ਵਾਟਰ ਮਾਰਕ' ਵਾਲ਼ੇ ਕਾਗ਼ਜ਼ ਉੱਪਰ ਹੀ ਛਪਵਾਈਆਂ ਜਾਂਦੀਆਂ ਹਨ।)

ਇਹ ਪੁਸਤਕ ਵਿਕਰੀ ਲਈ ਨਹੀਂ ਹੈ।

ਸਕੱਤਰ, ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ, ਵਿੱਦਿਆ ਭਵਨ, ਫੇਜ਼-8, ਸਾਹਿਬਜ਼ਾਦਾ ਅਜੀਤ ਸਿੰਘ ਨਗਰ-160062 ਰਾਹੀਂ ਪ੍ਰਕਾਸ਼ਿਤ ਅਤੇ ਮੈਸ. ਪਾਇਨੀਅਰ ਪ੍ਰਿੰਟਰਜ਼, ਆਗਰਾ ਦੁਆਰਾ ਛਾਪੀ ਗਈ।

ਮੁੱਖ ਬੰਧ

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਹਮੇਸ਼ਾ ਹੀ ਵਿਦਿਆਰਥੀਆਂ ਦੀ ਵੱਖ-ਵੱਖ ਸਿੱਖਣ ਦੀਆਂ ਲੋੜਾਂ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਉੱਚ-ਗੁਣਵੱਤਾ ਵਾਲੀਆਂ ਪਾਠ-ਪੁਸਤਕਾਂ ਤਿਆਰ ਕਰਨ ਲਈ ਹਮੇਸ਼ਾ ਵਚਨਬੱਧ ਰਿਹਾ ਹੈ। ਸਾਲ 2005 ਵਿੱਚ ਸ਼ੁਰੂਆਤੀ 6ਵੀਂ ਤੋਂ 12ਵੀਂ ਤੱਕ ਦੇ ਵਿਦਿਆਰਥੀਆਂ ਲਈ ਲਾਜ਼ਮੀ ਵਿਸ਼ੇ ਵਜੋਂ ਲਾਗੂ ਕੀਤਾ ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਵਿਸ਼ਾ, ਪੰਜਾਬੀ ਦੇ ਵਿਦਿਆਰਥੀਆਂ ਵਿੱਚ ਤਕਨੀਕੀ ਜਾਣਕਾਰੀ ਅਤੇ ਗਣਨਾਤਮਕ ਸੋਚ ਦੇ ਵਿਕਾਸ ਵਿੱਚ ਮਹੱਤਵਪੂਰਨ ਭੂਮਿਕਾ ਨਿਭਾ ਰਿਹਾ ਹੈ।

ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਸ਼ੁਰੂਆਤੀ ਗਿਆਰਵੀਂ ਦੀ ਅਪਡੇਟ ਕੀਤੀ ਗਈ ਇਸ ਪੁਸਤਕ ਦਾ ਸੰਪਾਦਨ, ਬੋਰਡ ਦੁਆਰਾ ਜਾਰੀ ਯਤਨਾਂ ਅਤੇ ਐਕਟੀਵਿਟੀਜ਼ ਉੱਪਰ ਅਧਾਰਿਤ ਲਰਨਿੰਗ ਰਿਸੋਰਸਾਂ ਦਾ ਹਿੱਸਾ ਹੈ ਤਾਂ ਜੋ ਵਿਦਿਆਰਥੀਆਂ ਨੂੰ ਉਹ ਗਿਆਨ ਅਤੇ ਹੁਨਰ ਪ੍ਰਦਾਨ ਕੀਤੇ ਜਾ ਸਕਣ ਜੋ ਆਧੁਨਿਕ ਡਿਜ਼ੀਟਲ ਦੁਨੀਆਂ ਵਿੱਚ ਸਫਲਤਾ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਜ਼ਰੂਰੀ ਹਨ।

ਇਸ ਸੰਪਾਦਨ ਵਿੱਚ ਅਸੀਂ ਕਈ ਨਵੇ ਟੋਪਿਕ ਸ਼ਾਮਲ ਕੀਤੇ ਹਨ ਤਾਂ ਜੋ ਵਿਦਿਆਰਥੀਆਂ ਨੂੰ ਆਧੁਨਿਕ ਤਕਨੀਕਾਂ ਅਤੇ ਇਸਦੇ ਐਪਲੀਕੇਸ਼ਨ ਖੇਤਰਾਂ ਦੀ ਪੂਰੀ ਸਮਝ ਮਿਲ ਸਕੇ। ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਦੇ ਮੂਲ ਸਿਧਾਂਤਾਂ ਨੂੰ ਕਾਇਮ ਰੱਖਦੇ ਹੋਏ, ਇਹ ਪਾਠ-ਪੁਸਤਕ ਲਿਖਤੀ ਅਤੇ ਪ੍ਰਯੋਗੀ ਅਭਿਆਸਾਂ ਦੇ ਗਤੀਸ਼ੀਲ ਮਿਸ਼ਰਣ ਦੀ ਪੇਸ਼ਕਸ਼ ਕਰਦੀ ਹੈ, ਜੋ ਹੱਥੀ ਸਿਖਲਾਈ ਅਤੇ ਨਵੀਨਤਾਵਾਂ ਨੂੰ ਉਤਸਾਹਿਤ ਕਰਦੀ ਹੈ।

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਇਸ ਪੁਸਤਕ ਨੂੰ ਤਿਆਰ ਕਰਨ ਵਿੱਚ ਯੋਗਦਾਨ ਪਾਉਣ ਵਾਲੇ ਲੇਖਕਾਂ, ਅਨੁਵਾਦਕਾਂ ਅਤੇ ਸੋਧਕਾਂ ਵੱਲੋਂ ਕੀਤੇ ਗਏ ਸੁਹਿਰਦ ਯਤਨਾਂ ਦੀ ਸ਼ਲਾਘਾ ਕਰਦਾ ਹੈ। ਇਸ ਪੁਸਤਕ ਨੂੰ ਹੋਰ ਵਧੀਆਂ ਬਣਾਉਣ ਲਈ ਪ੍ਰਾਪਤ ਹੋਣ ਵਾਲੇ ਉਸਾਰੂ ਸੁਝਾਵਾਂ ਅਤੇ ਟਿੱਪਣੀਆਂ ਦਾ ਬੋਰਡ ਵੱਲੋਂ ਸਵਾਗਤ ਕੀਤਾ ਜਾਵੇਗਾ।

ਚੇਅਰਮੈਨ

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ

‘ਸਮਾਜਿਕ ਨਿਆਂ, ਅਧਿਕਾਰਤਾ ਅਤੇ ਘੱਟ ਗਿਣਤੀ ਵਿਭਾਗ’ ਪੰਜਾਬ।

ਪਾਠ-ਪੁਸਤਕ ਰਚਨਾ ਕਮੇਟੀ

ਲੇਖਕ ਅਤੇ ਅਨੁਵਾਦਕ :

ਸ੍ਰੀ ਵਿਕਾਸ ਕਾਂਸਲ,
ਕੰਪਿਊਟਰ ਫੈਕਲਟੀ
ਸ਼ਹੀਦ ਊਧਮ ਸਿੰਘ ਸ.ਸ.ਸ.ਸ. ਸਕੂਲ (ਕੰਨਿਆ), ਸੁਨਾਮ ਊਧਮ ਸਿੰਘ ਵਾਲਾ, ਸੰਗਰੂਰ (ਪੰਜਾਬ)।

ਸ੍ਰੀ ਸੁਖਵਿੰਦਰ ਸਿੰਘ,
ਕੰਪਿਊਟਰ ਫੈਕਲਟੀ
ਸ਼ਹੀਦ ਊਧਮ ਸਿੰਘ ਸ.ਸ.ਸ.ਸ. ਸਕੂਲ (ਕੰਨਿਆ), ਸੁਨਾਮ ਊਧਮ ਸਿੰਘ ਵਾਲਾ, ਸੰਗਰੂਰ (ਪੰਜਾਬ)।

ਸ੍ਰੀ ਅਰਵਿੰਦਰ ਸਿੰਘ,
ਕੰਪਿਊਟਰ ਫੈਕਲਟੀ
ਸਰਕਾਰੀ ਸੀਨੀਅਰ ਸੈਕੰਡਰੀ ਸਕੂਲ, ਸੁਨੇਤ, ਲੁਧਿਆਣਾ (ਪੰਜਾਬ)।

ਸ੍ਰੀ ਗਗਨਦੀਪ ਸਿੰਘ,
ਕੰਪਿਊਟਰ ਫੈਕਲਟੀ
ਸਕੂਲ ਆਫ ਐਮਿਨੈਂਸ, ਫੇਜ਼ 3 ਬੀ-1, ਐੱਸ. ਏ. ਐੱਸ. ਨਗਰ (ਪੰਜਾਬ)।

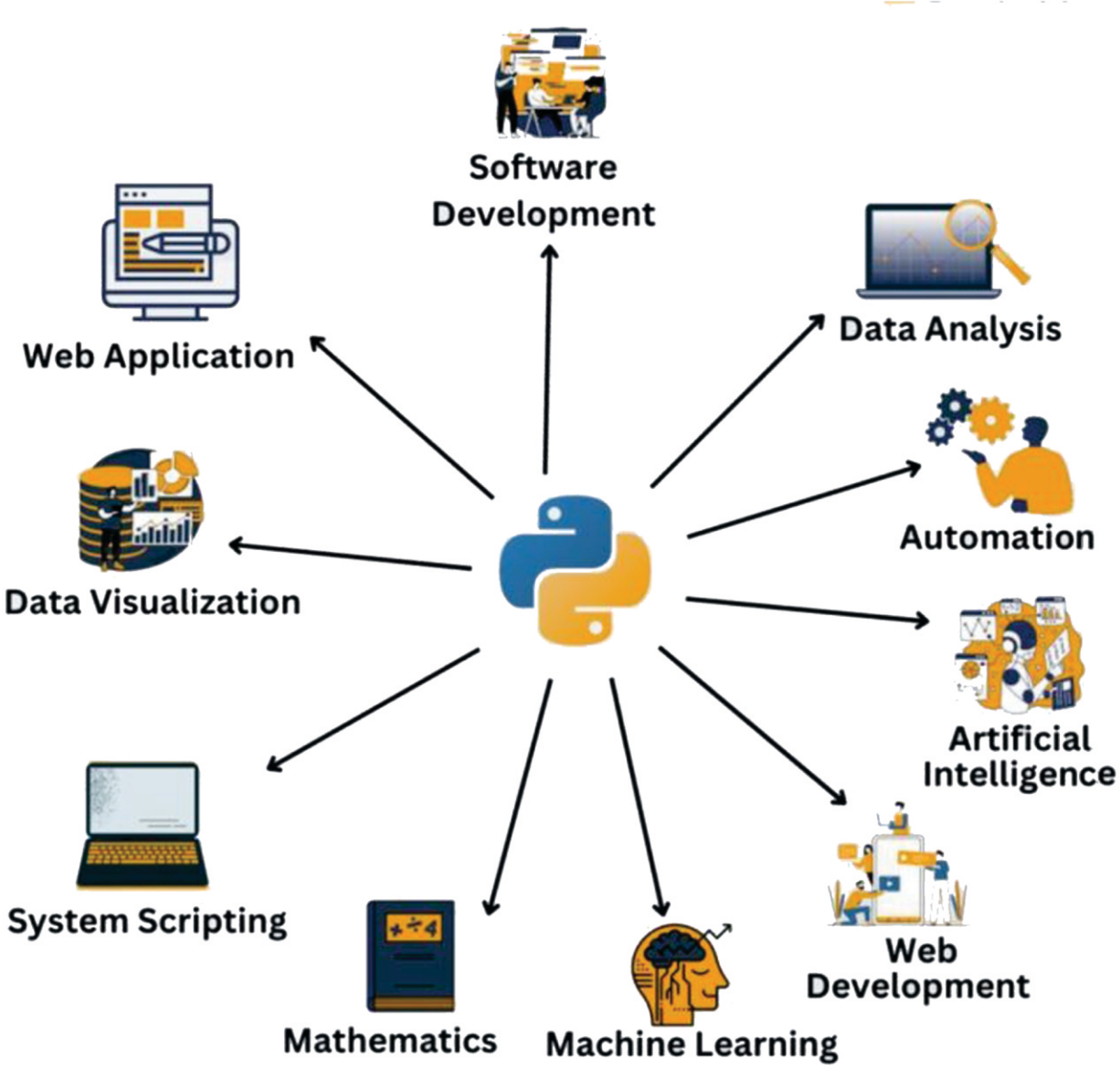
ਸ੍ਰੀਮਤੀ ਮੀਨੂੰ,
ਕੰਪਿਊਟਰ ਫੈਕਲਟੀ
ਸਰਕਾਰੀ ਹਾਈ ਸਕੂਲ, ਗੜਾਂਗਾ, ਐੱਸ. ਏ. ਐੱਸ ਨਗਰ (ਪੰਜਾਬ)।

ਕੋਆਰਡੀਨੇਟਰ :

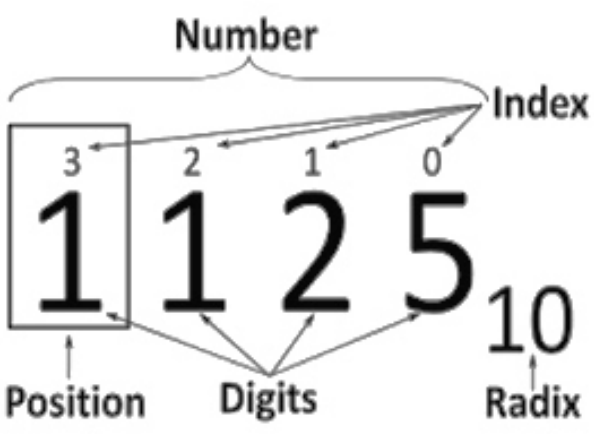
ਸ੍ਰੀ ਮਨਵਿੰਦਰ ਸਿੰਘ, ਵਿਸ਼ਾ ਮਾਹਿਰ (ਕੰਪਿਊਟਰ),
ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ, ਐੱਸ. ਏ. ਐੱਸ. ਨਗਰ (ਪੰਜਾਬ)।

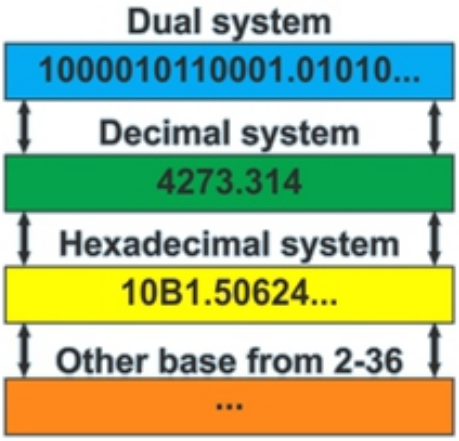
ਇੰਡੈਕਸ

APPLICATIONS OF PYTHON



ਨੰਬਰ ਸਿਸਟਮ





ਬਹੁਤ ਸਮਾਂ ਪਹਿਲਾਂ, ਜਦੋਂ ਸੰਖਿਆਵਾਂ ਦੀ ਖੋਜ ਨਹੀਂ ਹੋਈ ਸੀ, ਤਾਂ ਵੱਖ-ਵੱਖ ਚੀਜ਼ਾਂ ਨੂੰ ਗਿਣਨ ਲਈ ਛੋਟੇ ਪੱਥਰਾਂ , ਸੋਟੀਆਂ ਅਤੇ ਚੱਟਾਨਾਂ 'ਤੇ ਰੇਖਾਵਾਂ ਆਦਿ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਗਿਣਤੀ ਕੀਤੀ ਜਾਂਦੀ ਸੀ। ਜਦੋਂ ਅੰਕਾਂ ਦੀ ਕਾਢ ਕੱਢੀ ਗਈ ਤਾਂ ਗਿਣਤੀ ਲਈ ਇੱਕ ਸਟੈਂਡਰਟ ਸਿਸਟਮ ਅਪਣਾਉਣ ਦੀ ਲੋੜ ਸੀ। ਅੰਕਾਂ ਅਤੇ ਚਿੰਨ੍ਹਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ, ਗਿਣਤੀ ਲਈ ਇੱਕ ਗਣਿਤਿਕ ਸੰਕੇਤ ਅਪਣਾਇਆ ਗਿਆ। ਗਿਣਤੀ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਇਸ ਸੰਕੇਤ ਨੂੰ ਨੰਬਰ ਸਿਸਟਮ ਕਿਹਾ ਗਿਆ।

ਪਾਠ ਦੇ ਉਦੇਸ਼

- ਨੰਬਰ ਸਿਸਟਮ ਨਾਲ ਜਾਣ ਪਛਾਣ।
- ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਮੁੱਢਲੀਆਂ ਸ਼੍ਰੇਣੀਆਂ: ਨਾਨ ਪੁਜੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ, ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ।
- ਆਮ ਵਰਤੇ ਜਾਂਦੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਕਿਸਮਾਂ - ਜਿਵੇਂ ਕਿ: ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ, ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ, ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ, ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ।
- ਵੱਖ-ਵੱਖ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚਕਾਰ ਬਦਲਾਵ (Conversion)।

ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ਵਿਦਿਆਰਥੀ ਸੰਖਿਆਵਾਂ ਨੂੰ ਦਰਸਾਉਣ ਦੇ ਵੱਖ-ਵੱਖ ਤਰੀਕਿਆਂ ਸਬੰਧੀ ਸਿੱਖਣਗੇ, ਜਿਵੇਂ ਕਿ ਡੈਸੀਮਲ, ਬਾਈਨਰੀ, ਔਕਟਲ ਅਤੇ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮਜ਼
- ਉਹ ਆਮ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚਕਾਰ ਬਦਲਾਵ (Conversions) ਨੂੰ ਸਮਝਣਗੇ, ਜਿਵੇਂ ਕਿ ਡੈਸੀਮਲ ਤੋਂ ਬਾਈਨਰੀ, ਡੈਸੀਮਲ ਤੋਂ ਔਕਟਲ, ਡੈਸੀਮਲ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ, ਬਾਈਨਰੀ ਤੋਂ ਡੈਸੀਮਲ, ਬਾਈਨਰੀ ਤੋਂ ਔਕਟਲ, ਬਾਇਨਰੀ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ, ਔਕਟਲ ਤੋਂ ਡੈਸੀਮਲ, ਔਕਟਲ ਤੋਂ ਬਾਈਨਰੀ, ਔਕਟਲ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ, ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਡੈਸੀਮਲ, ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਬਾਈਨਰੀ, ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਔਕਟਲ।

01 ਪਾਠ

ਨੰਬਰ ਸਿਸਟਮ

ਨੰਬਰ ਸਿਸਟਮਜ਼ ਸੰਖਿਆਵਾਂ ਨੂੰ ਪ੍ਰਸਤੁਤ ਕਰਨ ਦਾ ਤਰੀਕਾ ਹੁੰਦੇ ਹਨ ਜੋ ਅਰਥਮੈਟਿਕ ਆਪਰੇਸ਼ਨ (Arithmetic Operation) ਕਰਨ ਲਈ ਇੱਕ ਸਟਰਕਚਰਡ ਤਰੀਕਾ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ। ਹਰੇਕ ਨੰਬਰ ਸਿਸਟਮ ਦੇ ਆਪਣੇ ਚਿੰਨ੍ਹ ਅਤੇ ਸੰਖਿਆਵਾਂ ਨੂੰ ਦਰਸਾਉਣ ਦੇ ਨਿਯਮ ਹੁੰਦੇ ਹਨ। ਅੰਕਾਂ ਅਤੇ ਨਿਯਮਾਂ ਦੀ ਚੋਣ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿਚਕਾਰ ਵੱਖ-ਵੱਖ ਹੁੰਦੀ ਹੈ। ਕੰਪਿਊਟਰ ਵਿਚ ਅਸੀਂ ਜੋ ਵੀ ਲਿਖਦੇ ਹਾਂ, ਚਾਹੇ ਉਹ ਅੱਖਰ ਹੋਣ ਜਾਂ ਸ਼ਬਦ, ਉਹਨਾਂ ਨੂੰ ਕੰਪਿਊਟਰ ਦੁਆਰਾ ਸਭ ਤੋਂ ਪਹਿਲਾਂ ਨੰਬਰਾਂ ਵਿਚ ਤਬਦੀਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਕਿਉਂਕਿ ਕੰਪਿਊਟਰ ਉਹਨਾਂ ਨੂੰ ਸਿਰਫ ਨੰਬਰਾਂ ਦੇ ਰੂਪ ਵਿਚ ਹੀ ਸਮਝਦਾ ਹੈ।

ਇੱਕ ਕੰਪਿਊਟਰ ਅੰਕਾਂ ਦੀ ਪੁਜੀਸ਼ਨ ਅਨੁਸਾਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਨੂੰ ਸਮਝ ਸਕਦਾ ਹੈ। ਸੰਖਿਆਵਾਂ ਨੂੰ ਦਰਸਾਉਣ ਅਤੇ ਕੰਮ ਕਰਨ ਦੀ ਤਕਨੀਕ ਨੂੰ ਨੰਬਰ ਸਿਸਟਮ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀ ਵਰਤੋਂ ਕਿਸੇ ਚੀਜ਼ ਨੂੰ ਗਿਣਨ ਜਾਂ ਮਾਪਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਰੋਜ਼ਾਨਾਂ ਜੀਵਨ ਵਿੱਚ, ਅਸੀਂ **ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ** ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ।

1.1 ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਮੁੱਢਲੀ ਸ਼੍ਰੇਣੀਆਂ (Basic Categories of Number System) :

ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਦੋ ਮੁਢੱਲੀਆਂ ਸ਼੍ਰੇਣੀਆਂ ਹਨ:

1. ਨਾਨ-ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ
2. ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ

1.1.1 ਨਾਨ-ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ (Non-Positional Number System): ਇਹਨਾਂ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ, ਨੰਬਰ ਦਾ ਮੁੱਲ ਉਸਦੀ ਪੁਜ਼ੀਸ਼ਨ ਉੱਪਰ ਨਿਰਭਰ ਨਹੀਂ ਕਰਦਾ। ਇੱਕ ਨਾਨ-ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਸੀਮਤ ਸੰਖਿਆ ਵਿੱਚ ਚਿੰਨ੍ਹਾਂ (Symbols) ਦੀ ਵਰਤੋਂ ਕਰਦੀ ਹੈ ਜਿਸ ਵਿੱਚ ਹਰੇਕ ਚਿੰਨ੍ਹ ਦਾ ਇੱਕ ਮੁੱਲ ਹੁੰਦਾ ਹੈ। ਹਾਲਾਂਕਿ, ਸੰਖਿਆ ਵਿੱਚ ਚਿੰਨ੍ਹ (Symbol) ਦੀ ਪੁਜੀਸ਼ਨ ਦਾ ਆਮ ਤੌਰ 'ਤੇ ਇਸਦੇ ਮੁੱਲ ਨਾਲ ਕੋਈ ਸਬੰਧ ਨਹੀਂ ਹੁੰਦਾ। ਹਰੇਕ ਚਿੰਨ੍ਹ ਦਾ ਮੁੱਲ ਨਿਸ਼ਚਿਤ ਹੁੰਦਾ ਹੈ। ਰੋਮਨ ਨੰਬਰ ਸਿਸਟਮ ਨਾਨ-ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਇੱਕ ਵਧੀਆ ਉਦਾਹਰਣ ਹੈ। ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਸੰਖਿਆਵਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਚਿੰਨ੍ਹਾਂ ਦਾ ਸੈੱਟ S = (I, V, C, X, L, C, D, M) ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।

ਚਿੰਨ੍ਹ	I	V	X	L	C	D	M
ਮੁੱਲ	1	5	10	50	100	500	1000

ਟੇਬਲ 1.1 ਰੋਮਨ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਚਿੰਨ੍ਹਾਂ ਦੇ ਡੈਸੀਮਲ ਮੁੱਲ

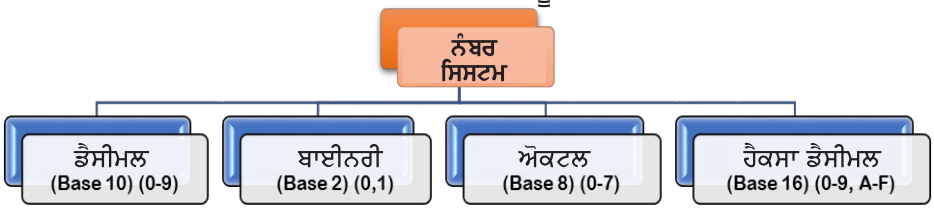
1.1.2 ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ (Positional Number System): ਇਹਨਾਂ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ, ਨੰਬਰ ਦਾ ਮੁੱਲ ਉਸਦੀ ਪੁਜ਼ੀਸ਼ਨ ਉੱਪਰ ਨਿਰਭਰ ਕਰਦਾ ਹੈ। ਇੱਕ ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ, ਹਰੇਕ ਅੰਕ ਦਾ ਮੁੱਲ ਉਸ ਸਥਾਨ ਦੁਆਰਾ ਨਿਰਧਾਰਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਸ ਪੁਜੀਸ਼ਨ ਉੱਪਰ ਉਹ ਅੰਕ ਉਸ ਪੂਰੀ ਸੰਖਿਆ ਵਿੱਚ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ। ਕਿਸੇ ਨੰਬਰ ਵਿਚ ਸਭ ਤੋਂ ਘੱਟ ਸਥਾਨ ਮੁੱਲ (lowest place value) ਉਸ ਨੰਬਰ ਦੇ ਸਭ ਤੋਂ ਸੱਜੇ ਹੱਥ ਦੀ ਪੁਜੀਸ਼ਨ (rightmost position) ਵਾਲੇ ਅੰਕ ਦਾ ਹੁੰਦਾ ਹੈ, ਅਤੇ ਉਸ ਅੰਕ ਦੇ ਖੱਬੇ ਪਾਸੇ ਵੱਲ ਜਾਂਦੇ ਹੋਏ ਅੰਕਾਂ ਦੀ ਪੁਜੀਸ਼ਨ ਅਨੁਸਾਰ ਮੁੱਲ ਦੀ ਕੀਮਤ ਵੱਧ ਸਥਾਨ ਮੁੱਲ (higher place value) ਵੱਲ ਵਧਦੀ ਜਾਂਦੀ ਹੈ। ਹਰੇਕ ਪੁਜੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਇੱਕ ਅਧਾਰ (base) ਜਾਂ ਰੇਡੀਕਸ (radix) ਮੁੱਲ ਹੁੰਦਾ ਹੈ। ਰੇਡੀਕਸ ਮੁੱਲ ਅਨੁਸਾਰ ਪੁਜੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀਆਂ ਉਦਾਹਰਨਾਂ ਹਨ: ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ (ਰੇਡੀਕਸ 10 ਨਾਲ), ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ (ਰੇਡੀਕਸ 2 ਨਾਲ), ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ (ਰੇਡੀਕਸ 8 ਨਾਲ), ਅਤੇ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ (ਰੇਡੀਕਸ 16 ਨਾਲ)। ਇੱਕ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਵਿਲੱਖਣ ਅੰਕਾਂ (unique digits) ਦੀ ਕੁੱਲ ਸੰਖਿਆ ਨੂੰ ਇਸਦੇ ਅਧਾਰ ਜਾਂ ਰੇਡੀਕਸ ਮੁੱਲ ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।

"ਸੰਖਿਆਵਾਂ ਦੀ ਦੁਨੀਆ ਵਿੱਚ ਹਰ ਅੰਕ ਦੀ ਅਹਿਮੀਅਤ ਹੁੰਦੀ ਹੈ"

ਕੀ ਅਸੀਂ ਬਿਨਾਂ ਨੰਬਰਾਂ ਵਾਲੀ ਦੁਨੀਆ ਬਾਰੇ ਸੋਚ ਸਕਦੇ ਹਾਂ ?

1.2 ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਆਮ ਕਿਸਮਾਂ (Common Types of Number Systems)

ਕੰਪਿਊਟਰਾਂ ਦੁਆਰਾ ਆਮ ਤੌਰ ਤੇ ਚਾਰ ਕਿਸਮਾਂ ਦੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਨੂੰ ਸਪੋਰਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ:



ਚਿੱਤਰ 1.1: ਨੰਬਰ ਸਿਸਟਮ ਦੀਆਂ ਆਮ ਕਿਸਮਾਂ

ਲੜੀ ਨੰ	ਨੰਬਰ ਸਿਸਟਮ	ਅਧਾਰ	ਵਰਣਨ	ਉਦਾਹਰਨ
1.	ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ	10	• ਵਰਤੇ ਗਏ ਅੰਕ: 0 ਤੋਂ 9	27
2.	ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ	2	• ਵਰਤੇ ਗਏ ਅੰਕ: 0, 1	11011
3.	ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ	8	• ਵਰਤੇ ਗਏ ਅੰਕ: 0 ਤੋਂ 7	33
4.	ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ	16	• ਵਰਤੇ ਗਏ ਅੰਕ ਅਤੇ ਅੱਖਰ: 0 ਤੋਂ 9 ਅਤੇ A ਤੋਂ F	5A

ਟੇਬਲ 1.2: ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਆਮ ਕਿਸਮਾਂ ਦਾ ਵਰਣਨ

ਨਿਮਨਲਿਖਤ ਟੇਬਲ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸੰਖਿਆਵਾਂ ਦੀ ਲੜੀ (0 ਤੋਂ 20 ਤੱਕ) ਦਿਖਾਉਂਦਾ ਹੈ:

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾਡੇਸਿਮਲ
0	**0000**	**0**	**0**
1	**0001**	**1**	**1**
2	**0010**	**2**	**2**
3	**0011**	**3**	**3**
4	**0100**	**4**	**4**
5	**0101**	**5**	**5**
6	**0110**	**6**	**6**
7	**0111**	**7**	**7**
8	**1000**	**10**	**8**
9	**1001**	**11**	**9**
10	**1010**	**12**	**A**
11	**1011**	**13**	**B**
12	**1100**	**14**	**C**
13	**1101**	**15**	**D**
14	**1110**	**16**	**E**
15	**1111**	**17**	**F**
16	**10000**	**20**	**10**
17	**10001**	**21**	**11**
18	**10010**	**22**	**12**
19	**10011**	**23**	**13**
20	**10100**	**24**	**14**

ਟੇਬਲ 1.3: ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦ ਹੋਏ ਸੰਖਿਆਵਾਂ (0 ਤੋਂ 20 ਤੱਕ) ਦੀ ਉਦਾਹਰਨ

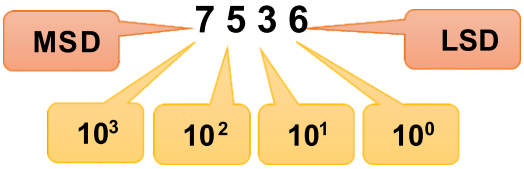
1.2.1 ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ (Decimal Number System):

ਇੱਕ ਨੰਬਰ ਸਿਸਟਮ ਜਿਸ ਵਿੱਚ 10 ਵੱਖ-ਵੱਖ ਚਿੰਨ੍ਹ, 0, 1, 2, 3, 4, 5, 6, 7, 8 ਅਤੇ 9 ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਨੂੰ ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਲਈ ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਅਧਾਰ/ਰੇਡੀਕਸ ਮੁੱਲ 10 ਹੈ। ਇਹੀ ਕਾਰਨ ਹੈ ਕਿ ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਨੂੰ ਬੇਸ-10 ਸਿਸਟਮ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਰੋਜ਼ਾਨਾਂ ਜੀਵਨ ਵਿੱਚ ਸਭ ਤੋਂ ਵੱਧ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਨੰਬਰ ਸਿਸਟਮ ਹੈ। ਇਹ ਨੰਬਰ ਸਿਸਟਮ ਸੰਸਾਰ ਵਿੱਚ ਕਈ ਤਰ੍ਹਾਂ ਦੀਆਂ ਚੀਜ਼ਾਂ ਜਿਵੇਂ ਕਿ ਪੈਸਾ, ਸਮਾਂ, ਮਾਪ (measurements), ਅਤੇ ਹੋਰ ਵੀ ਬਹੁਤ ਚੀਜ਼ਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀ ਕਿਸੇ ਵੀ ਸੰਖਿਆ ਨੂੰ ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ।

ਨੰਬਰਾਂ ਵਿਚ ਹਰੇਕ ਅੰਕ ਦਾ ਉਸਦੀ ਪੁਜੀਸ਼ਨ ਅਨੁਸਾਰ ਸਥਾਨ ਮੁੱਲ (place value) ਹੁੰਦਾ ਹੈ। ਕਿਸੇ ਵੀ ਡੈਸੀਮਲ ਨੰਬਰ (ਬਿਨ੍ਹਾਂ ਫਰੈਕਸ਼ਨਲ ਭਾਗ ਵਾਲਾ (without fractional part) ਵਿਚ ਸਭ ਤੋਂ ਸੱਜੇ ਵੱਲ ਦੀ ਪੁਜੀਸ਼ਨ (right most position) ਵਾਲੇ ਅੰਕ ਨੂੰ ਸਭ ਤੋਂ ਘੱਟ ਮਹੱਤਵਪੂਰਨ ਅੰਕ (Least Significant Digit (LSD)) ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਉਸ ਨੰਬਰ ਦੇ ਸਭ ਤੋਂ ਖੱਬੇ ਪਾਸੇ (left most position) ਵਾਲੇ ਅੰਕ ਨੂੰ ਸਭ ਤੋਂ ਵੱਧ ਮਹੱਤਵਪੂਰਨ ਅੰਕ (Most Significant Digit (MSD)) ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਅਸੀਂ ਨੰਬਰ ਵਿੱਚ ਸੱਜੇ ਤੋਂ ਖੱਬੇ ਪਾਸੇ ਵੱਲ ਜਾਂਦੇ ਹਾਂ, ਤਾਂ ਨੰਬਰ ਦਾ ਮੁੱਲ 10 ਦੇ ਗੁਣਾਂਕ ਨਾਲ ਵਧਦਾ ਜਾਂਦਾ ਹੈ।

ਉਦਾਹਰਨ ਲਈ: ਡੈਸੀਮਲ ਨੰਬਰ 7536 ਵਿੱਚ ਇਕਾਈ ਪੁਜੀਸ਼ਨ ਉੱਪਰ 6, ਦਹਾਈ ਪੁਜੀਸ਼ਨ ਉੱਪਰ 3, ਸੈਂਕੜਾ ਪੁਜੀਸ਼ਨ ਉੱਪਰ 5, ਅਤੇ ਹਜ਼ਾਰ ਪੁਜੀਸ਼ਨ ਉੱਪਰ 7 ਸ਼ਾਮਲ ਹੈ:

$(7x10^3)+(5x10^2)+(3x10^1)+(6x10^0)$
$(7x1000)+(5x100)+(3x10)+(6xl)$
$7000+500+30+6$
7536



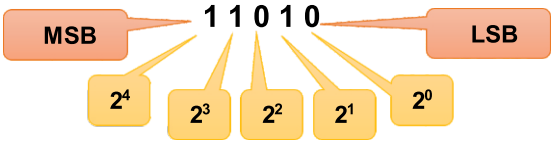
1.2.2 ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ (Binary Number System):

ਇੱਕ ਨੰਬਰ ਸਿਸਟਮ ਜਿਸ ਵਿੱਚ ਦੋ ਵੱਖ-ਵੱਖ ਚਿੰਨ੍ਹ 0 ਅਤੇ 1 ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਨੂੰ ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਅਧਾਰ/ਰੇਡੀਕਸ 2 ਹੈ। ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਨੂੰ ਬੇਸ-2 ਸਿਸਟਮ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਇੱਕ ਅਜਿਹਾ ਨੰਬਰ ਸਿਸਟਮ ਹੈ ਜੋ ਸਿਰਫ਼ ਦੋ ਅੰਕਾਂ 0 ਅਤੇ 1 ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਕਿਸੇ ਵੀ ਬਾਈਨਰੀ ਨੰਬਰ ਵਿਚ ਹਰੇਕ ਅੰਕ ਦੀ ਪੁਜੀਸ਼ਨ 2 ਦੀ ਸ਼ਕਤੀ (power) ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਬਾਈਨਰੀ ਨੰਬਰਾਂ ਦੀ ਵਰਤੋਂ ਆਮ ਤੌਰ 'ਤੇ ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਅਤੇ ਡਿਜੀਟਲ ਇਲੈਕਟ੍ਰੋਨਿਕਸ ਵਿੱਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀ ਕਿਸੇ ਵੀ ਸੰਖਿਆ ਨੂੰ ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ।

ਡਿਜ਼ੀਟਲ ਡਿਵਾਈਸਾਂ ਵਿੱਚ ਸਟੋਰੇਜ ਦੀ ਸਭ ਤੋਂ ਬੁਨਿਆਦੀ ਇਕਾਈ ਨੂੰ ਇੱਕ ਬਿੱਟ (Bit) ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। ਬਿੱਟ (Bit) ਦਾ ਅਰਥ ਹੈ **ਬਾਈਨਰੀ ਡਿਜ਼ਿਟ**। ਬਾਈਨਰੀ ਡਿਜ਼ਿਟ 0 ਅਤੇ 1 ਹਨ। ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਇਹਨਾਂ ਅੰਕਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਇੱਕ ਡਿਜ਼ੀਟਲ ਕੰਪਿਊਟਰ ਹਰ ਚੀਜ਼ ਨੂੰ ਬਾਈਨਰੀ ਅੰਕਾਂ ਦੇ ਰੂਪ ਵਿੱਚ ਸਟੋਰ ਕਰਦਾ ਹੈ। 8 ਬਿੱਟਾਂ ਦੇ ਸੁਮੇਲ ਨੂੰ **ਬਾਈਟ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਦੂਜੇ ਸ਼ਬਦਾਂ ਵਿੱਚ, ਇੱਕ ਬਾਈਟ 8 ਬਿੱਟ ਦੇ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ। ਬਿੱਟ ਮੈਮਰੀ ਦੀ ਸਭ ਤੋਂ ਛੋਟੀ ਇਕਾਈ ਹੈ। ਕੰਪਿਊਟਰ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੀ ਜਾਣਕਾਰੀ ਦਿਖਾਉਣ ਲਈ ਬਿੱਟਾਂ (bits) ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ।

ਸਿਰਫ਼ ਦੋ ਚਿੰਨ੍ਹ 0 ਅਤੇ 1 ਕਿਸੇ ਵੀ ਸੰਖਿਆ ਨੂੰ ਦਰਸਾ ਸਕਦੇ ਹਨ। ਬਿੱਟ 0 ਘੱਟ ਦਰ (lower rate) ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਕਿ ਬਿੱਟ 1 ਨੂੰ ਉੱਚ ਦਰ (higher rate) ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਬਾਈਨਰੀ ਨੰਬਰ ਆਮ ਤੌਰ 'ਤੇ 1 ਅਤੇ 0 ਦੇ ਸਮੂਹਾਂ ਤੋਂ ਬਣੇ ਹੁੰਦੇ ਹਨ। ਹਰੇਕ ਬਾਈਨਰੀ ਨੰਬਰ ਵਿੱਚ, ਬਿੱਟ ਦਾ ਸਥਾਨ ਮੁੱਲ (place value) ਸੱਜੇ ਤੋਂ ਖੱਬੇ ਵੱਲ ਜਾਂਦੇ ਹੋਏ 2 ਦੀ ਸ਼ਕਤੀ (power) ਨਾਲ ਵਧਦਾ ਹੈ। ਬਾਈਨਰੀ ਨੰਬਰ ਦੇ ਸਭ ਤੋਂ ਸੱਜੇ ਬਿੱਟ ਨੂੰ ਸਭ ਤੋਂ ਘੱਟ

ਮਹੱਤਵਪੂਰਨ ਬਿੱਟ (Least Significant Bit (LSB)) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਕਿ ਸਭ ਤੋਂ ਖੱਬੇ ਬਿੱਟ ਨੂੰ ਸਭ ਤੋਂ ਵੱਧ ਮਹੱਤਵਪੂਰਨ ਬਿੱਟ (Most Significant Bit (MSB)) ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਬਾਈਨਰੀ ਨੰਬਰ 11010 ਨੂੰ ਇਸ ਤਰ੍ਹਾਂ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।



ਗੋਟਫ੍ਰਾਈਡ ਵਿਲਹੈਲਮ ਲੀਬਨਿਜ਼ (Gottfried Wilhelm Leibniz)
ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਖੋਜੀ ਹੈ।

1.2.3 ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ (Octal Number System) :

ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ, ਜਿਸ ਨੂੰ ਬੇਸ-8 ਸਿਸਟਮ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ, ਇੱਕ ਅਜਿਹਾ ਨੂਮੇਰੀਕਲ ਸਿਸਟਮ ਹੈ ਜੋ ਅੱਠ ਵੱਖ-ਵੱਖ ਅੰਕਾਂ 0, 1, 2, 3, 4, 5, 6 ਅਤੇ 7 ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਹਰੇਕ ਅੰਕ ਦੀ ਪੁਜੀਸ਼ਨ 8 ਦੀ ਸ਼ਕਤੀ (power) ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਡੈਸੀਮਲ ਅਤੇ ਬਾਈਨਰੀ ਨੰਬਰਾਂ ਦੇ ਮੁਕਾਬਲੇ ਆਮ ਤੌਰ 'ਤੇ ਔਕਟਲ ਨੰਬਰਾਂ ਦੀ ਵਰਤੋਂ ਘੱਟ ਹੁੰਦੀ ਹੈ, ਪਰ ਫਿਰ ਵੀ ਵੱਖ-ਵੱਖ ਖੇਤਰਾਂ ਵਿੱਚ ਜਿਵੇਂ ਕਿ ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮਿੰਗ, ਯੂਨਿਕਸ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਫਾਈਲ ਪਰਮੀਸ਼ਨਾਂ (Permissions) ਸੈੱਟ ਕਰਨ ਲਈ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

Octal ਸ਼ਬਦ ਲਾਤੀਨੀ ਸ਼ਬਦ Oct ਤੋਂ ਆਇਆ ਹੈ ਜਿਸਦਾ ਅਰਥ ਹੈ ਅੱਠ। ਇਸ ਲਈ ਇੱਕ ਅਜਿਹਾ ਨੰਬਰ ਸਿਸਟਮ ਜਿਸ ਵਿੱਚ 8 ਵੱਖ-ਵੱਖ ਚਿੰਨ੍ਹ 0, 1, 2, 3, 4, 5, 6 ਅਤੇ 7 ਹੁੰਦੇ ਹਨ, ਨੂੰ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਕਿਹ ਜਾਂਦਾ ਹੈ। ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਅਧਾਰ/ਰੇਡੀਕਸ 8 ਹੈ। ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀ ਕਿਸੇ ਵੀ ਸੰਖਿਆ ਨੂੰ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ।

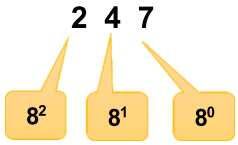
ਬਾਈਨਰੀ ਫਾਰਮੈਟ ਵਿੱਚ ਔਕਟਲ ਨੰਬਰਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ $2^3 = 8$ ਵਾਲਾ ਫਾਰਮੂਲਾ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਤਿੰਨ ਬਾਈਨਰੀ ਅੰਕਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ, ਕਿਸੇ ਵੀ ਔਕਟਲ ਨੰਬਰ (0-7) ਨੂੰ ਬਾਈਨਰੀ ਨੰਬਰ ਫਾਰਮੈਟ ਵਿੱਚ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ ਅਤੇ ਇਸਦੇ ਉਲਟ (Vice-versa) ਬਾਈਨਰੀ ਨੰਬਰ ਨੂੰ ਔਕਟਲ ਫਾਰਮੇਟ ਵਿਚ ਦਰਸਾਉਣ ਲਈ ਵੀ ਇਸੇ ਫਾਰਮੂਲੇ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੇ ਟੇਬਲ 'ਤੇ ਗੌਰ ਕਰੋ:

ਔਕਟਲ ਅੰਕ	ਬਾਈਨਰੀ ਬਰਾਬਰ
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

ਟੇਬਲ: 1.4 ਔਕਟਲ ਨੰਬਰਾਂ ਦੇ ਬਾਈਨਰੀ ਬਰਾਬਰ (Equivalent)

ਕੰਪਿਊਟਿੰਗ ਵਾਤਾਵਰਨ ਵਿੱਚ, ਆਮ ਤੌਰ 'ਤੇ ਇਸਦੀ ਵਰਤੋਂ ਨਾਲ ਕਿਸੇ ਵੀ ਬਾਈਨਰੀ ਨੰਬਰ ਵਿਚ ਤਿੰਨ-ਤਿੰਨ ਬਿਟਸ ਦਾ ਗਰੁੱਪ ਬਣਾ ਕੇ ਉਸਨੂੰ ਔਕਟਲ ਨੰਬਰ ਵਿਚ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ, ਉਪਰੋਕਤ ਟੇਬਲ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਬਾਈਨਰੀ ਨੰਬਰ 101100 ਨੂੰ ਔਕਟਲ ਫਾਰਮੈਟ ਵਿੱਚ 54 ਨਾਲ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸਦੇ ਉਲਟ (Vice-versa) ਅਰਥਾਤ ਔਕਟਲ ਨੰਬਰ ਨੂੰ ਬਾਈਨਰੀ ਫਾਰਮੇਟ ਵਿਚ ਦਰਸਾਉਣਾ ਵੀ ਸੰਭਵ ਹੈ।

ਇੱਕ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਨੂੰ ਪੁਜੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਇੱਕ ਕਿਸਮ ਵਜੋਂ ਵੀ ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ। ਕਿਸੇ ਵੀ ਔਕਟਲ ਸੰਖਿਆ ਵਿੱਚ ਸੱਜੇ ਤੋਂ ਖੱਬੇ ਵੱਲ ਜਾਂਦੇ ਹੋਏ ਔਕਟਲ ਅੰਕਾਂ ਦਾ ਸਥਾਨ ਮੁੱਲ (place value) 8 ਦੀ ਸ਼ਕਤੀ (power) ਨਾਲ ਵਧਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਔਕਟਲ ਨੰਬਰ 247 ਨੂੰ ਇਸ ਤਰ੍ਹਾਂ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ:



ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਦੇ ਮੁਕਾਬਲੇ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਅਤੇ ਟ੍ਰਾਂਸਮਿਟ ਕਰਨ ਦਾ ਵਧੇਰੇ ਕੁਸ਼ਲ ਤਰੀਕਾ ਹੈ ਕਿਉਂਕਿ ਇਸ ਨੂੰ ਘੱਟ ਡਿਜ਼ੀਟਸ (ਬਿੱਟਸ) ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ।

1.2.4 ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ (Hexa-Decimal Number System) :

ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ, ਜਿਸ ਨੂੰ ਅਕਸਰ "Hex" ਜਾ ਅਧਾਰ-16 ਵਾਲਾ ਨੰਬਰ ਸਿਸਟਮ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਹੈਕਸਾਡੈਸੀਮਲ ਸ਼ਬਦ ਦੋ ਸ਼ਬਦਾਂ 'ਹੈਕਸਾ' ਅਤੇ 'ਡੈਸੀਮਲ' ਤੋਂ ਬਣਿਆ ਹੈ, ਜਿੱਥੇ 'ਹੈਕਸਾ' ਦਾ ਅਰਥ 6 ਹੈ ਅਤੇ 'ਡੈਸੀਮਲ' ਦਾ ਮਤਲਬ 10 ਹੈ। ਇਸ ਲਈ ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ 16 ਚਿੰਨ੍ਹਾਂ: 0 ਤੋਂ 9 ਤੱਕ ਦਸ ਡੈਸੀਮਲ ਅੰਕ ਅਤੇ ਛੇ ਅੱਖਰ-ਮੁੱਲ A ਤੋਂ F (ਜਾਂ a ਤੋਂ f) ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਸਾਰੇ ਚਿੰਨ੍ਹ: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, ਅਤੇ F ਹਨ। ਇੱਥੇ, 'A' 10 ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ, 'B' 11 ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ, ਅਤੇ ਇਸੇ ਤਰ੍ਹਾਂ 'F' 15 ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰਾਂ ਦੀਆਂ ਉਦਾਹਰਨਾਂ ਹਨ: 8A3, 7F16, 439, FA ਆਦਿ।

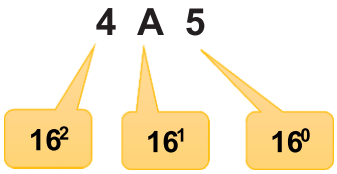
ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰਾਂ ਦੀ ਵਰਤੋਂ ਆਮ ਤੌਰ 'ਤੇ ਕੰਪਿਊਟਿੰਗ ਵਿੱਚ ਪ੍ਰੋਗਰਾਮਿੰਗ ਅਤੇ ਡਿਜੀਟਲ ਇਲੈਕਟ੍ਰੋਨਿਕਸ ਵਿੱਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਕਿਉਂਕਿ ਉਹ ਬਾਈਨਰੀ ਡਾਟਾ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਇੱਕ ਸੰਖੇਪ ਤਰੀਕਾ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ। ਬਾਈਨਰੀ ਫਾਰਮੈਟ ਵਿੱਚ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਫਾਰਮੂਲਾ 2^4=16 ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਹਰੇਕ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕ ਚਾਰ ਬਾਈਨਰੀ ਅੰਕਾਂ (ਬਿੱਟਾਂ) ਨਾਲ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਾਈਨਰੀ ਡਾਟਾ ਉਪਰ ਕੰਮ ਕਰਨਾ ਮਨੁੱਖ ਲਈ ਆਸਾਨ ਹੋ ਜਾਂਦਾ ਹੈ। ਹੇਠ ਦਿੱਤੀ ਟੇਬਲ 'ਤੇ ਗੌਰ ਕਰੋ:

ਹੈਕਸਾ ਡੈਸੀਮਲ ਨੰਬਰ	ਬਾਈਨਰੀ ਬਰਾਬਰ
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

ਟੇਬਲ : 1.5 ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰਾਂ ਦੇ ਬਾਈਨਰੀ ਬਰਾਬਰ (Equivalent)

ਉਦਾਹਰਨ ਲਈ: ਉਪਰੋਕਤ ਟੇਬਲ 1.5 ਅਨੁਸਾਰ ਬਾਈਨਰੀ ਨੰਬਰ 10111000 ਨੂੰ ਹੈਕਸਾਡੈਸੀਮਲ ਫਾਰਮੈਟ ਵਿੱਚ B8 ਨਾਲ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸੇ ਤਰਾਂ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਨੂੰ ਬਾਈਨਰੀ ਫਾਰਮੇਟ ਵਿਚ ਦਰਸਾਉਣਾ ਵੀ ਸੰਭਵ ਹੈ।

ਇੱਕ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਨੂੰ ਪੁਜੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਇੱਕ ਕਿਸਮ ਵਜੋਂ ਵੀ ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ। ਕਿਸੇ ਵੀ ਹੈਕਸਾਡੈਸੀਮਲ ਸੰਖਿਆ ਵਿੱਚ ਸੱਜੇ ਤੋਂ ਖੱਬੇ ਵੱਲ ਜਾਂਦੇ ਹੋਏ ਹਰੇਕ ਹੈਕਸਾਡੈਸੀਮਲ ਚਿੰਨ੍ਹ ਦਾ ਸਥਾਨ ਮੁੱਲ (place value) 16 ਦੀ ਸ਼ਕਤੀ ਨਾਲ ਵਧਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ 4A5 ਨੂੰ ਇਸ ਤਰ੍ਹਾਂ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ:

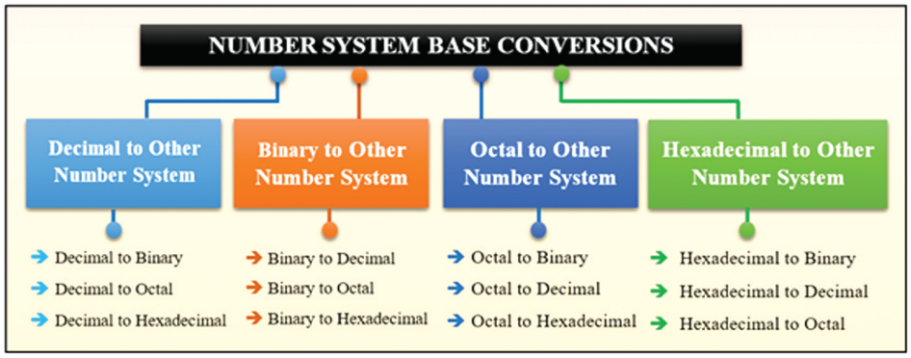


ਜੌਹਨ ਵਿਲੀਅਮਜ਼ ਨਿਸਟ੍ਰੋਮ (John Williams Nystrom) ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਪਿਤਾਮਾ ਹੈ। ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਵਰਤੋਂ ਕੰਪਿਊਟਰ ਮੈਮਰੀ ਦੇ ਐਡਰੈਸਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

1.3 ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿਚਕਾਰ ਬਦਲਾਵ (Conversion among Number Systems)

ਇਸ ਪਾਠ ਵਿੱਚ ਅਸੀਂ ਚਾਰ ਆਮ ਕਿਸਮਾਂ ਦੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਬਾਰੇ ਚਰਚਾ ਕੀਤੀ ਹੈ। ਇਹਨਾਂ ਵਿੱਚੋਂ ਹਰ ਇੱਕ ਸਿਸਟਮ ਦੇ ਨੰਬਰਾਂ ਨੂੰ ਬਾਕੀ ਤਿੰਨ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਿਆ (convert) ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਇਹਨਾਂ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚਕਾਰ ਹੇਠ ਲਿਖੀਆਂ ਕਿਸਮਾਂ ਦੇ ਬਦਲਾਵ (conversion) ਸੰਭਵ ਹਨ:

1. ਡੈਸੀਮਲ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਾਵ
2. ਬਾਈਨਰੀ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਾਵ
3. ਔਕਟਲ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਾਵ
4. ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਾਵ



ਚਿੱਤਰ 1.2: ਨੰਬਰ ਸਿਸਟਮ ਬੇਸ ਕਨਵਰਜ਼ਨਸ

1.3.1 ਡੈਸੀਮਲ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਾਵ (Conversion from Decimal to Other Number Systems):

ਡੈਸੀਮਲ ਨੰਬਰ ਨੂੰ ਇੱਕ ਇੰਟੀਜ਼ਰ ਅੰਕ ਦੇ ਰੂਪ ਵਿਚ ਜਾਂ ਫ਼ੈਕਸ਼ਨਲ ਭਾਗ ਸਮੇਤ ਇਕ ਇੰਟੀਜ਼ਰ ਅੰਕ (ਫਲੋਟਿੰਗ ਨੰਬਰ) ਦੇ ਨਾਲ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ। ਜਦੋਂ ਡੈਸੀਮਲ ਨੰਬਰ ਨੂੰ ਫ਼ੈਕਸ਼ਨਲ ਭਾਗ ਸਮੇਤ ਇੱਕ ਇੰਟੀਜ਼ਰ ਅੰਕ ਦੇ ਨਾਲ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਸਾਨੂੰ ਉਸ ਡੈਸੀਮਲ ਨੰਬਰ ਦੇ ਦੋਵੇਂ ਭਾਗਾਂ (ਇੰਟੀਜ਼ਰ ਭਾਗ ਅਤੇ ਫ਼ੈਕਸ਼ਨ ਭਾਗ) ਨੂੰ ਵੱਖ-ਵੱਖ ਤੌਰ ਤੇ ਬਦਲਾਵ (Convert) ਕਰਨਾ ਪਵੇਗਾ। ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਅਸੀਂ ਡੈਸੀਮਲ ਨੰਬਰ ਨੂੰ ਕਿਸੇ ਵੀ ਅਧਾਰ 'r' ਵਾਲੇ ਨੰਬਰ ਸਿਸਟਮ ਵਿਚ ਬਦਲ ਸਕਦੇ ਹਾਂ:

- **ਇੰਟੀਜ਼ਰ ਭਾਗ ਦਾ ਬਦਲਾਵ (Conversion of Integer Part):** ਇੰਟੀਜ਼ਰ ਭਾਗ ਅਤੇ ਇਸਨੂੰ ਭਾਗ

(divide) ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਭਾਗਫਲ (quotient) ਉਪਰ ਬੇਸ 'r' (ਜਿਵੇਂ ਕਿ 2 ਜਾਂ 8 ਜਾਂ 16) ਨਾਲ ਡਿਵੀਜ਼ਨ (division) ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰਦੇ ਰਹੋ ਅਤੇ ਭਾਗਫਲ ਜ਼ੀਰੋ ਆਉਣ ਤੱਕ ਸਾਰੇ ਸ਼ੇਸ਼ਫਲਾਂ (Reminders) ਦੀ ਲਿਸਟ ਬਣਾਓ। ਫਿਰ ਸਭ ਤੋਂ ਵੱਧ ਮਹੱਤਵਪੂਰਨ ਬਿੱਟ (MSB) ਤੋਂ ਸਭ ਤੋਂ ਘੱਟ ਮਹੱਤਵਪੂਰਨ ਬਿੱਟ (LSB) ਵੱਲ ਸਾਰੇ ਸ਼ੇਸ਼ਫਲਾਂ (Remainders) ਨੂੰ ਨੋਟ ਕਰਦੇ ਰਹੋ ਤਾਂ ਜੋ ਬੇਸ 'r' ਵਾਲੇ ਨੰਬਰ ਸਿਸਟਮ ਵਿਚ ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ (ਇੰਟੀਜ਼ਰ ਭਾਗ) ਦੇ ਬਰਾਬਰ ਦੀ ਸੰਖਿਆ ਪ੍ਰਾਪਤ ਕੀਤੀ ਜਾ ਸਕੇ। ਸਭ ਤੋਂ ਪਹਿਲੇ ਪ੍ਰਾਪਤ ਕੀਤੇ ਗਏ ਸ਼ੇਸ਼ਫਲ ਮੁੱਲ (Remainder Values) ਨੂੰ LSB ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਕਿ ਆਖਰੀ ਪ੍ਰਾਪਤ ਹੋਏ ਸ਼ੇਸ਼ਫਲ ਮੁੱਲ (Remainder Values) ਨੂੰ MSB ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।

- **ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਦਾ ਬਦਲਾਵ (Conversion of Fractional Part):** ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਅਤੇ ਇਸਨੂੰ ਗੁਣਾ (multiply) ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਕ੍ਰਮਵਾਰ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਉਪਰ ਬੇਸ 'r' (ਜਿਵੇਂ ਕਿ 2 ਜਾਂ 8 ਜਾਂ 16) ਨਾਲ ਮਲਟੀਪਲਾਈ (ਗੁਣਾ) ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰਦੇ ਰਹੋ। ਗੁਣਾਂ (multiply) ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੀਆਂ ਕੈਰੀਜ਼ (carries) ਨੂੰ ਉਦੋਂ ਤੱਕ ਨੋਟ ਕਰਦੇ ਰਹੋ ਜਦੋਂ ਤੱਕ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਜ਼ੀਰੋ ਜਾਂ ਲਗਭਗ ਜ਼ੀਰੋ ਨਹੀਂ ਬਣ ਜਾਂਦਾ। ਡੈਸੀਮਲ ਫਾਰਮੇਟ ਦੇ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਦੇ ਬਰਾਬਰ ਦੀ ਸੰਖਿਆਂ ਨੂੰ ਬੇਸ 'r' ਵਿਚ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਕੈਰੀਜ਼ ਨੂੰ ਆਮ ਕ੍ਰਮ (ਭਾਵ ਜਿਸ ਕ੍ਰਮ ਵਿਚ ਕੈਰੀਜ਼ ਪ੍ਰਾਪਤ ਹੋਈਆਂ ਉਸੇ ਕ੍ਰਮ ਵਿਚ) ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ।

ਹੁਣ, ਆਓ ਕੁਝ ਉਦਾਹਰਣਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸ ਪ੍ਰਕਿਰਿਆ 'ਤੇ ਇੱਕ ਨਜ਼ਰ ਮਾਰੀਏ:

1.3.1.1 ਡੈਸੀਮਲ ਤੋਂ ਬਾਈਨਰੀ ਵਿਚ ਬਦਲਾਵ (Decimal to Binary Conversion):

ਡੈਸੀਮਲ ਤੋਂ ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਕਨਵਰਜ਼ਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਪ੍ਰੋਸੈਸ ਲਾਗੂ ਕਰੋ:

- ਪਹਿਲੇ ਪੜਾਅ ਵਿੱਚ, ਬਾਈਨਰੀ ਅਧਾਰ/ਰੇਡੀਕਸ ਮੁੱਲ (ਜੋ ਕਿ 2 ਹੈ) ਦੇ ਨਾਲ ਦਿੱਤੇ ਗਏ ਇੰਟੀਜ਼ਰ ਭਾਗ ਅਤੇ ਭਾਗ (divide) ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਕ੍ਰਮਵਾਰ ਭਾਗਫਲਾਂ (quotient) ਉਪਰ ਡਿਵੀਜ਼ਨ (division) ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰੋ। ਭਾਗਫਲ ਜ਼ੀਰੋ ਆਉਣ ਤੱਕ ਸਾਰੇ ਸ਼ੇਸ਼ਫਲਾਂ (Remainder) ਨੂੰ ਸੂਚੀਬੱਧ ਕਰੋ। MSB ਤੋਂ LSB ਵੱਲ ਸਾਰੇ ਸ਼ੇਸ਼ਫਲਾਂ (Remainders) ਨੂੰ ਲਿਖੋ।
- ਅੱਗਲੇ ਪੜਾਅ ਵਿੱਚ, ਬਾਈਨਰੀ ਬੇਸ/ਰੇਡੀਕਸ ਮੁੱਲ (ਜੋ ਕਿ 2 ਹੈ) ਦੇ ਨਾਲ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਅਤੇ ਗੁਣਾ (multiply) ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਕ੍ਰਮਵਾਰ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਉਪਰ ਮਲਟੀਪਲਾਈ (ਗੁਣਾ) ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰੋ। ਕੈਰੀਜ਼ (carries) ਨੂੰ ਉਦੋਂ ਤੱਕ ਨੋਟ ਕਰੋ ਜਦੋਂ ਤੱਕ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਜ਼ੀਰੋ ਜਾਂ ਲਗਭਗ ਜ਼ੀਰੋ ਨਹੀਂ ਹੋ ਜਾਂਦਾ।

ਉਦਾਹਰਨ: $(38.25)_{10}$

ਸਟੈੱਪ 1 : ਇੰਟੀਜ਼ਰ ਅੰਕ 38 ਅਤੇ ਇਸਨੂੰ ਭਾਗ (divide) ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਭਾਗਫਲਾਂ (quotients) ਨੂੰ ਬਾਈਨਰੀ ਬੇਸ/ਰੇਡੀਕਸ ਮੁੱਲ 2 ਨਾਲ ਵੰਡਦੇ ਰਹੋ ਇਸਦੇ ਕ੍ਰਮਵਾਰ ਭਾਗਫਲਾਂ (quotients) ਅਤੇ ਸ਼ੇਸ਼ਫਲਾਂ (Remainder Values) ਦੀ ਸੂਚੀ ਬਣਾਓ।

ਓਪਰੇਸ਼ਨ	ਭਾਗਫਲ (Quotient)	ਸ਼ੇਸ਼ਫਲ (Reminder)
38/2	19	0 (LSB)
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1 (MSB)

ਅਸੀਂ ਪ੍ਰਾਪਤ ਕਰਦੇ ਹਾਂ $(38)_{10}$ = $(100110)_2$

ਸਟੈੱਪ 2 : ਹੁਣ, ਫਰੈਕਸ਼ਨਲ ਭਾਗ 0.25 ਅਤੇ ਇਸਨੂੰ ਗੁਣਾ ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਕ੍ਰਮਵਾਰ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਨੂੰ ਬਾਈਨਰੀ ਬੇਸ/ਰੇਡੀਕਸ ਅਰਥਾਤ 2 ਨਾਲ ਗੁਣਾ ਕਰਦੇ ਰਹੋ ਅਤੇ ਕੈਰੀਜ਼ ਨੂੰ ਕ੍ਰਮਵਾਰ ਨੋਟ ਕਰੋ।

ਓਪਰੇਸ਼ਨ	ਨਤੀਜਾ	ਕੈਰੀਜ਼ (Carry)
0.25×2	0.50	0 ↓
0.50×2	0.00	1

ਅਸੀਂ ਪ੍ਰਾਪਤ ਕਰਦੇ ਹਾਂ $(0.25)_{10} = (0.1)_2$

ਹੁਣ, ਅਸੀਂ ਦਿੱਤੇ ਗਏ ਡੈਸੀਮਲ ਨੰਬਰ ਲਈ ਪੂਰਾ ਬਾਈਨਰੀ ਨੰਬਰ ਲਿਖ ਸਕਦੇ ਹਾਂ:

$$(38.25)_{10} = (100110.01)_2$$

1.3.1.2 ਡੈਸੀਮਲ ਤੋਂ ਔਕਟਲ ਵਿਚ ਬਦਲਾਵ (Decimal to Octal Conversion):

ਡੈਸੀਮਲ ਤੋਂ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿਚ ਬਦਲਾਵ ਲਈ, ਉਹੀ ਪ੍ਰਕਿਰਿਆ ਵਰਤੀ ਜਾ ਸਕਦੀ ਹੈ ਜਿਵੇਂ ਅਸੀਂ ਪਿਛਲੇ ਕਿਸਮ ਦੇ ਬਦਲਾਵ ਵਿੱਚ ਕੀਤੀ ਸੀ। ਇੱਥੇ ਸਿਰਫ ਫਰਕ ਇਹ ਹੈ ਕਿ ਸਾਨੂੰ 2 ਦੀ ਥਾਂ ਅਧਾਰ/ਰੇਡੀਕਸ ਮੁੱਲ 8 ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਪਵੇਗੀ:

ਉਦਾਹਰਨ: $(69.25)_{10}$

ਸਟੈੱਪ 1 : ਇੰਟੀਜ਼ਰ ਅੰਕ 69 ਅਤੇ ਇਸਨੂੰ ਭਾਗ (divide) ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਕ੍ਰਮਵਾਰ ਭਾਗਫਲਾਂ (Quotients) ਨੂੰ ਅਧਾਰ 8 ਨਾਲ ਭਾਗ ਕਰੋ ਅਤੇ ਆਉਣ ਵਾਲੇ ਸ਼ੇਸ਼ਫਲਾਂ (remainders) ਨੂੰ ਕ੍ਰਮਵਾਰ ਨੋਟ ਕਰਦੇ ਰਹੋ:

ਓਪਰੇਸ਼ਨ	ਭਾਗਫਲ	ਸ਼ੇਸ਼ਫਲ
69/8	8	5 ↑
8/8	1	0

$(69)_{10} = (105)_8$

ਸਟੈੱਪ 2: ਹੁਣ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ 0.25 ਅਤੇ ਗੁਣਾ ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਫ੍ਰੈਕਸ਼ਨਲ ਬਾਗ ਨੂੰ ਅਧਾਰ 8 ਨਾਲ ਗੁਣਾ ਕਰਦੇ ਰਹੋ ਅਤੇ ਕੈਰੀਜ਼ ਨੂੰ ਨੋਟ ਕਰਦੇ ਰਹੋ।

ਓਪਰੇਸ਼ਨ	ਨਤੀਜਾ	ਕੈਰੀਜ਼ (Carry)
0.25×8	0	2 ↓

$(0.25)_{10} = (2)_8$

ਇਸ ਤਰ੍ਹਾਂ ਡੈਸੀਮਲ ਨੰਬਰ ਦੇ ਬਰਾਬਰ ਦਾ ਔਕਟਲ ਨੰਬਰ ਪ੍ਰਾਪਤ ਹੋ ਜਾਵੇਗਾ:

$$(69.25)_{10} = (105.2)_8$$

1.3.1.3 ਡੈਸੀਮਲ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ ਵਿਚ ਬਦਲਾਵ (Decimal to Hexadecimal Conversin) :

ਡੈਸੀਮਲ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ ਬਦਲਾਵ ਲਈ ਉਹੀ ਪ੍ਰਕਿਰਿਆ ਵਰਤੀ ਜਾ ਸਕਦੀ ਹੈ ਜੋ ਅਸੀਂ ਪਿਛਲੇ ਕਨਵਰਜ਼ਨ ਵਿੱਚ ਕੀਤੀ ਸੀ। ਇੱਥੇ ਸਿਰਫ ਫਰਕ ਇਹ ਹੈ ਕਿ ਸਾਨੂੰ 2 ਜਾਂ 8 ਦੀ ਜਗ੍ਹਾ ਅਧਾਰ 16 ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਪਵੇਗੀ:

ਉਦਾਹਰਨ: $(70.25)_{10}$

ਸਟੈੱਪ 1: ਇੰਟੀਜ਼ਰ ਅੰਕ 70 ਅਤੇ ਇਸਨੂੰ ਭਾਗ ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਕ੍ਰਮਵਾਰ ਭਾਗਫਲਾਂ (Quotient) ਨੂੰ ਅਧਾਰ 16 ਨਾਲ ਭਾਗ ਕਰੋ ਅਤੇ ਆਉਣ ਵਾਲੇ ਸ਼ੇਸ਼ਫਲਾਂ (remainders) ਨੂੰ ਕ੍ਰਮਵਾਰ ਨੋਟ ਕਰਦੇ ਰਹੋ:

ਓਪਰੇਸ਼ਨ	ਭਾਗਫਲ	ਸ਼ੇਸ਼ਫਲ
70/16	4	6 ↑
4/16	0	4

$(70)_{10} = (46)_{16}$

ਸਟੈੱਪ 2: ਹੁਣ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ 0.25 ਅਤੇ ਗੁਣਾ ਕਰਨ ਉਪਰੰਤ ਆਉਣ ਵਾਲੇ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ ਨੂੰ ਅਧਾਰ 16 ਨਾਲ ਗੁਣਾ ਕਰਦੇ ਰਹੋ ਅਤੇ ਕੈਰੀਜ਼ ਨੂੰ ਨੋਟ ਕਰਦੇ ਰਹੋ।

ਉਪਰੇਸ਼ਨ	ਭਾਗਫਲ	ਸ਼ੇਸ਼ਫਲ
0.25×16	0	4 ↓

$(0.25)_{10} = (4)_{16}$

ਇਸ ਤਰ੍ਹਾਂ ਡੈਸੀਮਲ ਨੰਬਰ ਦੇ ਬਰਾਬਰ ਦਾ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਪ੍ਰਾਪਤ ਹੋ ਜਾਵੇਗਾ। $(70.25)_{10}$ ਹੈ $(46.4)_{16}$

test yourself

ਹੇਠ ਦਿੱਤੇ ਨੂੰ ਪੂਰਾ ਕਰੋ:

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾਡੈਸੀਮਲ
52			

1.3.2 ਬਾਈਨਰੀ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿਚਕਾਰ ਬਦਲਾਵ (Binary to Other Number Systems) :

ਬਾਈਨਰੀ ਸੰਖਿਆਵਾਂ ਲਈ ਤਿੰਨ ਸੰਭਾਵਿਤ ਬਦਲਾਵ ਹਨ, ਜਿਵੇਂ ਕਿ, ਬਾਈਨਰੀ ਤੋਂ ਡੈਸੀਮਲ, ਬਾਈਨਰੀ ਤੋਂ ਔਕਟਲ, ਅਤੇ ਬਾਈਨਰੀ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ। ਬਾਈਨਰੀ ਨੰਬਰ ਦਾ ਡੈਸੀਮਲ ਵਿੱਚ ਬਦਲਾਵ (conversion) ਪ੍ਰੋਸੈਸ ਬਾਕੀ ਸਾਰੀਆਂ ਤੋਂ ਵੱਖਰਾ ਹੈ।

1.3.2.1 ਬਾਈਨਰੀ ਤੋਂ ਡੈਸੀਮਲ ਵਿਚ ਬਦਲਾਵ (Binary to Decimal Conversion):

ਇਸ ਪ੍ਰੋਸੈਸ ਵਿੱਚ ਅਸੀਂ ਬਾਈਨਰੀ ਨੰਬਰ ਦੇ ਬਿੱਟਾਂ ਨੂੰ ਇਸਦੇ ਅਨੁਸਾਰੀ ਸਥਿਤੀ ਦੇ ਭਾਰਾਂ (corresponding positional weights) ਨਾਲ ਗੁਣਾ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰਦੇ ਹਾਂ ਅਤੇ ਅੰਤ ਵਿੱਚ, ਅਸੀਂ ਇਹਨਾਂ ਸਾਰੀਆਂ ਗਣਨਾਵਾਂ (products) ਦੇ ਨਤੀਜਿਆਂ ਨੂੰ ਜੋੜਦੇ ਹਾਂ। ਆਉ ਇਹ ਸਮਝਣ ਲਈ ਇੱਕ ਉਦਾਹਰਣ ਲੈਂਦੇ ਹਾਂ ਕਿ ਬਾਈਨਰੀ ਤੋਂ ਡੈਸੀਮਲ ਵਿਚ ਬਦਲਾਵ ਕਿਵੇਂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ:

ਉਦਾਹਰਨ: $(1001.001)_2$

$(10010.001)_2$ ਦੇ ਹਰੇਕ ਬਿੱਟ ਨੂੰ ਉਹਨਾਂ ਦੇ ਅਨੁਸਾਰੀ ਸਥਿਤੀ ਦੇ ਭਾਰਾਂ (corresponding positional weights) ਦੇ ਨਾਲ ਗੁਣਾ ਕਰੋ, ਅਤੇ ਅੰਤ ਵਿੱਚ ਇਹਨਾਂ ਸਾਰੀਆਂ ਗਣਨਾਵਾਂ (products) ਦੇ ਨਤੀਜਿਆਂ ਨੂੰ ਜੋੜੋ:

$$
\begin{aligned}
(1001.001)_2 &= (1\times2^3)+(0\times2^2)+(0\times2^1)+(1\times2^0)+(0\times2^{-1})+(0\times2^{-2})+(1\times2^{-3}) \\
&= (1\times8)+(0\times4)+(0\times2)+(1\times1)+(0\times\tfrac{1}{2})+(0\times\tfrac{1}{4})+(1\times\tfrac{1}{8}) \\
&= 8+0+0+1+0.125 \\
&= (9.125)_{10}
\end{aligned}
$$

1.3.2.2 ਬਾਈਨਰੀ ਤੋਂ ਔਕਟਲ ਵਿਚ ਬਦਲਾਵ (Binary to Octal Conversion):

ਇੱਕ ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਤਿੰਨ ਬਿੱਟਾਂ ਦਾ ਸਮੂਹ ਇੱਕ ਔਕਟਲ ਅੰਕ ਦੇ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਟੇਬਲ 1.4 ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਮਦਦ ਨਾਲ ਅਸੀਂ ਇੱਕ ਬਾਈਨਰੀ ਨੰਬਰ ਨੂੰ ਇਸਦੇ ਬਰਾਬਰ ਦੇ ਔਕਟਲ ਨੰਬਰ ਵਿੱਚ ਬਦਲ ਸਕਦੇ ਹਾਂ:

- ਸਭ ਤੋਂ ਪਹਿਲਾਂ ਸਾਨੂੰ ਬਾਇਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਨੂੰ ਤਿੰਨ-ਤਿੰਨ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਬਣਾਉਣੇ ਪੈਣਗੇ।
- ਜੇਕਰ ਤਿੰਨ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਵਿੱਚ ਇੱਕ ਜਾਂ ਦੋ ਬਿੱਟ ਘੱਟ ਹਨ, ਤਾਂ ਅਸੀਂ ਸਿਰੇ ਦੇ ਪਾਸਿਆਂ 'ਤੇ ਲੋੜੀਂਦੇ ਜ਼ੀਰੋ ਜੋੜ ਸਕਦੇ ਹਾਂ।
- ਉੱਪਰ ਦੱਸੇ ਅਨੁਸਾਰ ਬਿੱਟਾਂ ਨੂੰ ਜੋੜਨ ਤੋਂ ਬਾਅਦ, ਟੇਬਲ 1.4 ਵਿੱਚ ਦਿੱਤੇ ਅਨੁਸਾਰ ਹਰੇਕ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਔਕਟਲ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

ਆਓ ਹੁਣ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਦਲਾਵ ਦੇ ਇਸ ਪ੍ਰੋਸੈਸ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ:

ਉਦਾਹਰਨ: $(10101.0011)_2$

- ਸਭ ਤੋਂ ਪਹਿਲਾਂ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਤਿੰਨ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਬਣਾਓ:

1 0 1 0 1 . 0 0 1 1

- ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਦੇ ਸੱਜੇ ਪਾਸੇ, ਆਖਰੀ ਜੋੜੇ ਵਿੱਚ ਸਿਰਫ਼ ਇੱਕ ਬਿੱਟ ਹੈ। ਇਸ ਨੂੰ ਤਿੰਨ ਬਿੱਟਾਂ ਦਾ ਪੂਰਾ ਜੋੜਾ ਬਣਾਉਣ ਲਈ, ਅਸੀਂ ਸੱਜੇ ਪਾਸੇ ਵੱਲ ਦੋ ਜ਼ੀਰੋ ਹੋਰ ਜੋੜਾਂਗੇ। ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਦੇ ਖੱਬੇ ਪਾਸੇ, ਆਖਰੀ ਜੋੜੇ ਦੇ ਦੋ ਬਿੱਟ ਹਨ, ਇਸ ਲਈ ਅਸੀਂ ਖੱਬੇ ਪਾਸੇ ਇੱਕ ਜ਼ੀਰੋ ਹੋਰ ਜੋੜਾਂਗੇ।

0 1 0 1 0 1 . 0 0 1 1 0 0

- ਹੁਣ, ਔਕਟਲ ਦੇ ਹਰੇਕ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਔਕਟਲ ਅੰਕਾਂ ਨੂੰ ਟੇਬਲ 1.4 ਵਿੱਚ ਦਿੱਤੇ ਅਨੁਸਾਰ ਲਿਖੋ।

$$(10101.0011)_2=(25.14)_8$$

1.3.2.3 ਬਾਈਨਰੀ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ ਬਦਲਾਵ (Binary to Hexadecimal Conversion):

ਇੱਕ ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਚਾਰ ਬਿੱਟਾਂ ਦਾ ਜੋੜਾ ਇੱਕ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕ ਦੇ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਟੇਬਲ 1.5 ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਸੀਂ ਇੱਕ ਬਾਈਨਰੀ ਨੰਬਰ ਨੂੰ ਇਸਦੇ ਬਰਾਬਰ ਦੇ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਵਿੱਚ ਬਦਲ ਸਕਦੇ ਹਾਂ:

- ਸਭ ਤੋਂ ਪਹਿਲਾਂ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਚਾਰ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਬਣਾਓ।
- ਜੇਕਰ ਬਹੁਤ ਖੱਬੇ ਅਤੇ ਸੱਜੇ ਪਾਸੇ ਵਾਲੇ ਚਾਰ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਵਿੱਚ ਇੱਕ, ਦੋ ਜਾਂ ਤਿੰਨ ਬਿੱਟ ਘੱਟ ਹਨ, ਤਾਂ ਅਸੀਂ ਸਿਰੇ ਵਾਲੇ ਪਾਸਿਆਂ 'ਤੇ ਲੋੜੀਂਦੇ ਜ਼ੀਰੋ ਜੋੜ ਸਕਦੇ ਹਾਂ।
- ਉੱਪਰ ਦੱਸੇ ਅਨੁਸਾਰ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਬਨਾਉਣ ਤੋਂ ਬਾਅਦ ਟੇਬਲ 1.5 ਵਿੱਚ ਦਿੱਤੇ ਅਨੁਸਾਰ ਹਰੇਕ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਹੈਕਸਾਡੈਸੀਮਲ ਚਿੰਨ੍ਹ ਲਿਖੋ।
- ਆਓ ਹੁਣ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਦਲਾਵ (conversion) ਦੇ ਇਸ ਪ੍ਰੋਸੈਸ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ

ਉਦਾਹਰਨ: $(110101011.0011)_2$

- ਸਭ ਤੋਂ ਪਹਿਲਾਂ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਚਾਰ-ਚਾਰ ਬਿੱਟਾਂ ਦੇ ਜੋੜੇ ਬਣਾਓ।

1 1 0 1 0 1 0 1 1 . 0 0 1 1

- ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਦੇ ਸੱਜੇ ਪਾਸੇ ਜੋੜਾ ਪੂਰਾ ਹੋ ਰਿਹਾ ਹੈ ਕਿਉਂਕਿ ਇਸ ਵਿਚ ਚਾਰ ਬਿੱਟ ਪੂਰੇ ਹਨ, ਇਸ ਲਈ ਇਸ ਵਿਚ ਕੋਈ ਵਾਧੂ ਬਿੱਟ ਜੋੜਨ ਦੀ ਲੋੜ ਨਹੀਂ ਹੈ। ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਦੇ ਸਭ ਤੋਂ ਖੱਬੇ ਪਾਸੇ ਵੱਲ ਜੋੜੇ ਵਿੱਚ ਸਿਰਫ ਇੱਕ ਬਿੱਟ ਹੈ, ਇਸਲਈ ਇਸਨੂੰ ਚਾਰ ਬਿੱਟਾਂ ਦਾ ਇੱਕ ਪੂਰਾ ਜੋੜਾ ਬਣਾਉਣ ਲਈ ਸਾਨੂੰ ਇਸ ਦੇ ਖੱਬੇ ਪਾਸੇ ਤਿੰਨ ਜ਼ੀਰੋ ਹੋਰ ਜੋੜਨੇ ਪੈਣਗੇ।

0 0 0 1 1 0 1 0 1 0 1 1 . 0 0 1 1

- ਹੁਣ ਟੇਬਲ 1.5 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ:

$$(110101011.0011)_2=(1AB.3)_{16}$$

TEST YOURSELF

ਹੇਠ ਦਿੱਤੇ ਨੂੰ ਪੂਰਾ ਕਰੋ:

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾਡੈਸੀਮਲ
	10101		

1.3.3 ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮ (Octal to Other Number System):

ਬਾਈਨਰੀ ਅਤੇ ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਤਰ੍ਹਾਂ, ਔਕਟਲ ਨੰਬਰ ਨੂੰ ਵੀ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਆਉ ਇਹ ਸਮਝਣਾ ਸ਼ੁਰੂ ਕਰੀਏ ਕਿ ਇਹ ਬਦਲਾਵ ਕਿਵੇਂ ਹੁੰਦੇ ਹਨ:

1.3.3.1 ਔਕਟਲ ਤੋਂ ਡੈਸੀਮਲ ਵਿੱਚ ਬਦਲਾਵ (Octal to Decimal Conversion):

ਔਕਟਲ ਨੰਬਰ ਨੂੰ ਡੈਸੀਮਲ ਨੰਬਰ ਵਿੱਚ ਤਬਦੀਲ ਕਰਨ ਦਾ ਪ੍ਰੋਸੈਸ ਠੀਕ ਉਸੇ ਤਰ੍ਹਾਂ ਹੈ ਜਿਸ ਤਰ੍ਹਾਂ ਅਸੀਂ ਬਾਇਨਰੀ ਤੋਂ ਡੈਸੀਮਲ ਬਦਲਾਵ (conversion) ਵਿੱਚ ਕੀਤਾ ਸੀ। ਇਹ ਪ੍ਰੋਸੈਸ ਔਕਟਲ ਸੰਖਿਆ ਦੇ ਵਿਅਕਤੀਗਤ ਅੰਕਾਂ (individual digits) ਨੂੰ ਇਸਦੇ ਅਨੁਸਾਰੀ ਸਥਿਤੀ ਦੇ ਭਾਰਾਂ (corresponding positional weights) ਨਾਲ ਗੁਣਾ ਕਰਕੇ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। ਅੰਤ ਵਿੱਚ, ਅਸੀਂ ਇਹਨਾਂ ਸਾਰੀਆਂ ਗਣਨਾਵਾਂ (products) ਦਾ ਜੋੜ ਕਰਦੇ ਹਾਂ। ਆਉ ਇੱਕ ਉਦਾਹਰਨ ਨਾਲ ਸਮਝਦੇ ਹਾਂ ਕਿ ਔਕਟਲ ਤੋਂ ਡੈਸੀਮਲ ਵਿੱਚ ਬਦਲਾਵ ਕਿਵੇਂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ:

ਉਦਾਹਰਨ: $(62.25)_8$

62.25 ਦੇ ਹਰੇਕ ਅੰਕ ਨੂੰ ਉਹਨਾਂ ਦੇ ਅਨੁਸਾਰੀ ਸਥਿਤੀ ਦੇ ਭਾਰ (corresponding positional weight) ਨਾਲ ਗੁਣਾ ਕਰੋ, ਅਤੇ ਅੰਤ ਵਿੱਚ ਇਹਨਾਂ ਸਾਰੀਆਂ ਗਣਨਾਵਾਂ (products) ਦਾ ਜੋੜ ਕਰੋ:

$$
\begin{aligned}
(62.25)_8 &= (6\times8^1)+(2\times8^0)+(2\times8^{-1})+(5\times8^{-2}) \\
&= 48+2+\frac{(2\times1)}{8}+\frac{(5\times1)}{8^2} \\
&= 48+2+0.25+0.078125 \\
&= (50.328125)_{10}
\end{aligned}
$$

1.3.3.2 ਔਕਟਲ ਤੋਂ ਬਾਈਨਰੀ ਵਿੱਚ ਬਦਲਾਵ (Octal to Binary Conversion):

ਔਕਟਲ ਨੂੰ ਬਾਇਨਰੀ ਵਿੱਚ ਬਦਲਣ ਦਾ ਪ੍ਰੋਸੈਸ ਬਾਇਨਰੀ ਤੋਂ ਔਕਟਲ ਬਦਲਾਵ ਦਾ ਉਲਟ ਪ੍ਰੋਸੈਸ ਹੈ। ਇਸ ਪ੍ਰੋਸੈਸ ਵਿੱਚ ਸਾਨੂੰ ਟੇਬਲ 1.4 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹਰੇਕ ਔਕਟਲ ਅੰਕ ਦੇ ਤਿੰਨ-ਤਿੰਨ ਬਿੱਟਸ ਵਾਲੇ ਬਾਈਨਰੀ ਕੋਡ ਲਿਖਣੇ ਪੈਂਦੇ ਹਨ।

ਉਦਾਹਰਨ: $(62.25)_8$

ਟੇਬਲ 1.4 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹਰੇਕ ਔਕਟਲ ਅੰਕ 6, 2 ਅਤੇ 5 ਲਈ ਤਿੰਨ-ਬਿੱਟਸ ਵਾਲੇ ਜੋੜੇ ਦੇ ਬਾਈਨਰੀ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

$$(62.25)_8 = (110010.010101)_2$$

1.3.3.3 ਔਕਟਲ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ ਵਿੱਚ ਬਦਲਾਵ (Octal to Hexadecimal Conversion):

ਔਕਟਲ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ ਬਦਲਾਵ ਲਈ, ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਪਾਲਣਾ ਕਰੋ:

- ਟੇਬਲ 1.4 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਔਕਟਲ ਨੰਬਰ ਦੇ ਹਰੇਕ ਵਿਅਕਤੀਗਤ (individual) ਅੰਕ ਦੇ ਤਿੰਨ-ਤਿੰਨ ਬਿੱਟਸ ਵਾਲੇ ਬਾਈਨਰੀ ਕੋਡ ਲਿਖੋ।
- ਹੋਣ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਚਾਰ-ਚਾਰ ਬਿੱਟਸ ਦੇ ਜੋੜੇ ਬਣਾਓ। ਜੇਕਰ ਖੱਬੇ (left most) ਅਤੇ ਸੱਜੇ ਪਾਸੇ (right most) ਚਾਰ ਬਿੱਟਸ ਵਾਲੇ ਜੋੜ ਵਿੱਚ ਇੱਕ, ਦੋ ਜਾਂ ਤਿੰਨ ਬਿੱਟਸ ਘੱਟ ਹਨ, ਤਾਂ ਅਸੀਂ ਸਿਰੇ ਵਾਲੇ ਪਾਸਿਆਂ 'ਤੇ ਲੋੜੀਂਦੇ ਜ਼ੀਰੋ ਜੋੜ ਸਕਦੇ ਹਾਂ।
- ਹੁਣ, ਟੇਬਲ 1.5 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹਰੇਕ ਚਾਰ-ਚਾਰ ਬਿਟਸ ਵਾਲੇ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

ਆਓ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਦਲਾਵ (conversion) ਦੇ ਇਸ ਪ੍ਰੋਸੈਸ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ:

ਉਦਾਹਰਨ: $(62.25)_8$

- ਟੇਬਲ 1.4 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ, ਔਕਟਲ ਸੰਖਿਆਵਾਂ 6, 2 ਅਤੇ 5 ਦੇ ਤਿੰਨ-ਤਿੰਨ ਬਿੱਟਸ ਵਾਲੇ ਬਾਈਨਰੀ ਕੋਡ ਲਿਖੋ।

$$(62.25)_8=(110010.010101)_2$$

- ਹੁਣ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਜਾਂਦੇ ਹੋਏ ਚਾਰ-ਚਾਰ ਬਿੱਟਸ ਦੇ ਜੋੜੇ ਬਣਾਓ।

$$11\ \ 0010.0101\ \ 01$$

- ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਦੇ ਖੱਬੇ ਪਾਸੇ ਵਾਲੇ ਸਿਰੇ ਦੇ ਜੋੜੇ ਦੇ ਦੋ ਬਿੱਟਸ ਹਨ, ਅਤੇ ਸੱਜੇ ਪਾਸੇ ਵੱਲ ਸਿਰੇ ਦੇ, ਆਖਰੀ ਜੋੜੇ ਵਿੱਚ ਵੀ ਦੋ-ਬਿੱਟਸ ਹਨ। ਉਹਨਾਂ ਨੂੰ ਚਾਰ-ਚਾਰ ਬਿੱਟਸ ਦੇ ਪੂਰੇ ਜੋੜੇ ਬਣਾਉਣ ਲਈ ਸਿਰੇ ਦੇ ਪਾਸਿਆਂ 'ਤੇ ਜ਼ੀਰੋ ਜੋੜੇ।

$$0011\ \ 0010.0101\ \ 0100$$

- ਹੁਣ, ਟੇਬਲ 1.5 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹਰੇਕ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ

$$(00110010.01010100)_2=(32.54)_{16}$$

ਇਸ ਤਰ੍ਹਾਂ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਦੇ ਬਰਾਬਰ ਦਾ ਔਕਟਲ ਨੰਬਰ ਪ੍ਰਾਪਤ ਹੋ ਜਾਵੇਗਾ।

$$(32.54)_{16}=(62.25)_8$$

ਹੇਠ ਦਿੱਤੇ ਨੂੰ ਪੂਰਾ ਕਰੋ:

TEST YOURSELF

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾਡੈਸੀਮਲ
		37	

1.3.4 ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮ (Hexadecimal to Other Number System):

ਬਾਈਨਰੀ, ਡੈਸੀਮਲ ਅਤੇ ਔਕਟਲ ਨੰਬਰਾਂ ਦੀ ਤਰ੍ਹਾਂ, ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰਾਂ ਨੂੰ ਵੀ ਹੋਰ ਨੰਬਰ ਸਿਸਟਮਾਂ ਵਿੱਚ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਆਉ ਸਮਝਣਾ ਸ਼ੁਰੂ ਕਰਦੇ ਹਾਂ ਕਿ ਇਹ ਬਦਲਾਵ ਕਿਵੇਂ ਹੁੰਦੇ ਹਨ:

1.3.4.1 ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਡੈਸੀਮਲ ਬਦਲਾਵ (Hexadecimal to Decimal Conversion):

ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਨੂੰ ਡੈਸੀਮਲ ਨੰਬਰ ਵਿੱਚ ਬਦਲਣ ਦਾ ਪ੍ਰੋਸੈਸ ਵੀ ਠੀਕ ਉਸੇ ਤਰ੍ਹਾਂ ਹੀ ਹੈ ਜਿਵੇਂ ਅਸੀਂ ਬਾਈਨਰੀ ਤੋਂ ਡੈਸੀਮਲ ਬਦਲਾਵ ਵਿਚ ਕੀਤਾ ਸੀ। ਇਹ ਪ੍ਰੋਸੈਸ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਦੇ ਵਿਅਕਤੀਗਤ (individual) ਅੰਕਾਂ ਨੂੰ ਇਸਦੇ ਅਨੁਸਾਰ ਸਥਿਤੀ ਦੇ ਭਾਰਾਂ (corresponding positional weights) ਨਾਲ ਗੁਣਾ ਕਰਕੇ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। ਅੰਤ ਵਿੱਚ, ਅਸੀਂ ਇਹਨਾਂ ਗਣਨਾਵਾਂ (products) ਨੂੰ ਜੋੜ ਦਿੰਦੇ ਹਾਂ। ਆਉ ਇੱਕ ਉਦਾਹਰਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸਮਝਦੇ ਹਾਂ ਕਿ ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਡੈਸੀਮਲ ਕਨਵਰਜ਼ਨ ਕਿਵੇਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ:

ਉਦਾਹਰਨ:

ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ **62A.25** ਦੇ ਹਰੇਕ ਅੰਕ ਨੂੰ ਉਹਨਾਂ ਦੇ ਅਨੁਸਾਰੀ ਸਥਿਤੀ ਦੇ ਭਾਰਾਂ (corresponding positional weights) ਨਾਲ ਗੁਣਾ ਕਰੋ, ਅਤੇ ਅੰਤ ਵਿੱਚ ਇਹਨਾਂ ਸਾਰੀਆਂ ਗਣਨਾਵਾਂ (products) ਦਾ ਜੋੜ ਕਰੋ:

$$\begin{aligned}(62A.25)_{16} &= (6\times16^2)+(2\times16^1)+(A\times16^0)+(2\times16^{-1})+(5\times16^{-2})\\ &= (6\times256)+(2\times16)+(10\times1)+(2\times16^{-1})+(5\times16^{-2})\\ &= 1536+32+10+(2\times\tfrac{1}{16})+(5\times\tfrac{1}{256})\\ &= 1578+0.125+0.1953125\\ &= (1578.14453125)_{10}\end{aligned}$$

1.3.4.2 ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਬਾਈਨਰੀ ਬਦਲਾਵ (Hexadecimal to Binary Conversion) :

ਹੈਕਸਾਡੈਸੀਮਲ ਨੂੰ ਬਾਇਨਰੀ ਵਿੱਚ ਬਦਲਣ ਦਾ ਪ੍ਰੋਸੈਸ ਬਾਇਨਰੀ ਤੋਂ ਹੈਕਸਾਡੈਸੀਮਲ ਵਿਚ ਬਦਲਾਵ ਦਾ ਉਲਟ ਪ੍ਰੋਸੈਸ ਹੈ। ਇਸ ਪ੍ਰੋਸੈਸ ਵਿਚ ਸਾਨੂੰ ਹਰੇਕ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕ ਦਾ ਚਾਰ-ਚਾਰ ਬਿੱਟਸ ਵਾਲਾ ਬਾਈਨਰੀ ਕੋਡ ਲਿਖਣਾ ਹੋਵੇਗਾ।

ਉਦਾਹਰਨ: $(62A.25)_{16}$

ਟੇਬਲ 1.5 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੈਕਸਾਡੈਸੀਮਲ ਚਿੰਨ੍ਹਾਂ 6, 2, A ਅਤੇ 5 ਲਈ ਚਾਰ-ਚਾਰ ਬਿੱਟਸ ਵਾਲੇ ਬਾਈਨਰੀ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

$$(62A.25)_{16}=(0110\ 0010\ 1010.0010\ 0101)_2$$

1.3.4.3 ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਔਕਟਲ ਬਦਲਾਵ (Hexadecimal to Octal Conversion) :

ਹੈਕਸਾਡੈਸੀਮਲ ਤੋਂ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿਚ ਬਦਲਾਵ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈੱਪਸ ਦੀ ਪਾਲਣਾ ਕਰੋ:

- ਟੇਬਲ 1.5 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਦੇ ਹਰੇਕ ਵਿਅਕਤੀਗਤ (individual) ਅੰਕ ਲਈ ਚਾਰ-ਚਾਰ ਬਿਟਸ ਵਾਲ ਬਾਈਨਰੀ ਕੋਡ ਲਿਖੋ।
- ਹੁਣ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਜਾਂਦੇ ਹੋਏ ਤਿੰਨ-ਤਿੰਨ ਬਿੱਟਸ ਦੇ ਜੋੜੇ ਬਣਾਓ। ਜੇਕਰ ਅਖੀਰਲੇ ਖੱਬੇ (left most) ਅਤੇ ਅਖੀਰਲੇ ਸੱਜੇ ਪਾਸੇ (right most) ਤਿਨ ਬਿੱਟਸ ਵਾਲੇ ਜੋੜੇ ਵਿੱਚ ਇੱਕ ਜਾਂ ਦੋ ਬਿੱਟਸ ਘੱਟ ਹਨ, ਤਾਂ ਅਸੀਂ ਸਿਰੇ ਵਾਲੇ ਪਾਸਿਆਂ 'ਤੇ ਲੋੜੀਂਦੇ ਜ਼ੀਰੋ ਜੋੜ ਸਕਦੇ ਹਾਂ।
- ਹੁਣ, ਟੇਬਲ 1.4 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹਰੇਕ ਤਿੰਨ-ਤਿੰਨ ਬਿਟਸ ਵਾਲੇ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਔਕਟਲ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

ਆਓ ਇਕ ਉਦਾਹਰਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਬਦਲਾਵ ਦੇ ਇਸ ਪ੍ਰੋਸੈਸ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ:

ਉਦਾਹਰਨ: $(62A.25)_{16}$

- ਟੇਬਲ 1.5 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੈਕਸਾਡੈਸੀਮਲ ਅੰਕ 6, 2, A, ਅਤੇ 5 ਲਈ ਚਾਰ-ਚਾਰ ਬਿੱਟਸ ਵਾਲੇ ਬਾਈਨਰੀ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

$$(62A.25)_{16}=(0110\ 0010\ 1010.0010\ 0101)_2$$

- ਹੁਣ, ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਤੋਂ ਬਾਹਰ ਵੱਲ ਜਾਂਦੇ ਹੋਏ ਤਿੰਨ-ਤਿੰਨ ਬਿੱਟਸ ਦੇ ਜੋੜੇ ਬਣਾਓ।

011 000 101 010 . 001 001 01

- ਬਾਈਨਰੀ ਪੁਆਇੰਟ ਦੇ ਸਿਰੇ ਦੇ ਸੱਜੇ ਪਾਸੇ ਵਾਲੇ ਜੋੜੇ ਵਿੱਚ ਤਿੰਨ ਬਿੱਟਾਂ ਦੀ ਜੋੜੀ ਪੂਰੀ ਕਰਨ ਲਈ ਇੱਕ ਬਿੱਟ ਘੱਟ ਹੈ, ਇਸਲਈ ਸਾਨੂੰ ਅਖੀਰਲੇ ਸੱਜੇ (right most) ਪਾਸੇ ਇੱਕ ਵਾਧੂ ਜ਼ੀਰੋ ਜੋੜਨ ਦੀ ਲੋੜ ਹੈ। ਬਿੱਟਸ ਦਾ ਅਖੀਰਲੇ ਖੱਬੇ (left most) ਪਾਸੇ ਵਾਲੇ ਜੋੜੇ ਵਿੱਚ ਬਿੱਟਸ ਦੀ ਗਿਣਤੀ ਪੂਰੀ ਹੈ, ਇਸਲਈ ਵਾਧੂ ਜ਼ੀਰੋ ਜੋੜਨ ਦੀ ਕੋਈ ਲੋੜ ਨਹੀਂ ਹੈ।

011 000 101 010 . 001 001 010

- ਹੁਣ, ਟੇਬਲ 1.4 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹਰੇਕ ਤਿੰਨ-ਤਿੰਨ ਬਿਟਸ ਵਾਲੇ ਜੋੜੇ ਦੇ ਅਨੁਸਾਰੀ ਔਕਟਲ ਅੰਕਾਂ ਨੂੰ ਲਿਖੋ।

$$(011000101010.001001010)_2=(3052.112)_8$$

ਇਸ ਤਰ੍ਹਾਂ ਨੂੰ ਸਾਨੂੰ ਔਕਟਲ ਨੰਬਰ ਦੇ ਬਰਾਬਰ ਦਾ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਪ੍ਰਾਪਤ ਹੋ ਜਾਵੇਗਾ।

$$(3052.112)_8=(62A.25)_{16}$$

TEST YOURSELF

ਹੇਠ ਦਿੱਤੇ ਨੂੰ ਪੂਰਾ ਕਰੋ :-

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾ ਡੈਸੀਮਲ
			A5

ਯਾਦ ਰੱਖਣ ਯੋਗ ਗੱਲਾਂ

1. ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਦੋ ਬੁਨਿਆਦੀ ਕਿਸਮਾਂ ਹਨ: ਨਾਨ-ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਅਤੇ ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ।
2. ਕਿਸੇ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਅੰਕਾਂ ਦੀ ਕੁੱਲ ਸੰਖਿਆ ਨੂੰ ਇਸਦਾ ਅਧਾਰ ਜਾਂ ਰੇਡੀਕਸ ਮੁੱਲ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
3. ਕੰਪਿਊਟਰਾਂ ਸਿਸਟਮਾਂ ਦੁਆਰਾ ਚਾਰ ਕਿਸਮਾਂ ਦੇ ਨੰਬਰ ਸਿਸਟਮਾਂ ਨੂੰ ਸਪੋਰਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ: ਡੈਸੀਮਲ, ਬਾਈਨਰੀ, ਔਕਟਲ ਅਤੇ ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ।
4. ਇੱਕ ਨੰਬਰ ਸਿਸਟਮ ਜਿਸ ਵਿੱਚ 8 ਵੱਖ-ਵੱਖ ਚਿੰਨ੍ਹ 0, 1, 2, 3, 4, 5, 6 ਅਤੇ 7 ਹੁੰਦੇ ਹਨ, ਨੂੰ ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
5. ਅਧਾਰ-10 (ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ) ਵਿਚ 0-9 ਅੰਕਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਰੋਜ਼ਾਨਾ ਗਣਿਤ ਵਿੱਚ ਵਰਤੀਆ ਜਾਣ ਵਾਲਾ ਸਭ ਤੋਂ ਆਮ ਨੰਬਰ ਸਿਸਟਮ ਹੈ।
6. ਬੇਸ-2 (ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ) ਵਿਚ ਅੰਕ 0 ਅਤੇ 1 ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਨੰਬਰ ਸਿਸਟਮ ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਅਤੇ ਡਿਜੀਟਲ ਇਲੈਕਟ੍ਰੋਨਿਕਸ ਵਿੱਚ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਇੱਕ ਬੁਨਿਆਦੀ ਨੰਬਰ ਸਿਸਟਮ ਹੈ।
7. ਬੇਸ-8 (ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ) ਵਿਚ 0,1,2,3,4,5,6 ਅਤੇ 7 ਅੰਕਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਨੰਬਰ ਸਿਸਟਮ ਕਈ ਵਾਰ ਕੰਪਿਊਟਿੰਗ ਵਿੱਚ ਬਾਈਨਰੀ ਲਈ ਸ਼ਾਰਟਹੈਂਡ ਵਜੋਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
8. ਬੇਸ-16 (ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ) ਵਿਚ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ਅਤੇ ਅੱਖਰ A, B, C, D, E ਅਤੇ F ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਨੰਬਰ ਸਿਸਟਮ ਆਮ ਤੌਰ 'ਤੇ ਮੈਮਰੀ ਐਡਰੈਸਿੰਗ ਅਤੇ ਕਲਰ ਕੋਡ ਲਈ ਕੰਪਿਊਟਿੰਗ ਵਿੱਚ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।

ਅਭਿਆਸ

ੳ. ਬਹੁ ਪਸੰਦੀ ਪ੍ਰਸ਼ਨ:

1. ਹੈਕਸਾਡੈਸੀਮਲ ਸਿਸਟਮ ਵਿੱਚ ਹੇਠ ਲਿਖੀਆਂ ਵਿੱਚੋਂ ਕਿਹੜੀ ਸੰਖਿਆ ਸਭ ਤੋਂ ਛੋਟੀ ਹੈ ?

   ੳ) 0 ਅ) 1 ਇ) A ਸ) F

2. ਡੈਸੀਮਲ ਨੰਬਰ 15 ਦਾ ਬਾਈਨਰੀ ਨੰਬਰ ਕੀ ਹੈ ?

   ੳ) 0111 ਅ) 1111 ਇ) 0101 ਸ) 0011

3. ਰੋਜ਼ਾਨਾ ਗਣਿਤ ਵਿੱਚ ਕਿਹੜੀ ਨੰਬਰ ਸਿਸਟਮ ਸਭ ਤੋਂ ਵੱਧ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ?

   ੳ) ਬਾਈਨਰੀ ਅ) ਡੈਸੀਮਲ ਇ) ਹੈਕਸਾਡੈਸੀਮਲ ਸ) ਔਕਟਸ

4. ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਦਾ ਬੇਸ/ਰੇਡੀਕਸ ਕੀ ਹੈ ?

   ੳ) 2 ਅ) 5 ਇ) 8 ਸ) 16

5. $(38CB)_{16}$ ਕਿਹੜੀ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਉਦਾਹਰਨ ਹੈ ?

   ੳ) ਬਾਈਨਰੀ ਅ) ਡੈਸੀਮਲ ਇ) ਹੈਸਾਡੈਸੀਮਲ ਸ) ਔਕਟਲ

ਅ. ਹੇਠ ਲਿਖਿਆਂ ਵਿੱਚੋਂ ਸਹੀ ਜਾਂ ਗਲਤ ਚੁਣੋ:

1. ਬਾਈਨਰੀ ਨੰਬਰ 1111 ਡੈਸੀਮਲ ਨੰਬਰ 15 ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।
2. ਹੈਕਸਾਡੈਸੀਮਲ ਸਿਸਟਮ 2 ਨੂੰ ਇਸਦੇ ਅਧਾਰ/ਰੇਡੀਕਸ ਵਜੋਂ ਵਰਤਦਾ ਹੈ।
3. ਔਕਟਲ ਸ਼ਬਦ ਲਾਤੀਨੀ ਸ਼ਬਦ Oct ਤੋਂ ਆਇਆ ਹੈ ਜਿਸਦਾ ਅਰਥ 8 ਹੈ।
4. ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਸਿਰਫ 0 ਅਤੇ 1 ਅੰਕਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕਿਸੇ ਵੀ ਸੰਖਿਆ ਨੂੰ ਦਰਸਾ ਸਕਦਾ ਹੈ।
5. ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ ਅੰਕ '9' ਵੈਧ (valid) ਅੰਕ ਹੈ।
6. ਹੈਕਸਾਡੈਸੀਮਲ ਸਿਸਟਮ ਵਿੱਚ, ਅੱਖਰ 'B' ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਵਿੱਚ 12 ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।
7. ਰੋਮਨ ਨੰਬਰ ਸਿਸਟਮ ਪੁਜ਼ੀਸ਼ਨਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਇੱਕ ਉਦਾਹਰਨ ਹੈ।

ਏ. ਛੋਟੇ ਉਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

1. ਨੰਬਰ ਸਿਸਟਮ ਕੀ ਹੈ ? ਨੰਬਰ ਸਿਸਟਮਾਂ ਦੀਆਂ ਮੂਲ ਸ਼੍ਰੇਣੀਆਂ ਦੇ ਨਾਂ ਲਿਖੋ ?
2. ਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਵਿਆਖਿਆ ਕਰੋ ?
3. ਬਾਈਨਰੀ ਨੰਬਰ ਸਿਸਟਮ ਕੀ ਹੈ ? ਇਸ ਨੰਬਰ ਪ੍ਰਣਾਲੀ ਦਾ ਖੋਜੀ ਕੌਣ ਹੈ ?
4. ਔਕਟਲ ਨੰਬਰ ਸਿਸਟਮ ਦੀ ਵਿਆਖਿਆ ਕਰੋ ?
5. ਹੈਕਸਾਡੈਸੀਮਲ ਨੰਬਰ ਸਿਸਟਮ ਕੀ ਹੈ ?

ਸ. ਨੰਬਰ ਸਿਸਟਮ ਦੇ ਬਦਲਾਵ ਨੂੰ ਪੂਰਾ ਕਰੋ:

1. $(25)_{10}$ = ( )$_2$ = ( )$_8$ = ( )$_{16}$
2. $(10111)_2$ = ( )$_{10}$ = ( )$_8$ = ( )$_{16}$
3. $(47)_8$ = ( )$_{10}$ = ( )$_2$ = ( )$_{16}$
4. $(A4)_{16}$ = ( )$_{10}$ = ( )$_2$ = ( )$_8$

ਐਕਟੀਵਿਟੀ

1.1 ਹੇਠ ਦਿੱਤੇ ਟੇਬਲ ਨੂੰ ਪੂਰਾ ਕਰੋ:

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾਡੈਸੀਮਲ
0	0000	0	0
1	0001	---	1
2	0010	2	---
3	---	3	3
4	0100	---	4
5	0101	---	5
6	0110	6	---
7	---	7	
8	1000	10	8
9	---	11	9
10	1010	12	---
11	1011	---	B
12	---	14	C
13	1101	15	---
14	1110	---	E
15	1111	17	---

1.2 ਹੇਠਾਂ ਦਿੱਤੇ ਟੇਬਲ ਨੂੰ ਪੂਰਾ ਕਰੋ :

ਡੈਸੀਮਲ	ਬਾਈਨਰੀ	ਔਕਟਲ	ਹੈਕਸਾਡੈਸੀਮਲ
23			
	111101		
		27	
			38C
89			
	10001		
			8F
		37	
			9A
99			

ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਿਦਿਆਰਥੀਆਂ ਨੂੰ 21ਵੀਂ ਸਦੀ ਦੇ ਕਰੀਅਰ ਵਿੱਚ ਕਾਮਯਾਬ ਹੋਣ ਲਈ ਲੋੜੀਂਦੇ ਹੁਨਰ ਸਿਖਾਉਂਦੀ ਹੈ। ਪ੍ਰੋਗਰਾਮਿੰਗ ਨਾਲ ਵਿਦਿਆਰਥੀ ਨਾ ਸਿਰਫ਼ ਤਕਨਾਲੋਜੀ ਅਤੇ ਕੋਡਿੰਗ ਬਾਰੇ ਸਿੱਖਦੇ ਹਨ, ਸਗੋਂ ਉਹ ਆਲੋਚਨਾਤਮਕ ਸੋਚ, ਸਮੱਸਿਆ-ਹੱਲ ਕਰਨ, ਸਹਿਯੋਗ ਕਰਨ, ਅਤੇ ਹੋਰ ਬਹੁਤ ਕੁੱਝ ਗੱਲਾਂ ਬਾਰੇ ਵੀ ਸਿੱਖ ਰਹੇ ਹੁੰਦੇ ਹਨ। ਕੋਡਿੰਗ ਇੱਕ ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਇਹ ਦੱਸਦੀ ਹੈ ਕਿ ਕਿਵੇਂ ਕੰਮ ਨੂੰ ਕਰਨਾ ਹੈ।

ਪਾਠ ਦਾ ਉਦੇਸ਼

- ਪ੍ਰੋਗਰਾਮਿੰਗ ਦੀਆਂ ਬੁਨਿਆਦੀ ਗੱਲਾਂ, ਪਾਈਥਨ ਅਤੇ ਇਸਦੇ IDE (IDLE) ਨਾਲ ਜਾਣ-ਪਛਾਣ, ਪਾਈਥਨ ਵਿੱਚ ਕੋਡ ਐਗਰਜ਼ੀਕਿਊਸ਼ਨ ਦੇ ਮੋਡ: ਇੰਟਰਐਕਟਿਵ ਅਤੇ ਸਕ੍ਰਿਪਟ ਮੋਡ, IDLE ਨਾਲ ਕੋਡ ਲਿਖਣਾ ਅਤੇ ਲਾਗੂ ਕਰਨਾ।
- ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਦੇ ਬੁਨਿਆਦੀ ਸੰਕਲਪ: ਕਰੈਕਟਰ ਸੈੱਟ, ਟੋਕਨ (ਕੀਵਰਡਸ, ਆਈਡੈਂਟੀਫਾਇਰ, ਲਿਟਰਲ, ਓਪਰੇਟਰ ਅਤੇ ਡੀਲੀਮੀਟਰ), ਵੇਰੀਏਬਲ, ਕਮੈਂਟਸ, ਮਲਟੀਪਲ-ਲਾਈਨ ਸਟੇਟਮੈਂਟਸ, ਪਾਈਥਨ ਵਿੱਚ ਕੁਟੇਸ਼ਨ-ਮਾਰਕਸ, ਪਾਈਥਨ ਵਿੱਚ ਖਾਲੀ ਲਾਈਨਾਂ, ਪਾਈਥਨ ਵਿੱਚ ਆਉਪੁੱਟ ਅਤੇ ਇਨਪੁੱਟ ਸਟੇਟਮੈਂਟਸ।

ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ✓ ਵਿਦਿਆਰਥੀ ਪ੍ਰੋਗਰਾਮਿੰਗ ਦੇ ਮੂਲ ਤੱਥਾਂ ਨੂੰ ਸਮਝਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀ ਪਾਈਥਨ ਵਿੱਚ ਕੋਡ ਲਿਖਣਾ ਅਤੇ ਚਲਾਉਣਾ ਸਿੱਖ ਜਾਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀ ਟੋਕਨਜ਼ ਦੇ ਰੂਪ ਵਿੱਚ ਪ੍ਰੋਗਰਾਮਾਂ ਦੇ ਵੱਖ-ਵੱਖ ਤੱਤਾਂ ਨੂੰ ਸਮਝਣਗੇ।
- ✓ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਕਮੈਂਟਸ ਦੀ ਭੂਮਿਕਾ ਨੂੰ ਸਮਝਣਯੋਗ ਬਨਣਗੇ।

02 ਪਾਠ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਸਬੰਧੀ ਮੂਲ ਧਾਰਨਾਵਾਂ

ਕੰਪਿਊਟਰ ਦੀ ਵਰਤੋਂ ਰੋਜ਼ਾਨਾਂ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਹੱਲ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸ ਲਈ ਸਮੱਸਿਆਵਾਂ ਹੱਲ ਕਰਨ ਦਾ ਹੁਨਰ ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਦੇ ਵਿਦਿਆਰਥੀ ਨੂੰ ਪਤਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਇੱਥੇ ਇਹ ਦੱਸਣਾ ਉਚਿੱਤ ਹੋਵੇਗਾ ਕਿ ਕੰਪਿਊਟਰ ਖੁਦ ਕਿਸੇ ਸਮੱਸਿਆ ਦਾ ਹੱਲ ਨਹੀਂ ਕਰ ਸਕਦੇ। ਸਮੱਸਿਆ ਨੂੰ ਹੱਲ ਕਰਨ ਲਈ ਸਾਡੇ ਦੁਆਰਾ ਕੰਪਿਊਟਰ ਨੂੰ ਸਹੀ ਕਦਮ-ਦਰ-ਕਦਮ ਨਿਰਦੇਸ਼ ਦਿੱਤੇ ਜਾਣੇ ਜ਼ਰੂਰੀ ਹੁੰਦੇ ਹਨ। ਇਸ ਤਰ੍ਹਾਂ, ਕਿਸੇ ਸਮੱਸਿਆ ਨੂੰ ਹੱਲ ਕਰਨ ਵਿੱਚ ਕੰਪਿਊਟਰ ਦੀ ਸਫਲਤਾ ਇਸ ਗੱਲ 'ਤੇ ਨਿਰਭਰ ਕਰਦੀ ਹੈ ਕਿ ਅਸੀਂ ਸਮੱਸਿਆ ਨੂੰ ਕਿੰਨ੍ਹੇ ਸਹੀ ਢੰਗ ਨਾਲ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦੇ ਹਾਂ। ਅਜਿਹਾ ਕਰਨ ਲਈ ਅਸੀਂ ਇੱਕ ਹੱਲ (ਐਲਗੋਰਿਥਮ) ਤਿਆਰ ਕਰਦੇ ਹਾਂ ਅਤੇ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਹੱਲ (ਪ੍ਰੋਗਰਾਮ) ਨੂੰ ਲਾਗੂ ਕਰਦੇ ਹਾਂ। ਸਧਾਰਣ ਸ਼ਬਦਾਂ ਵਿਚ ਕਿਹਾ ਜਾ ਸਕਦਾ ਹੈ ਕਿ ਕਿਸੇ ਸਮੱਸਿਆ ਨੂੰ ਕੰਪਿਊਟਰ ਦੀ ਮਦਦ ਨਾਲ ਹੱਲ ਕਰਨਾ ਇਕ ਅਜਿਹਾ ਪ੍ਰੋਸੈਸ ਹੈ ਜਿਸ ਵਿਚ ਸਮੱਸਿਆ ਦੀ ਪਛਾਣ ਕਰਕੇ, ਪਛਾਣੀ ਗਈ ਸਮੱਸਿਆ ਲਈ ਐਲਗੋਰਿਥਮ ਤਿਆਰ ਕਰਨਾ ਅਤੇ ਅੰਤ ਵਿੱਚ ਐਲਗੋਰਿਥਮ ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਡਿਵੈਲਪ ਕਰਨਾ ਹੁੰਦਾ ਹੈ।

ਐਲਗੋਰਿਥਮ (Algorithm) ਇੱਕ ਚੰਗੀ ਤਰ੍ਹਾਂ ਪਰਿਭਾਸ਼ਿਤ ਸਮੱਸਿਆ ਨੂੰ ਹੱਲ ਕਰਨ ਦੀ ਇੱਕ ਕਦਮ-ਦਰ-ਕਦਮ ਪ੍ਰਕਿਰਿਆ ਹੈ। ਐਲਗੋਰਿਥਮ ਦੀ ਵਰਤੋਂ ਪ੍ਰੋਗਰਾਮ ਦੇ ਲਾਗੂ ਕਰਨ (implementation) ਨੂੰ ਸਰਲ ਬਣਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਇੱਕ ਕੰਪਿਊਟਰ **ਪ੍ਰੋਗਰਾਮ** ਅਸਲ ਵਿੱਚ ਨਿਰਦੇਸ਼ਾਂ (instructions) ਦਾ ਇੱਕ ਸਮੂਹ ਹੁੰਦਾ ਹੈ ਜੋ ਕੰਪਿਊਟਰ ਦੁਆਰਾ ਕਿਸੇ ਕੰਮ ਨੂੰ ਕਰਨ ਜਾਂ ਕਿਸੇ ਸਮੱਸਿਆ ਨੂੰ ਹੱਲ ਕਰਨ ਲਈ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਲਿਖਣ ਦੀ ਪ੍ਰਕਿਰਿਆ ਨੂੰ **ਪ੍ਰੋਗਰਾਮਿੰਗ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਲਿਖਣ ਵਾਲੇ ਵਿਅਕਤੀ ਨੂੰ **ਪ੍ਰੋਗਰਾਮਰ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰਨ ਬਾਰੇ ਸਿੱਖਣਾ ਕਿਸੇ ਵੀ ਹੁਨਰ ਨੂੰ ਸਿੱਖਣ ਵਾਂਗ ਹੀ ਹੁੰਦਾ ਹੈ। ਪਹਿਲਾਂ ਸਾਨੂੰ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਚੰਗੀ ਤਰ੍ਹਾਂ ਸਮਝਣਾ ਚਾਹੀਦਾ ਹੈ ਅਤੇ ਫਿਰ ਤਰਕਪੂਰਨ ਢੰਗ ਨਾਲ ਉਹਨਾਂ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਹੱਲ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ। ਜਦੋਂ ਪ੍ਰੋਗਰਾਮਰ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖਦਾ ਹੈ, ਤਾਂ ਉਹ ਇੱਕ ਖਾਸ ਪ੍ਰੋਸੈਸ ਵਿੱਚੋਂ ਲੰਘਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਤਿਆਰ ਕਰਨ ਦੇ ਇਸ ਪ੍ਰੋਸੈਸ ਨੂੰ **ਪ੍ਰੋਗਰਾਮਿੰਗ ਪ੍ਰੋਸੈਸ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। **ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ** ਦੀ ਵਰਤੋਂ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਨਿਰਦੇਸ਼ (instructions) ਲਿਖਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਪ੍ਰੋਗਰਾਮਰ ਕਿਸੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਤਿਆਰ ਕਰਨ ਲਈ ਉਪਲਬਧ ਸੈਂਕੜੇ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚੋਂ ਕਿਸੇ ਵੀ ਭਾਸ਼ਾ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦਾ ਹੈ। ਪਾਈਥਨ ਉਨ੍ਹਾਂ ਵਿੱਚੋਂ ਇੱਕ ਹੈ।

2.1 ਪਾਈਥਨ ਨਾਲ ਜਾਣ ਪਛਾਣ (INTRODUCTION TO PYTHON)

ਪਾਈਥਨ ਇੱਕ ਪ੍ਰਸਿੱਧ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਹੈ। ਇਹ ਗਾਈਡੋ ਵੈਨ ਰੋਸਮ (Guido Van Rossum) ਦੁਆਰਾ ਵਿਕਸਤ ਕੀਤੀ ਗਈ ਸੀ। ਇਸਨੂੰ 1991 ਵਿੱਚ ਪੇਸ਼ ਕੀਤਾ ਗਿਆ। ਇਸ ਤੋਂ ਬਾਅਦ ਇਸਨੂੰ ਪਾਈਥਨ ਸਾਫਟਵੇਅਰ ਫਾਊਂਡੇਸ਼ਨ ਦੁਆਰਾ ਵਿਕਸਤ ਕੀਤਾ ਗਿਆ। ਇਹ ਇੱਕ ਆਮ-ਉਦੇਸ਼ (General-Purpose) ਵਾਲੀ ਹਾਈ ਲੇਵਲ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਹੈ। ਇਸ ਵਿਚ ਆਬਜੈਕਟ ਓਰੀਐਂਟਿਡ (Object Oriented) ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਿਧੀ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਐਪਲੀਕੇਸ਼ਨਾਂ ਤਿਆਰ ਕੀਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਇਹ ਭਾਸ਼ਾ ਡਾਟਾ ਸਾਇੰਸ ਲਈ ਵਧੇਰੇ ਪ੍ਰਸਿੱਧ ਹੋ ਰਹੀ ਹੈ।

ਪਾਈਥਨ ਇੱਕ ਡਾਇਨਾਮਿੱਕ (Dynamic), ਇੰਟਰਪ੍ਰੈਟਿਡ (ਬਾਈਟਕੋਡ (Bytecode)-ਕੰਪਾਈਲਡ) ਭਾਸ਼ਾ ਹੈ। ਪਾਈਥਨ ਸੋਰਸ ਕੋਡ ਵਿੱਚ ਵੇਰੀਏਬਲਾਂ, ਪੈਰਾਮੀਟਰਾਂ, ਫੰਕਸ਼ਨਾਂ, ਜਾਂ ਮੈਥਡਜ਼ ਦੀ ਕੋਈ ਟਾਈਪ ਡਿਕਲੇਰੇਸ਼ਨ ਨਹੀਂ ਕੀਤੀ ਜਾਂਦੀ। ਇਸ ਭਾਸ਼ਾ ਵਿਚ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਕੋਡ ਛੋਟਾ ਅਤੇ ਲਚਕਦਾਰ (Flexible) ਹੁੰਦਾ ਹੈ। ਪਾਈਥਨ ਰਨਟਾਈਮ 'ਤੇ ਸਾਰੇ ਮੁੱਲਾਂ ਦੀਆਂ ਕਿਸਮਾਂ ਨੂੰ ਟਰੈਕ ਕਰਦਾ ਹੈ।

ਪਾਈਥਨ ਭਾਸ਼ਾ ਪ੍ਰੋਸੀਜਰਲ (Procedural), ਆਬਜੈਕਟ ਓਰੀਐਂਟਿਡ (Object Oriented) ਅਤੇ ਫੰਕਸ਼ਨਲ (Functional) ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਸਮੇਤ ਕਈ ਪ੍ਰੋਗਰਾਮਿੰਗ ਪੈਰਾਡਾਈਮਜ਼ (Paradigms) ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ। ਪਾਈਥਨ ਦਾ ਡਿਜ਼ਾਈਨ ਫ਼ਲਸਫ਼ਾ (Philosophy) ਇੰਡੈਂਟੇਸ਼ਨ (Indentation) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸੋਰਸ ਕੋਡ ਦੀ ਪੜ੍ਹਨਯੋਗਤਾ (Code Readability) 'ਤੇ ਜ਼ੋਰ ਦਿੰਦਾ ਹੈ।

ਅੱਜ ਦੇ ਸਮੇਂ ਪਾਈਥਨ ਦੀ ਮੰਗ ਬਹੁਤ ਜ਼ਿਆਦਾ ਹੈ। ਸਾਰੀਆਂ ਵੱਡੀਆਂ ਕੰਪਨੀਆਂ ਵਧੀਆ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਰਾਂ ਦੀ ਭਾਲ ਕਰ ਰਹੀਆਂ ਹਨ। ਡਾਟਾ ਸਾਇੰਸ, **AI** (ਆਰਟੀਫੀਸ਼ੀਅਲ ਇੰਟੈਲੀਜੈਂਸ), ਅਤੇ **ML** (ਮਸ਼ੀਨ ਲਰਨਿੰਗ) ਤਕਨਾਲੋਜੀਆਂ ਨਾਲ ਕੰਮ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮਰ ਪਾਈਥਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੈਬਸਾਈਟਾਂ, ਸਾਫਟਵੇਅਰ ਅਤੇ ਐਪਲੀਕੇਸ਼ਨਾਂ ਨੂੰ ਵਿਕਸਤ ਕਰ ਸਕਦੇ ਹਨ। ਪਾਈਥਨ ਨੂੰ ਲਗਾਤਾਰ ਵਿਸ਼ਵ ਦੀਆਂ ਸਭ ਤੋਂ ਪ੍ਰਸਿੱਧ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚੋਂ ਇੱਕ ਵਿਸ਼ੇਸ਼ ਦਰਜਾ ਮਿਲਦਾ ਆ ਰਿਹਾ ਹੈ।

2.2 ਪਾਇਥਨ ਨਾਲ ਜਾਣ-ਪਛਾਣ (INTRODUCTION TO PYTHON IDE)

IDE (**ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਿਵੈਲਪਮੈਂਟ ਇਨਵਾਇਰਨਮੈਂਟ**) ਇੱਕ ਸਾਫਟਵੇਅਰ ਹੁੰਦਾ ਹੈ ਜੋ ਸਾਫਟਵੇਅਰ ਡਿਵੈਪਲਮੈਂਟ ਲਈ ਸਮਰਪਿਤ (dedicated) ਹੁੰਦਾ ਹੈ। ਇਹ ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮਰਾਂ ਨੂੰ ਵਿਆਪਕ ਪੱਧਰ ਤੇ ਸਾਫਟਵੇਅਰ ਡਿਵੈਪਲਮੈਂਟ ਲਈ ਕਈ ਯੋਗਤਾਵਾਂ (abilities) ਪੇਸ਼ ਕਰਦਾ ਹੈ। IDEs ਖਾਸ ਤੌਰ 'ਤੇ ਸਾਫਟਵੇਅਰ ਡਿਵੈਲਪਮੈਂਟ ਲਈ ਤਿਆਰ ਕੀਤੇ ਗਏ ਕਈ ਟੂਲਜ਼ (tools) ਨੂੰ ਇੰਟੀਗ੍ਰੇਟ ਕਰਦੇ ਹਨ। ਇਹਨਾਂ ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਹੇਠਾਂ ਦਿਤੇ ਟੂਲਜ਼ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ:

- **ਸੋਰਸ ਕੋਡ ਨੂੰ ਹੈਂਡਲ ਕਰਨ ਲਈ ਇੱਕ ਐਡੀਟਰ (Editor):** ਆਮ ਤੌਰ 'ਤੇ, ਇਹ ਐਡੀਟਰ ਸਿੰਟੈਕਸ ਹਾਈਲਾਈਟਿੰਗ (Syntax Highlighting) ਅਤੇ ਆਟੋ ਕੋਡ-ਕੰਪਲੀਸ਼ਨ (Auto Code-Completion) ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ।
- ਬਿਲਡ (Build), ਐਗਜ਼ੀਕਿਊਸ਼ਨ (Execution), ਅਤੇ ਡੀਬਗਿੰਗ ਟੂਲ (Debugging Tools)

IDEs ਦੀਆਂ ਅਜਿਹੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਪ੍ਰੋਗਰਾਮਰਾਂ ਨੂੰ ਆਪਣੀ ਉਤਪਾਦਕਤਾ (Productivity) ਵਧਾਉਣ ਵਿੱਚ ਮਦਦ ਕਰਦੀਆਂ ਹਨ। ਜ਼ਿਆਦਾਤਰ IDE ਬਹੁਤ ਸਾਰੀਆਂ ਵੱਖ-ਵੱਖ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਨੂੰ ਸਪੋਰਟ ਕਰਦੇ ਹਨ ਅਤੇ ਇਹਨਾਂ ਵਿੱਚ ਕਈ ਹੋਰ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਵੀ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ।

ਪਾਈਥਨ ਸ਼ੈੱਲ (Python Shell) ਜਾਂ IDLE (**ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਿਵੈਲਪਮੈਂਟ ਐਂਡ ਲਰਨਿੰਗ ਇਨਵਾਇਰਮੈਂਟ**) ਇੱਕ ਡਿਫਾਲਟ ਐਡੀਟਰ ਹੈ ਜੋ ਪਾਈਥਨ ਨਾਲ ਆਪਣੇ ਆਪ ਇੰਸਟਾਲ ਹੋ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਧਾਰਨ IDE/ਐਡੀਟਰ ਸ਼ੁਰੂਆਤੀ ਪੱਧਰ ਦੇ ਡਿਵੈਲਪਰਾਂ ਲਈ ਢੁਕਵਾਂ ਹੈ। IDLE ਟੂਲ ਨੂੰ Mac OS, Windows ਅਤੇ Linux ਪਲੇਟਫਾਰਮਾਂ 'ਤੇ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਹ ਮੁਫਤ ਵਿਚ (free of cost) ਪ੍ਰਦਾਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। IDLE ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪਾਈਥਨ ਕੋਡ ਲਿਖਣਾ ਸਧਾਰਨ ਲੇਵਲ ਦੇ ਪ੍ਰੋਗਰਾਮਾਂ ਲਈ ਬਹੁਤ ਵਧੀਆ ਹੈ, ਪਰ IDLE ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੱਡੇ ਪ੍ਰੋਗਰਾਮਿੰਗ ਪ੍ਰੋਜੈਕਟਾਂ ਨੂੰ ਹੈਂਡਲ ਕਰਨਾ ਮੁਸ਼ਕਲ ਹੋ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਪ੍ਰੋਫੈਸ਼ਨਲ IDE ਜਾਂ ਇੱਕ ਵਧੀਆਂ-ਸਮਰਪਿਤ (good-dedicated) ਕੋਡ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰਨਾ, ਕੋਡਿੰਗ ਨੂੰ ਮਜ਼ੇਦਾਰ ਬਣਾਉਂਦਾ ਹੈ। IDLE ਵਿਚ **ਇੰਟਰਪ੍ਰੇਟਰ (Interpreter)** ਵੀ ਹੁੰਦਾ ਹੈ ਜਿਸਦੀ ਵਰਤੋਂ ਪਾਈਥਨ ਨਿਰਦੇਸ਼ਾਂ ਅਤੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਚਲਾਉਣ (execute) ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਪਾਈਥਨ ਲਈ ਕਈ ਥਰਡ-ਪਾਰਟੀ IDE ਅਤੇ ਕੋਡ ਐਡੀਟਰ ਉਪਲਬਧ ਹਨ, ਉਦਾਹਰਨ ਲਈ: PyCharm, Visual Code Editor, Sublime Text 3, Jupyter Notebook, Spyder ਆਦਿ।

(**IDE ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਿਵੈਲਪਮੈਂਟ ਇਨਵਾਇਰਨਮੈਂਟ**) ਇੱਕ ਸਾਫਟਵੇਅਰ ਹੁੰਦਾ ਹੈ ਜੋ ਸਾਫਟਵੇਅਰ ਡਿਵੈਪਲਮੈਂਟ ਲਈ ਸਮਰਪਿਤ **(dedicated)** ਹੁੰਦਾ ਹੈ।

2.3 ਪਾਈਥਨ ਵਿਚ ਕੋਡ ਚਲਾਉਣ ਦੇ ਮੋਡਜ਼ (MODES OF CODE EXECUTING IN PYTHON)

ਪਾਈਥਨ ਨਿਰਦੇਸ਼ਾਂ (Instructions) ਅਤੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਚਲਾਉਣ (Execution) ਲਈ ਸਾਨੂੰ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੇਟਰ (Interpreter) ਦੀ ਜ਼ਰੂਰਤ ਪੈਂਦੀ ਹੈ। **ਇੰਟਰਪ੍ਰੇਟਰ** ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਹੁੰਦਾ ਹੈ ਜੋ ਸਾਡੇ ਕੋਡ ਨੂੰ ਮਸ਼ੀਨ ਭਾਸ਼ਾ ਵਿੱਚ ਅਨੁਵਾਦ ਕਰਦਾ ਹੈ ਅਤੇ ਫਿਰ ਇਸਨੂੰ ਲਾਈਨ-ਦਰ-ਲਾਈਨ ਚਲਾਉਂਦਾ ਹੈ। ਅਸੀਂ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੇਟਰ ਨੂੰ ਦੋ ਮੋਡਜ਼ ਵਿੱਚ ਵਰਤ ਸਕਦੇ ਹਾਂ:

- **ਇੰਟਰਐਕਟਿੱਵ ਮੋਡ (Interactive Mode):** ਇਸ ਮੋਡ ਵਿੱਚ ਕੰਮ ਕਰਨ ਲਈ ਅਸੀਂ ਪਾਈਥਨ ਸ਼ੈੱਲ (Shell) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਇਸ ਵਿੱਚ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਸਾਡੇ ਵੱਲੋਂ ਕਮਾਂਡ ਪ੍ਰਾਪਤ ਕਰਨ ਦੀ ਉਡੀਕ ਕਰਦਾ ਹੈ। ਜਦੋਂ ਅਸੀਂ ਕਮਾਂਡ ਟਾਈਪ ਕਰਦੇ ਹਾਂ ਤਾਂ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਇਸ ਨੂੰ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ ਅਤੇ ਕਮਾਂਡ ਨੂੰ ਚਲਾਉਂਦਾ ਹੈ, ਫਿਰ ਇਹ ਸਾਡੀ ਅਗਲੀ ਕਮਾਂਡ ਲਈ ਦੁਬਾਰਾ ਉਡੀਕ ਕਰਦਾ ਹੈ।
- **ਸਕ੍ਰਿਪਟ ਮੋਡ (Script Mode) :** ਇੰਟਰਐਕਟਿਵ ਮੋਡ (ਪਾਈਥਨ ਸ਼ੈੱਲ) ਕੋਡ ਦੇ ਛੋਟੇ ਹਿੱਸਿਆਂ ਦੀ ਜਾਂਚ ਕਰਨ ਲਈ ਬਹੁਤ ਵਧੀਆ ਹੈ, ਪਰ ਇਸ ਵਿੱਚ ਇੱਕ ਸਮੱਸਿਆ ਹੈ :- ਅਸੀਂ ਪਾਈਥਨ ਸ਼ੈੱਲ ਵਿੱਚ ਜੋ ਸਟੇਟਮੈਂਟਸ ਦਾਖਲ ਕਰਦੇ ਹਾਂ ਉਹ ਕਿਤੇ ਵੀ ਸੇਵ ਨਹੀਂ ਹੁੰਦੀਆਂ। ਜੇਕਰ ਅਸੀਂ ਸ਼ੈੱਲ ਵਿਚ ਦਾਖਲ ਕੀਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਸੈੱਟ ਨੂੰ ਵਾਰ-ਵਾਰ ਚਲਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਸਾਡੇ ਲਈ ਇਹਨਾਂ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਇੱਕ ਫਾਈਲ ਵਿੱਚ ਸੇਵ ਕਰਨਾ ਬਿਹਤਰ ਹੋਵੇਗਾ। ਫਿਰ ਇਸ ਸੋਰਸ ਕੋਡ (ਸਕ੍ਰੀਪਟ ਫਾਈਲ) ਨੂੰ ਚਲਾਉਣ ਲਈ ਸਕ੍ਰਿਪਟ ਮੋਡ ਵਿੱਚ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਸਕ੍ਰਿਪਟ ਫਾਈਲ (ਪ੍ਰੋਗਰਾਮ) ਬਣਾਉਣ ਲਈ ਅਸੀਂ ਕਿਸੇ ਵੀ ਟੈਕਸਟ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

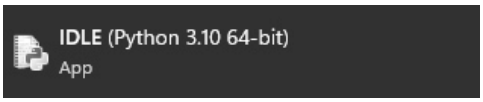
ਇੰਟਰਪ੍ਰੈਟਰ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਹੁੰਦਾ ਹੈ ਜੋ ਸਾਡੇ ਕੋਡ ਦਾ ਮਸ਼ੀਨ ਭਾਸ਼ਾ ਵਿੱਚ ਅਨੁਵਾਦ ਕਰਦੇ ਹੋਏ ਇਸਨੂੰ ਲਾਈਨ ਦਰ ਲਾਈਨ ਚਲਾਉਂਦਾ ਹੈ।

ਨੋਟ: **IDLE ਸਮੇਤ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਦੀ ਇੰਸਟਾਲੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ ਇਸ ਕਿਤਾਬ ਦੀ ਅੰਤਿਕਾ-1 ਵਿੱਚ ਦਿੱਤੀ ਗਈ ਹੈ।**

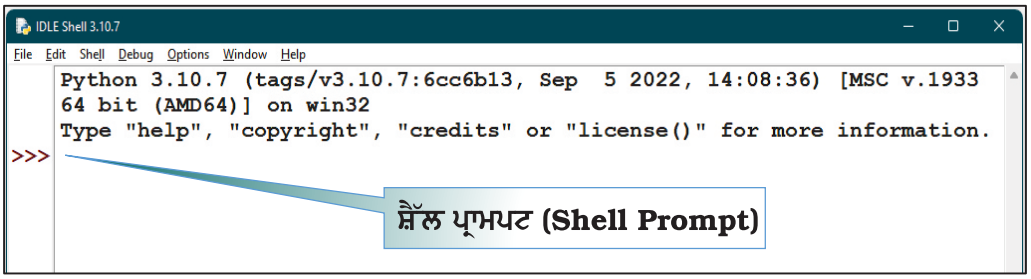
2.4 IDLE ਦੀ ਵਰਤੋਂ ਨਾਲ ਕੋਡ ਲਿਖਣਾ ਅਤੇ ਚਲਾਉਣਾ (Writing & Executing code with IDLE)

ਪਾਈਥਨ ਇੰਸਟਾਲਰ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਤੋਂ ਇਲਾਵਾ ਵਿੰਡੋਜ਼ ਲਈ ਇੱਕ ਲਾਈਟਵੇਟ (Light-weight) ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਿਵੈਲਪਮੈਂਟ ਐਂਡ ਲਰਨਿੰਗ ਇਨਵਾਇਰਮੈਂਟ (IDLE) ਨੂੰ ਵੀ ਇੰਸਟਾਲ ਕਰਦਾ ਹੈ। ਇਹ ਸਾਨੂੰ ਕਮਾਂਡ ਲਾਈਨ ਦੀ ਵਰਤੋਂ ਕੀਤੇ ਬਿਨਾਂ ਇੱਕੋ ਜਗ੍ਹਾ ਸਾਡੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਬਣਾਉਣ/ਪੜ੍ਹਨ/ਐਡਿਟ ਕਰਨ ਅਤੇ ਚਲਾਉਣ ਦੀ ਇਜਾਜ਼ਤ ਦਿੰਦਾ ਹੈ। ਅਸੀਂ IDLE ਦੀ ਵਰਤੋਂ ਦੋਵਾਂ ਮੋਡਜ਼ ਵਿੱਚ ਕਰ ਸਕਦੇ ਹਾਂ, ਅਰਥਾਤ ਇੰਟਰਐਕਟਿਵ ਅਤੇ ਸਕ੍ਰਿਪਟ ਮੋਡ ਵਿੱਚ। ਵਿੰਡੋਜ਼ ਪਲੇਟਫਾਰਮ ਵਿਚ IDLE ਸ਼ੁਰੂ ਕਰਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਵਰਤੋਂ ਕਰੋ:

1. ਸਟਾਰਟ ਮੀਨੂੰ ਓਪਨ ਕਰੋ।
2. "IDLE" ਸਰਚ ਕਰੋ।
3. ਚਿੱਤਰ ਵਿਚ ਦਿਖਾਏ ਅਨੁਸਾਰ IDLE ਉਪਰ ਕਲਿੱਕ ਕਰੋ:



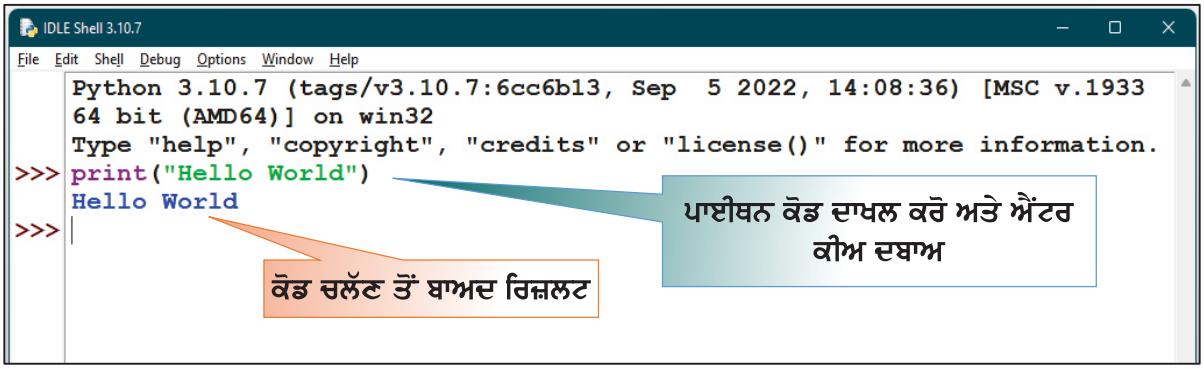
ਤੁਹਾਨੂੰ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ ਇੱਕ ਵਿੰਡੋ ਨਜ਼ਰ ਆਵੇਗੀ:



ਚਿੱਤਰ 2.1 : ਪਾਈਥਨ IDLE ਸ਼ੈੱਲ ਅਤੇ ਇਸਦਾ ਪ੍ਰਾਮਪਟ (Prompt)

2.4.1 ਇੰਟਰਐਕਟਿਵ ਮੋਡ ਵਿਚ ਕੋਡ ਲਿਖਣਾ ਅਤੇ ਚਲਾਉਣਾ (Write & Execute Code in Interactive Mode):

IDLE ਵਿੰਡੋ ਪਾਈਥਨ ਸ਼ੈੱਲ ਮੋਡ ਵਿੱਚ ਓਪਨ ਹੁੰਦੀ ਹੈ। ਚਿੰਨ੍ਹ >>> ਸ਼ੈੱਲ ਪ੍ਰਾਮਪਟ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਜਿੱਥੇ ਅਸੀਂ ਆਪਣੀਆਂ ਕਮਾਂਡਜ਼ ਟਾਈਪ ਕਰਾਂਗੇ। ਸ਼ੈੱਲ ਪ੍ਰਾਮਪਚ ਉਪਰ ਪਾਈਥਨ ਕੋਡ ਟਾਈਪ ਕਰੋ ਅਤੇ ਐਂਟਰ ਕੀਅ ਦਬਾਓ। ਇਹ ਤੁਰੰਤ ਦਾਖਲ ਕੀਤੇ ਕੋਡ ਦਾ ਨਤੀਜਾ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇਗਾ। ਹੇਠ ਦਿੱਤੀ ਉਦਾਹਰਨ 'ਤੇ ਧਿਆਣ ਦਵੋ ਜਿਸ ਵਿਚ ਅਸੀਂ print( ) ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਨਾਲ ਇਕ ਮੈਸੇਜ਼ ਦਰਸਾ ਰਹੇ ਹਾਂ:



ਚਿੱਤਰ 2.2: ਪਾਈਤਨ ਸ਼ੈੱਲ (ਇੰਟਰਐਕਟਿਵ ਮੋਡ) ਵਿਚ ਕੋਡ ਲਿਖਣਾ ਅਤੇ ਚਲਾਉਣਾ

2.1 TEST Yourself

ਹੇਠਾਂ ਦਿੱਤਾ ਕੋਡ ਪਾਈਥਨ ਦੇ ਇੰਟਰਐਕਟਿਵ ਮੋਡ ਵਿੱਚ ਚਲਾਓ ਅਤੇ ਆਉਟਪੁੱਟ ਲਿਖੋ:

- 7 * 4 / 2 _____
- print(7*4) _____
- print("Good Morning") _____

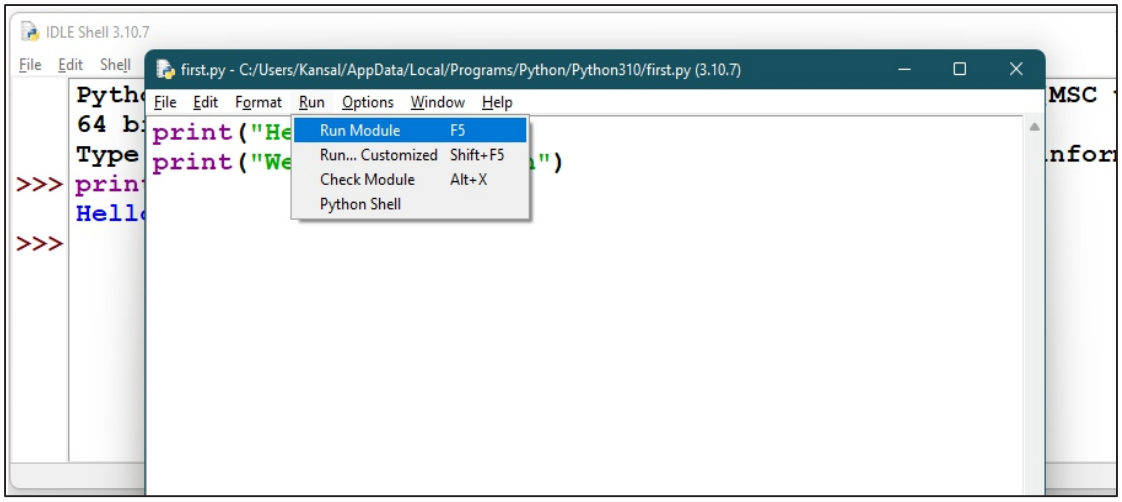
ਨੋਟ: ਜੇਕਰ ਅਸੀਂ ਇੰਟਰਐਕਟਿਵ ਵਿੰਡੋ ਵਿੱਚ ਕਿਸੇ ਪੁਰਾਣੀ ਕਮਾਂਡ ਨੂੰ ਦੁਬਾਰਾ ਤੋਂ ਵਰਤਣਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਅਸੀਂ ਕਮਾਂਡ ਹਿਸਟਰੀ ਰਾਹੀਂ ਪਿੱਛੇ ਵੱਲ ਸਕ੍ਰੋਲ ਕਰਨ ਲਈ ਅੱਪ ਐਰੋ ਕੀਅ ਅਤੇ ਅੱਗੇ ਵੱਲ ਸਕ੍ਰੋਲ ਕਰਨ ਲਈ ਡਾਊਨ ਐਰੋ ਕੀਅ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਕਮਾਂਡ ਸਿਲੈਕਟ ਕਰਨ ਲਈ ਐਂਟਰ ਕੀਅ ਦੀ ਵਰਤੋਂ ਕਰੋ। ਇਹਨਾਂ ਕੀਅਜ਼ ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਸੀਂ ਪੁਰਾਣੀਆਂ ਕਮਾਂਡਾਂ ਨੂੰ ਵਾਪਸ ਦੁਬਾਰਾ ਤੋਂ ਵਰਤ ਸਕਦੇ ਹਾਂ ਅਤੇ ਅਸੀਂ ਉਹਨਾਂ ਨੂੰ ਐਡਿਟ ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ।

2.4.2 ਸਕ੍ਰਿਪਟ ਮੋਡ ਵਿਚ ਕੋਡ ਲਿਖਣਾ ਅਤੇ ਚਲਾਉਣਾ (Write and Execute Code in Script Mode):

IDLE ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ (ਸਕ੍ਰਿਪਟ ਮੋਡ ਵਿਚ) ਲਿਖਣ ਲਈ ਇੱਕ ਬਿਲਟ-ਇਨ ਟੈਕਸਟ ਐਡੀਟਰ ਵੀ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਕ ਨਵਾਂ ਪ੍ਰੋਗਰਾਮ ਬਣਾਉਣ ਲਈ File —> New File 'ਤੇ ਕਲਿੱਕ ਕਰੋ। ਇੱਕ ਨਵੀਂ ਬਿਨ੍ਹਾਂ ਸਿਰਲੇਖ ਵਾਲੀ (Untitled) ਵਿੰਡੋ ਓਪਨ ਹੋਵੇਗੀ। ਇਹ ਵਿੰਡੋ ਇੱਕ ਟੈਕਸਟ ਐਡੀਟਰ ਹੈ ਜਿੱਥੇ ਅਸੀਂ ਪ੍ਰੋਗਰਾਮ ਲਿਖ ਸਕਦੇ ਹਾਂ। ਐਡੀਟਰ ਵਿੱਚ ਅੱਗੇ ਦਿੱਤਾ ਕੋਡ ਟਾਈਪ ਕਰੋ ਅਤੇ ਫਾਈਲ ਨੂੰ ਸੇਵ ਕਰੋ (Ctrl + S ਦਬਾਓ ਜਾਂ File —> Save As ਤੇ ਜਾਓ)। ਫਾਈਲ ਨੂੰ first.py ਦੇ ਤੌਰ 'ਤੇ ਸੇਵ ਕਰੋ।

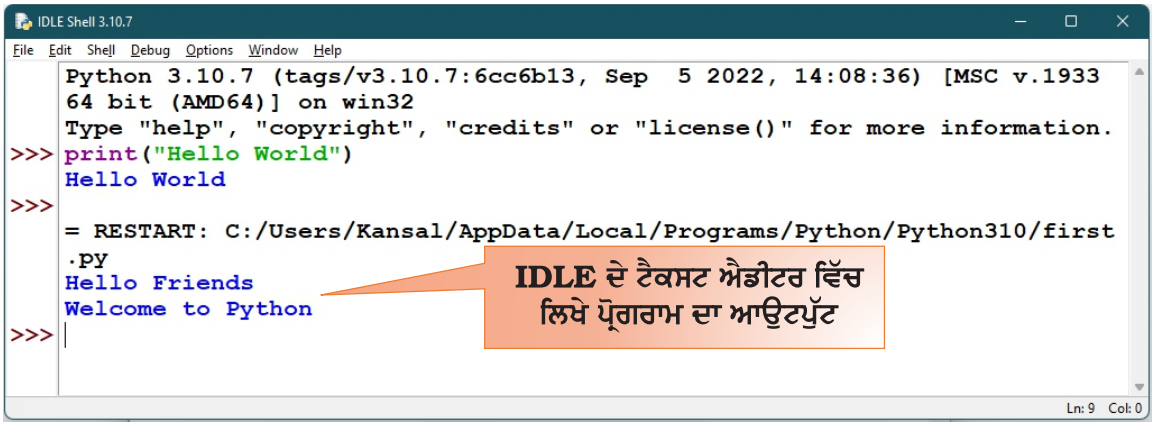
ਚਿੱਤਰ 2.3: **IDLE** ਦੇ ਟੈਕਸਟ ਐਡੀਟਰ ਵਿੱਚ ਪਾਈਥਨ ਕੋਡ ਲਿਖਣਾ (ਸਕ੍ਰਿਪਟ ਮੋਡ)

ਪ੍ਰੋਗਰਾਮ ਸੇਵ ਹੋਣ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਇਸਨੂੰ ਰਨ ਕਰਾਂਗੇ। ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਰਨ ਕਰਨ ਲਈ Run ਮੀਨੂੰ ਵਿਚ Run Module ਆਪਸ਼ਨ ਤੇ ਕਲਿੱਕ ਕਰੋ ਜਾਂ ਕੀਅਬੋਰਡ ਤੋਂ ਫੰਕਸ਼ਨ ਕੀਅ F5 ਪ੍ਰੈਸ ਕਰੋ।



ਚਿੱਤਰ 2.4: **IDLE** ਦੇ ਟੈਕਸਟ ਐਡੀਟਰ ਵਿਚ ਪਾਈਥਨ ਕੋਡ ਨੂੰ ਚਲਾਉਣਾ **(Execution)**

ਤੁਹਾਡੀ ਐਡੀਟਰ ਵਿੰਡੋ ਬੈਕਗ੍ਰਾਉਂਡ ਵਿੱਚ ਚਲੀ ਜਾਵੇਗੀ ਅਤੇ ਪਾਈਥਨ ਸ਼ੈੱਲ ਐਕਟਿਵ (active) ਹੋ ਜਾਵੇਗਾ, ਜਿੱਥੇ ਤੁਹਾਨੂੰ first.py ਪ੍ਰੋਗਰਾਮ ਦਾ ਆਉਟਪੁੱਟ ਇਸ ਤਰ੍ਹਾਂ ਦਿਖਾਈ ਦੇਵੇਗਾ:



ਚਿੱਤਰ 2.5: **IDLE** ਦੇ ਟੈਕਸਟ ਐਡੀਟਰ ਵਿੱਚ ਲਿਖੇ ਪਾਇਥਨ ਕੋਡ ਦਾ ਆਉਟਪੁੱਟ

ਇਸ ਕਿਤਾਬ ਵਿੱਚ ਅਸੀਂ ਵੱਖ-ਵੱਖ ਉਦਾਹਰਣਾਂ ਲਈ ਇਸ IDLE ਦੀ ਵਰਤੋਂ ਕਰਾਂਗੇ।

2.2 TEST YOURSELF

ਇਸ ਕੋਡ ਨੂੰ ਸਕ੍ਰਿਪਟ ਮੋਡ ਵਿੱਚ ਚਲਾਓ ਅਤੇ ਆਉਟਪੁੱਟ ਨੂੰ ਲਿਖੋ:

```
a=10
b=20
print(a+b)
```

ਸ਼ਾਰਟਕੱਟ ਕੀਅਜ਼:

Ctrl + N	-	ਨਵੀਂ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਬਨਾਉਣ ਲਈ
Ctrl + O	-	ਮੌਜੂਦਾ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਨੂੰ ਓਪਨ ਕਰਨ ਲਈ
Ctrl + S	-	ਮੌਜੂਦਾ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਨੂੰ ਸੇਵ ਕਰਨ ਲਈ
F5	-	ਮੌਜੂਦਾ ਮਾਡੀਉਲ (ਸਕ੍ਰਿਪਟ ਫਾਈਲ) ਨੂੰ ਚਲਾਉਣ ਲਈ

2.5 ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਲਈ ਬੁਨਿਆਦੀ ਧਾਰਨਾਵਾਂ (Basic Concepts for Python programming)

ਪਾਈਥਨ ਨੂੰ ਡਿਜ਼ਾਇਨ ਕਰਦੇ ਸਮੇਂ ਕੋਡ ਦੀ ਪੜ੍ਹਨਯੋਗਤਾ (readability) 'ਤੇ ਜ਼ੋਰ ਦਿਤਾ ਗਿਆ ਸੀ, ਅਤੇ ਇਸਦਾ ਸਿੰਟੈਕਸ ਇਸ ਤਰ੍ਹਾਂ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਸੀ ਕਿ ਪ੍ਰੋਗਰਾਮਰ ਆਪਣੇ ਲਾਜਿਕ (Logic) ਨੂੰ ਕੋਡ ਦੀਆਂ ਘੱਟ ਲਾਈਨਾਂ (fewer lines) ਵਿੱਚ ਪ੍ਰਗਟ ਕਰ ਸਕੇ। ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਵਿੱਚ ਕਰਲੀ ਬਰੇਸਿਸ (curly braces { }) ਜਾਂ ਸੈਮੀਕੋਲਨ (;) ਦੀ ਵਰਤੋਂ ਨਹੀਂ ਕੀਤੀ ਜਾਂਦੀ। ਇਹ ਅੰਗਰੇਜ਼ੀ ਭਾਸ਼ਾ ਵਰਗੀ ਭਾਸ਼ਾ ਹੈ। ਪਾਈਥਨ ਕਮਾਂਡ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਨਵੀਂ ਲਾਈਨ (new line) ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ।

ਪਾਈਥਨ ਨੂੰ ਸਿੱਖਣਾ ਸ਼ੁਰੂ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ, ਸਾਨੂੰ ਇਸ ਵਿੱਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਟੋਕਨਾਂ (Tokens) ਅਤੇ ਇਸ ਵਿਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਅੱਖਰਾਂ ਦੇ ਸਮੂਹ ਨੂੰ ਜਾਣਨਾ ਜ਼ਰੂਰੀ ਹੈ। ਕਿਸੇ ਵੀ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਇਹ ਤੱਤ ਮੌਜੂਦ ਹੁੰਦੇ ਹਨ। ਆਓ ਇਹਨਾਂ ਧਾਰਨਾਵਾਂ ਨੂੰ ਸਮਝਣਾ ਸ਼ੁਰੂ ਕਰੀਏ:

2.5.1 ਕਰੈਕਟਰ ਸੈੱਟ (Character Set)

ਇੱਕ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੁਆਰਾ ਸਵੀਕਾਰਯੋਗ ਅੱਖਰਾਂ ਦੇ ਸਮੂਹ (set of acceptable characters) ਨੂੰ ਕਰੈਕਟਰ ਸੈੱਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਅਸੀਂ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਲਿਖਣ ਵੇਲੇ ਸਿਰਫ਼ ਉਹੀ ਕਰੈਕਟਰ (ਅੱਖਰ) ਵਰਤ ਸਕਦੇ ਹਾਂ ਜੋ ਪਾਈਥਨ ਦੇ ਕਰੈਕਟਰ ਸੈੱਟ ਦਾ ਹਿੱਸਾ ਹੁੰਦੇ ਹਨ। ਪਾਈਥਨ ਸਾਰੇ ASCII / ਯੂਨੀਕੋਡ (UNICODE) ਅੱਖਰਾਂ ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ, ਜਿਸ ਵਿੱਚ ਹੇਠਾਂ ਅਨੁਸਾਰ ਅੱਖਰ/ਚਿੰਨ੍ਹ ਸ਼ਾਮਲ ਹਨ:

- **ਕਰੈਕਟਰਜ਼ (Characters):** A-Z ਅਤੇ a-z ਕਰੈਕਟਰਜ਼ ਅਤੇ ਹੋਰ ਸਾਰੇ ਯੂਨੀਕੋਡ ਕਰੈਕਟਰਜ਼
- **ਅੰਕ (Digits):** 0 ਤੋਂ 9 ਤੱਕ ਸਾਰੇ ਅੰਕ (Digits)
- **ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ (Special Symbols)** : ਸਾਰੇ ਪ੍ਰਿੰਟੇਬਲ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਜਿਵੇਂ ਕਿ: " ' | ; : ! ~ @ # $ % ^ ' & * ( ) - + – = {}[ ] \.
- **ਖਾਲੀ ਥਾਵਾਂ (White Spaces)** : ਜਿਵੇਂ ਕਿ ਟੈਬ ਸਪੇਸ (Tab Space), ਖਾਲੀ ਥਾਂ (Blank Space), ਨਵੀਂ ਲਾਈਨ (New Line), ਅਤੇ ਕੈਰੇਜ ਰਿਟਰਨ (Carriage Return)

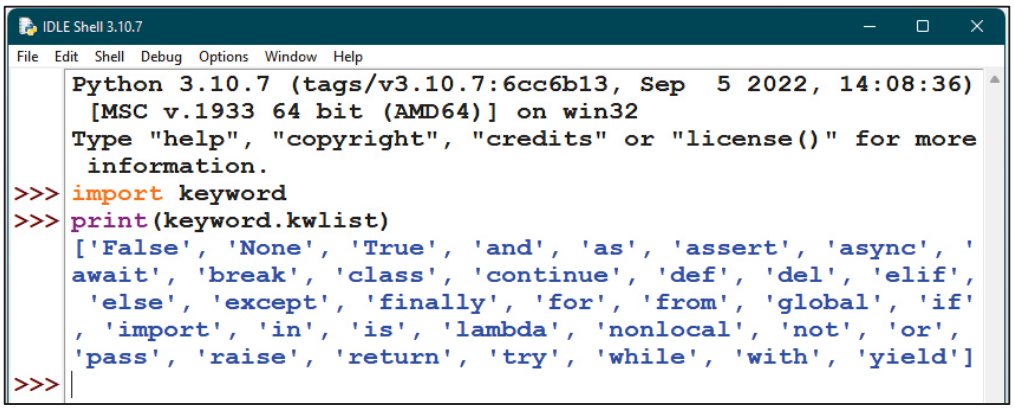
ਇੱਕ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੁਆਰਾ ਸਵੀਕਾਰਯੋਗ ਅੱਖਰਾਂ ਦੇ ਸਮੂਹ (Set of Acceptable Characters) ਨੂੰ ਕਰੈਕਟਰ ਸੈੱਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

2.5.2 ਟੋਕਨਜ਼ (Tokens)

ਟੋਕਨਜ਼ ਕਿਸੇ ਵੀ ਸੋਰਸ ਕੋਡ ਦੇ ਬਹੁਤ ਬੁਨਿਆਦੀ ਹਿੱਸੇ (Basic Components) ਹੁੰਦੇ ਹਨ। ਹਰੇਕ ਪ੍ਰੋਗਰਾਮ ਵੱਖ-ਵੱਖ ਤਰ੍ਹਾਂ ਦੇ ਟੋਕਨਜ਼ ਤੋਂ ਮਿਲ ਕੇ ਬਣਿਆ ਹੁੰਦਾ ਹੈ। ਇਹ ਟੋਕਨ ਅੰਗਰੇਜ਼ੀ ਭਾਸ਼ਾ ਵਿੱਚ ਸ਼ਬਦਾਂ (Words) ਅਤੇ ਵਿਰਾਮ ਚਿੰਨ੍ਹਾਂ (Punctuation Marks) ਵਾਂਗ ਹੁੰਦੇ ਹਨ। ਟੋਕਨ ਸਭ ਤੋਂ ਛੋਟੇ ਵਿਅਕਤੀਗਤ ਤੱਤ (Smallest Individual Elements) ਹੁੰਦੇ ਹਨ। ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਟੋਕਨ ਕੀਵਰਡਜ਼, (Keywords) ਆਇਡੈਂਟੀਫਾਇਰਜ਼ (Identifiers), ਲਿਟਰਲਜ਼ (Literals), ਓਪਰੇਟਰਜ਼ (Operators) ਅਤੇ ਡੀਲੀਮੀਟਰਜ਼ (Delimiters) ਹੁੰਦੇ ਹਨ। ਆਓ ਹੁਣ ਇਹਨਾਂ ਟੋਕਨਜ਼ ਬਾਰੇ ਇੱਕ-ਇੱਕ ਕਰਕੇ ਚਰਚਾ ਕਰੀਏ:

ਟੋਕਨ ਸਭ ਤੋਂ ਛੋਟੇ ਵਿਅਕਤੀਗਤ ਤੱਤ (Smallest Individual Elements) ਹੁੰਦੇ ਹਨ:
ਪਾਈਥਨ ਵਿਚ ਟੋਕਨਜ਼: ਕੀਵਰਡਜ਼, ਆਇਡੈਂਟੀਫਾਇਰਜ਼, ਲਿਟਰਲਜ਼, ਓਪਰੇਟਰਜ਼ ਅਤੇ ਡੀਲੀਮੀਟਰਜ਼

2.5.2.1 ਕੀਅਵਰਡਜ਼ (Keywords) : ਕੀਵਰਡਜ਼ ਰਿਜ਼ਰਵ ਵਰਡਜ਼ (Reserve Words) ਹੁੰਦੇ ਹਨ। ਇਹਨਾਂ ਸ਼ਬਦਾਂ ਦੇ ਵਿਸ਼ੇਸ਼ ਅਰਥ ਹੁੰਦੇ ਹਨ ਅਤੇ ਇਹਨਾਂ ਸ਼ਬਦਾਂ ਦੇ ਅਰਥ ਬਦਲੇ ਨਹੀਂ ਜਾ ਸਕਦੇ। ਜ਼ਿਆਦਾਤਰ ਕੀਵਰਡਜ਼ ਅੰਗਰੇਜ਼ੀ ਦੇ ਸਿਰਫ ਛੋਟੇ ਅੱਖਰਾਂ (Lowercase Letters) ਤੋਂ ਮਿਲ ਕੇ ਬਣੇ ਹੁੰਦੇ ਹਨ। ਅਸੀਂ ਕੀਵਰਡਜ਼ ਨੂੰ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਵਜੋਂ ਨਹੀਂ ਵਰਤ ਸਕਦੇ (ਜਿਵੇਂ ਕਿ: ਪ੍ਰੋਗਰਾਮ ਦੇ ਤੱਤਾਂ - ਵੇਰੀਏਬਲਜ਼, ਫੰਕਸ਼ਨਾਂ ਦੇ ਨਾਂ ਆਦਿ ਵੱਜੋਂ), ਪਾਈਥਨ ਕੀਵਰਡਜ਼ ਦੀ ਸੂਚੀ ਵੇਖਣ ਲਈ ਅਸੀਂ ਪਾਈਥਨ ਸ਼ੈੱਲ ਵਿੱਚ ਪਾਈਥਨ ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:



ਚਿੱਤਰ 2.6: ਕੀਅਵਰਡਜ਼ ਦੀ ਲਿਸਟ ਦੇਖਣ ਲਈ ਪਾਈਥਨ ਸਟੇਟਮੈਂਟਸ

2.5.2.2 ਆਈਡੈਂਟੀਫਾਇਰਜ਼ (Identifiers) : ਪ੍ਰੋਗਰਾਮ ਐਲੀਮੈਂਟਸ (Elements) ਜਿਵੇਂ ਕਿ ਵੇਰੀਏਬਲ, ਫੰਕਸ਼ਨ, ਕਲਾਸ, ਲਿਸਟ, ਟਪਲ, ਆਦਿ ਨੂੰ ਉਹਨਾਂ ਦੀ ਪਛਾਣ ਲਈ ਦਿੱਤੇ ਜਾਣ ਵਾਲੇ ਨਾਮ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਅਖਵਾਉਂਦੇ ਹਨ। ਅਸੀਂ ਪ੍ਰੋਗਰਾਮ ਦੇ ਐਲੀਮੈਂਟਸ ਦੀ ਪਛਾਣ ਲਈ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕਿਸੇ ਵੀ ਐਲੀਮੈਂਟ ਦਾ ਨਾਮ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਸਾਨੂੰ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਦੇ ਨਾਮਕਰਨ ਨਿਯਮਾਂ (Naming Rules) ਦੀ ਪਾਲਣਾ ਕਰਨੀ ਪਵੇਗੀ:

- ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਲੋਅਰਕੇਸ ਜਾਂ ਅੱਪਰਕੇਸ ਅਲਫਾਬੇਟਸ ਜਾਂ ਅੰਕ (0-9) ਜਾਂ ਅੰਡਰਸਕੋਰ (_) ਦੇ ਸੁਮੇਲ ਤੋਂ ਬਣਿਆ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।
- ਆਇਡੈਂਟੀਫਾਇਰ ਕਿਸੇ ਅੰਕ (digit) ਨਾਲ ਸ਼ੁਰੂ ਨਹੀਂ ਹੋ ਸਕਦਾ।
- ਕੋਈ ਵੀ ਕੀਵਰਡ (keyword) ਆਇਡੈਂਟੀਫਾਇਰ ਵਜੋਂ ਨਹੀਂ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ।
- ਅੰਡਰਸਕੋਰ (_) ਤੋਂ ਇਲਾਵਾ ਕੋਈ ਵੀ ਚਿੰਨ੍ਹ (Symbol) ਜਾਂ ਵਿਸ਼ੇਸ਼ ਅੱਖਰ (Special character) ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਵਿੱਚ ਨਹੀਂ ਵਰਤਿਆ ਜਾ ਸਕਦਾ।

- ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਕਿਸੇ ਵੀ ਲੰਬਾਈ ਦਾ ਹੋ ਸਕਦਾ ਹੈ; ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਦੀ ਲੰਬਾਈ 'ਤੇ ਕੋਈ ਪਾਬੰਦੀ ਨਹੀਂ ਹੈ।
- ਪਾਈਥਨ ਇੱਕ ਕੇਸ ਸੰਵੇਦਨਸ਼ੀਲ (Case Sensitive) ਭਾਸ਼ਾ ਹੈ। ਇਸ ਲਈ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਦੇ ਨਾਮਕਰਨ ਵਿੱਚ ਵਰਣਮਾਲਾ (Alphabets) ਦੀ ਕੇਸਿੰਗ ਮਹੱਤਵਪੂਰਨ ਹੈ। ਉਦਾਹਰਣ ਲਈ: roll ਅਤੇ ROLL ਦੋ ਵੱਖ-ਵੱਖ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਹਨ।
- ਇੱਕ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਵਿੱਚ ਖਾਲੀ ਥਾਂਵਾਂ (Whitespaces) ਦੀ ਸਖ਼ਤ ਮਨਾਹੀ ਹੈ।

ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਦੇ ਸਹੀ ਨਾਮਕਰਨ ਦੀਆਂ ਉਦਾਹਰਨਾਂ: rollno, House_No, sector 92 ਆਦਿ।

ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਦੇ ਗਲਤ ਨਾਮਕਰਨ ਦੀਆਂ ਉਦਾਹਰਨਾਂ: 92sector, roll@, roll no, if ਆਦਿ

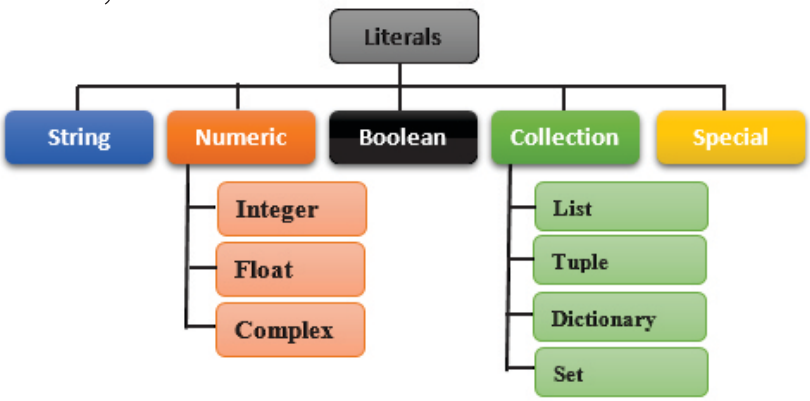
ਪ੍ਰੋਗਰਾਮ ਐਲੀਮੈਂਟਸ (Elements) ਜਿਵੇਂ ਕਿ ਵੇਰੀਏਬਲ, ਫੰਕਸ਼ਨ, ਕਲਾਸ, ਲਿਸਟ, ਟਪਲ, ਆਦਿ ਨੂੰ ਉਹਨਾਂ ਦੀ ਪਛਾਣ ਲਈ ਦਿੱਤੇ ਗਏ ਨਾਮ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਅਖਵਾਉਂਦੇ ਹਨ।

2.3 Test Yourself

ਪਾਈਥਨ ਵਿੱਚ ਹੇਠਾਂ ਦਿੱਤੇ ਆਈਡੈਂਟੀਫਾਇਰਜ਼ ਵਿੱਚੋਂ ਕਿਹੜੇ ਅਵੈਧ (Invalid) ਹਨ ?

%age	total_marks	total-sale	House_No.
False	Serial No	rollno	amountin$

2.5.2.3 ਲਿਟਰਲਜ਼ (Literals) : ਲਿਟਰਲ ਸੋਰਸ ਕੋਡ ਵਿੱਚ ਵਰਤੇ ਗਏ ਸਥਿਰ (fixed) ਮੁੱਲ ਹੁੰਦੇ ਹਨ। ਇਹ ਮੁੱਲ ਆਮ ਤੌਰ 'ਤੇ ਪ੍ਰੋਗਰਾਮ ਐਲੀਮੈਂਟਸ, ਜਿਵੇਂ ਕਿ ਵੇਰੀਏਬਲ ਆਦਿ, ਲਈ ਨਿਰਧਾਰਤ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਪਾਈਥਨ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਲਿਟਰਲਜ਼ ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ:



ਚਿੱਤਰ 2.7: ਪਾਈਥਨ ਵਿੱਚ ਲਿਟਰਲਜ਼ ਦੀਆਂ ਕਿਸਮਾਂ

- **ਸਟ੍ਰਿੰਗ ਲਿਟਰਲਜ਼ (String Literals) :** ਸਿੰਗਲ, ਡਬਲ ਜਾਂ ਟ੍ਰਿਪਲ ਕੋਟਸ (quotes) ਵਿੱਚ ਲਿਖਿਆ ਟੈਕਸਟ (Text) ਪਾਈਥਨ ਵਿੱਚ ਸਟ੍ਰਿੰਗ ਲਿਟਰਲ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: "Compuer Science", 'House # 174', ਆਦਿ। ਅਸੀਂ ਮਲਟੀ-ਲਾਈਨ ਸਟ੍ਰਿੰਗ ਲਿਖਣ ਲਈ ਟ੍ਰਿਪਲ ਕੋਟਸ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਸੰਖਿਆਤਮਕ ਲਿਟਰਲਜ਼ (Numeric Literals) :** ਇਹ ਸੰਖਿਆਵਾਂ (Numbers) ਦੇ ਰੂਪ ਵਿੱਚ ਲਿਖੇ ਜਾਣ ਵਾਲੇ ਗਏ ਲਿਟਰਲਜ਼ ਹਨ। ਪਾਈਥਨ ਹੇਠਾਂ ਦਿੱਤੇ ਸੰਖਿਆਤਮਕ ਲਿਟਰਲਜ਼ ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ:
  - **ਇੰਟੀਜ਼ਰ ਲਿਟਰਲਜ਼ (Integer Literals) :** ਇਹਨਾਂ ਲਿਟਰਲਜ਼ ਵਿੱਚ ਬਿਨਾਂ ਫ੍ਰੈਕਸ਼ਨਲ ਭਾਗ (Fractional Part) ਦੇ ਜ਼ੀਰੋ, ਸਕਾਰਾਤਮਕ (Positive) ਜਾਂ ਰਿਣਾਤਮਕ (Negative) ਸੰਖਿਆਵਾਂ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ। ਇਸ ਵਿੱਚ ਬਾਈਨਰੀ ਲਿਟਰਲ (ਜਿਵੇਂ ਕਿ 0b10101), ਡੈਸੀਮਲ ਲਿਟਰਲ (ਜਿਵੇਂ ਕਿ 9174), ਓਕਟਲ ਲਿਟਰਲ (ਉਦਾਹਰਨ ਲਈ 0o174), ਹੈਕਸਾਡੈਸੀਮਲ ਲਿਟਰਲ (ਉਦਾਹਰਨ ਲਈ

0x5A4) ਵੀ ਸ਼ਾਮਲ ਹੋ ਸਕਦੇ ਹਨ।

- ਫਲੋਟ ਲਿਟਰਲਜ਼ **(Float Literals)** : ਇਹਨਾਂ ਲਿਟਰਲਜ਼ ਨੂੰ ਰੀਅਲ (Real) ਲਿਟਰਲਜ਼ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਵਿੱਚ ਫਰੈਕਸ਼ਨਲ ਭਾਗ (Fractional Part) ਦੇ ਨਾਲ ਜ਼ੀਰੋ, ਸਕਾਰਾਤਮਕ ਜਾਂ ਰਿਣਾਤਮਕ ਨੰਬਰ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਇਹ ਲਿਟਰਲ ਦੋ ਤਰ੍ਹਾਂ ਦੇ ਹੋ ਸਕਦੇ ਹਨ: ਫਰੈਕਸ਼ਨਲ (Fractional) ਅਤੇ ਐਕਸਪੋਨੈਂਸ਼ੀਅਲ (Exponential):
  - **ਫ੍ਰੈਕਸ਼ਨਲ (Fractional) ਲਿਟਰਲ** ਵਿੱਚ ਪੂਰਨ ਅੰਕ (Whole numbers) ਅਤੇ ਦਸ਼ਮਲਵ ਅੰਕ ਦੋਵੇਂ ਹੁੰਦੇ ਹਨ।
  - **ਐਕਸਪੋਨੈਂਸ਼ੀਅਲ (Exponential) ਲਿਟਰਲ** 10 ਦੀ ਪਾਵਰ ਵਿੱਚ ਦਰਸਾਏ ਜਾਂਦੇ ਹਨ। 10 ਦੀ ਪਾਵਰ ਨੂੰ e ਜਾਂ E ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਐਕਸਪੋਨੈਂਸ਼ੀਅਲ ਲਿਟਰਲ ਦੇ ਦੋ ਭਾਗ ਹੁੰਦੇ ਹਨ: ਮੈਂਟੀਸਾ (Mantissa) ਅਤੇ ਐਕਸਪੋਨੈਂਟ (Exponent)। ਉਦਾਹਰਨ ਲਈ: 9.47E5 (ਇਸ ਉਦਾਹਰਣ ਵਿੱਚ 9.47 ਮੈਨਟੀਸਾ ਹੈ ਅਤੇ E ਤੋਂ ਬਾਅਦ ਵਾਲਾ 5 ਐਕਸਪੋਨੈਂਟ ਭਾਗ ਹੈ।)
- ਕੰਪਲੈਕਸ ਲਿਟਰਲਜ਼ **(Complex Literals)** : ਇਸ ਵਿੱਚ A+Bj ਦੇ ਰੂਪ ਵਿੱਚ ਕੰਪਲੈਕਸ ਨੰਬਰ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਇੱਥੇ A ਰੀਅਲ (Real) ਭਾਗ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਅਤੇ B ਕੰਪਲੈਕਸ ਨੰਬਰ ਦੇ ਕਾਲਪਨਿਕ (imaginary) ਭਾਗ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: 4+5j

- ਬੁਲੀਅਨ ਲਿਟਰਲਜ਼ **(Boolean Literals):** ਪਾਈਥਨ ਵਿੱਚ ਬੂਲੀਅਨ ਲਿਟਰਲ ਦੇ ਸਿਰਫ਼ ਦੋ ਮੁੱਲ ਹੁੰਦੇ ਹਨ: True ਅਤੇ false
- ਸਪੈਸ਼ਲ ਲਿਟਰਲਜ਼ **(Special Literals)** : ਪਾਈਥਨ ਵਿੱਚ ਇੱਕ ਵਿਸ਼ੇਸ਼ ਲਿਟਰਲ None ਹੈ। ਇਹ ਕੁੱਝ ਵੀ ਨਹੀਂ (Nothing), ਕੋਈ ਮੁੱਲ ਨਹੀਂ (No Values), ਜਾਂ ਮੁੱਲ ਦੀ ਅਣਹੋਂਦ (Absence of Value) ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।

ਲਿਟਰਲ ਸੋਰਸ ਕੋਡ ਵਿੱਚ ਵਰਤੇ ਗਏ ਸਥਿਰ (fixed) ਮੁੱਲ ਹੁੰਦੇ ਹਨ। ਇਹ ਮੁਲ ਆਮ ਤੌਰ 'ਤੇ ਪ੍ਰੋਗਰਾਮ ਐਲੀਮੈਂਟਸ, ਜਿਵੇਂ ਕਿ ਵੇਰੀਏਬਲ ਆਦਿ ਲਈ ਨਿਰਧਾਰਿਤ ਕੀਤੇ ਜਾਂਦੇ ਹਨ।

- ਕੁਲੈਕਸ਼ਨ ਲਿਟਰਲਜ਼ **(Collections Literals):** ਪਾਈਥਨ ਵਿੱਚ ਅਸੀਂ ਲਿਟਰਲਜ਼ ਦੇ ਸਮੂਹ ਨੂੰ ਇੱਕ ਵੇਰੀਏਬਲ ਵਿੱਚ ਸਟੋਰ ਕਰ ਸਕਦੇ ਹਾਂ। ਲਿਟਰਲਜ਼ ਦੇ ਸਮੂਹ ਨੂੰ ਸਟੋਰ ਕਰਨ ਦੇ ਬਹੁਤ ਸਾਰੇ ਵੱਖ-ਵੱਖ ਤਰੀਕੇ ਹਨ, ਜਿਸ ਵਿੱਚ ਲਿਸਟ (list), ਟਪਲ (tuple), ਡਿਕਸ਼ਨਰੀ (dictionary) ਅਤੇ ਸੈੱਟ (set) ਸ਼ਾਮਲ ਹਨ:
  - ਲਿਸਟ **(List)** : ਇਹ ਚਕੋਰ ਬਰੈਕਟਾਂ (Square brackets) ਵਿੱਚ ਦਰਸਾਏ ਗਏ ਐਲੀਮੈਂਟਸ ਦੀ ਇੱਕ ਸੂਚੀ ਹੁੰਦੀ ਹੈ। ਇਹਨਾਂ ਐਲੀਮੈਂਟਸ ਨੂੰ ਕਾਮਿਆਂ ਨਾਲ ਵੱਖ (Separate) ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਵਿਚ ਮੁੱਲਾਂ ਦੀ ਡੁਪਲੀਕੇਸੀ ਹੋ ਸਕਦੀ ਹੈ। ਇਹ ਐਲੀਮੈਂਟਸ ਕਿਸੇ ਵੀ ਕਿਸਮ (any type) ਦੇ ਹੋ ਸਕਦੇ ਹਨ। ਲਿਸਟਾਂ ਪਰਿਵਰਤਨਸ਼ੀਲ (Mutable) ਹੁੰਦੀਆਂ ਹਨ, ਭਾਵ ਅਸੀਂ ਲਿਸਟਾਂ ਨੂੰ ਬਣਾਉਣ ਤੋਂ ਬਾਅਦ ਉਹਨਾਂ ਵਿਚ ਬਦਲਾਵ (Modify) ਕਰ ਸਕਦੇ ਹਾਂ। ਲਿਸਟ ਦੀ ਉਦਾਹਰਨ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

['Param', 2005, 90.6]

  - ਟਪਲ **(Tuple)** : ਇਹ ਵੀ ਗੋਲ ਬਰੈਕਟਾਂ (Round brackets) ਵਿੱਚ ਕੌਮੇ ਨਾਲ ਵੱਖ ਕੀਤੇ ਐਲੀਮੈਂਟਸ ਜਾਂ ਮੁੱਲਾਂ ਦੀ ਸੂਚੀ ਹੁੰਦੀ ਹੈ। ਇਹ ਐਲੀਮੈਂਟਸ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੇ ਹੋ ਸਕਦੇ ਹਨ। ਟਪਲ ਅਟੱਲ (Immutable) ਹੁੰਦਾ ਹੈ ਜਿਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਇਸਨੂੰ ਬਨਾਉਣ ਤੋਂ ਬਾਅਦ ਇਸ ਵਿਚ ਬਦਲਾਵ ਨਹੀਂ ਕੀਤਾ ਜਾ ਸਕਦਾ। ਟਪਲ ਦੀ ਉਦਾਹਰਨ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

('Param', 2005, 90.6)

  - ਡਿਕਸ਼ਨਰੀ **(Dictionary)** : ਇਹ key: value ਜੋੜਿਆਂ (pairs) ਦੇ ਰੂਪ ਵਿੱਚ ਐਲੀਮੈਂਟਸ ਦਾ

ਸੰਗ੍ਰਹਿ ਹੁੰਦੀ ਹੈ। ਡਿਕਸ਼ਨਰੀ ਦੇ ਐਲੀਮੈਂਟਸ ਕਰਲੀ-ਬ੍ਰੇਸਿਸ {} ਵਿੱਚ ਰੱਖੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਹਰੇਕ ਜੋੜੇ (pair) ਨੂੰ ਕਾਮਿਆਂ (,) ਨਾਲ ਵੱਖ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਡਿਕਸ਼ਨਰੀ ਪਰਿਵਰਤਨਸ਼ੀਲ (Mutable) ਹੁੰਦੀ ਹੈ। ਡਿਕਸ਼ਨਰੀ ਦੀ ਉਦਾਹਰਨ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

{'name':'Param', 'birth_year':2005, 'marks':90.6}

- **ਸੈੱਟ (Set):** ਇਹ ਐਲੀਮੈਂਟਸ ਦਾ ਅਨਆਰਡਰਡ ਸਮੂਹ (Unordered collection) ਹੈ। ਸੈੱਟ ਦੇ ਐਲੀਮੈਂਟਸ ਕਰਲੀ-ਬ੍ਰੇਸਿਸ {} ਵਿੱਚ ਰੱਖੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਹਰੇਕ ਐਲੀਮੈਂਟ ਨੂੰ ਕਾਮੇ ਨਾਲ ਵੱਖ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਸੈੱਟ ਵੀ ਪਰਿਵਰਤਨਸ਼ੀਲ (Mutable) ਹੁੰਦੇ ਹਨ। ਸੈੱਟ ਦੀ ਉਦਾਹਰਨ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

{'Param', 'Karan', 'Sukhi', 'Divya'}

ਪਾਈਥਨ ਵਿੱਚ ਪਰਿਵਰਤਨਸ਼ੀਲ (Mutable) ਅਤੇ ਅਟੱਲ (Immutable) ਕੁਲੈਕਸ਼ਨ ਦੀਆਂ ਕਿਸਮਾਂ ਮੌਜੂਦ ਹਨ। ਸਟ੍ਰਿੰਗ ਅਤੇ ਟਪਲ ਅਟੱਲ ਕਿਸਮ ਦੇ ਹਨ, ਜਦੋਂ ਕਿ ਲਿਸਟਾਂ, ਡਿਕਸ਼ਨਰੀ, ਅਤੇ ਸੈੱਟ ਪਰਿਵਰਤਨਸ਼ੀਲ ਕਿਸਮ ਦੇ ਹੁੰਦੇ ਹਨ। ਪਰਿਵਰਤਨਸ਼ੀਲ ਕਿਸਮਾਂ ਉਹ ਹੁੰਦੀਆਂ ਹਨ ਜਿਨ੍ਹਾਂ ਦੇ ਮੁੱਲ ਬਦਲੇ ਜਾ ਸਕਦੇ ਹਨ, ਜਦੋਂ ਕਿ ਅਟੱਲ ਕਿਸਮਾਂ ਉਹ ਹੁੰਦੀਆਂ ਹਨ ਜਿਨ੍ਹਾਂ ਵਿੱਚ ਮੁੱਲ ਬਦਲੇ ਨਹੀਂ ਜਾ ਸਕਦੇ।

2.4 TEST Yourself

ਪਾਈਥਨ ਵਿਚ ਵਰਤੇ ਜਾਂਦੇ ਲਿਟਰਲਜ਼ ਦੀਆਂ ਕਿਸਮਾਂ ਦੀ ਪਛਾਣ ਕਰੋ:

ਲਿਟਰਲ	ਲਿਟਰਲ ਦੀ ਕਿਸਮ	ਲਿਟਰਲ	ਲਿਟਰਲ ਦੀ ਕਿਸਮ
242	_____	{47,'A',79}	_____
0X174	_____	(47,'A',79)	_____
3+2j	_____	[47,'A',79]	_____
4.25E3	_____	True	_____

2.5.2.4 ਓਪਰੇਟਰਜ਼ (Operators) : ਇਹ ਅਜਿਹੇ ਟੋਕਨ ਹੁੰਦੇ ਹਨ ਜੋ ਇੱਕ ਐਕਸਪ੍ਰੈਸ਼ਨ (Expression) ਵਿੱਚ ਕਿਸੇ ਓਪਰੇਸ਼ਨ ਨੂੰ ਕਰਵਾਉਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਹਨਾਂ ਟੋਕਨਾਂ ਲਈ ਅਸੀਂ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹਾਂ (Special Symbols) ਅਤੇ ਕਰੈਕਟਰਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ +, *, >, <, and, or, in, is ਆਦਿ ਜੋ ਓਪਰੈਂਡਜ਼ (Operands) ਉਪਰ ਅਰਥਮੈਟਿਕ, ਲਾਜ਼ੀਕਲ ਆਦਿ ਓਪਰੇਸ਼ਨ ਕਰਵਾਉਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਵੇਰੀਏਬਲ ਜਾਂ ਮੁੱਲ ਜਿਨ੍ਹਾਂ ਉਪਰ ਓਪਰੇਸ਼ਨ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਨੂੰ ਓਪਰੇਂਡ (Operand) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਓਪਰੇਟਰ ਯੂਨਰੀ (Unary) ਜਾਂ ਬਾਈਨਰੀ (Binary) ਕਿਸਮ ਦੇ ਹੋ ਸਕਦੇ ਹਨ। ਯੂਨਰੀ ਓਪਰੇਟਰ ਉਹ ਓਪਰੇਟਰ ਹੁੰਦੇ ਹਨ ਜੋ ਸਿੰਗਲ ਓਪਰੇਂਡ ਉਪਰ ਆਪਣਾ ਕੰਮ ਕਰਦੇ ਹਨ, ਜਿਵੇਂ ਕਿ: ਕਾਂਪਲੀਮੈਂਟ (Complement) ਆਪਰੇਟਰ (~a) ਆਦਿ। ਜਦੋਂ ਕਿ ਬਾਈਨਰੀ ਓਪਰੇਟਰਾਂ ਨੂੰ ਆਪਣਾ ਕੰਮ ਕਰਨ ਲਈ ਦੋ ਓਪਰੈਂਡਜ਼ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ, ਜਿਵੇਂ a+b, p*q ਆਦਿ। ਓਪਰੇਟਰਜ਼ ਬਾਰੇ ਵਿਸਥਾਰ ਵਿਚ ਜਾਣਕਾਰੀ ਅਸੀਂ ਅਗਲੇ ਪਾਠ ਵਿਚ ਹਾਸਿਲ ਕਰਾਂਗੇ।

2.5.2.5 ਡੀਲੀਮੀਟਰਜ਼ (Delimiters): ਡੀਲੀਮੀਟਰ ਉਹ ਚਿੰਨ੍ਹ ਹੁੰਦੇ ਹਨ ਜੋ ਪਾਈਥਨ ਵਿੱਚ ਤਿੰਨ ਵਿਸ਼ੇਸ਼ ਭੂਮਿਕਾਵਾਂ ਨਿਭਾਉਂਦੇ ਹਨ:

- ਗਰੁਪਿੰਗ (Grouping): [ ] {}( )
- ਵਿਰਾਮ ਚਿੰਨ੍ਹ (Punctuation): . , : @ ਆਦਿ
- ਨਾਮ ਨਾਲ ਓਬਜੈਕਸਟ ਦੀ ਅਸਾਈਨਮੈਂਟ/ਬਾਈਡਿੰਗ (Assignment/Binding of Objects to Name): - = + = * = // = **= = = , ਆਦਿ

2.5.3 ਪਾਈਥਨ ਵੇਰੀਏਬਲਜ਼ (Python Variables):

ਵੇਰੀਏਬਲਜ਼ ਉਹ ਆਈਡੈਂਟੀਫਾਇਰ ਹੁੰਦੇ ਹਨ ਜਿਹਨਾਂ ਨੂੰ ਮੁੱਲ ਸਟੋਰ ਕਰਨ ਲਈ (to store values) ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਅਸੀਂ ਰਨਟਾਈਮ ਦੌਰਾਨ ਇਹਨਾਂ ਦਾ ਮੁੱਲ ਬਦਲ ਵੀ ਸਕਦੇ ਹਾਂ। ਜ਼ਿਆਦਾਤਰ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚ, ਸਾਨੂੰ ਇੱਕ

ਡਾਟਾ ਟਾਈਪ ਨਾਲ ਵੇਰੀਏਬਲ ਡਿਕਲੇਅਰ (Declare) ਕਰਨਾ ਪੈਂਦਾ ਹੈ। ਪਰ ਪਾਈਥਨ ਵਿੱਚ ਇਸ ਤਰ੍ਹਾਂ ਨਹੀਂ ਕੀਤਾ ਜਾਂਦਾ। ਪਾਈਥਨ ਵਿੱਚ ਵੇਰੀਏਬਲ ਡਿਕਲੇਅਰ ਕਰਨ ਲਈ ਕੋਈ ਵਿਸ਼ੇਸ਼ ਕਮਾਂਡ ਨਹੀਂ ਹੁੰਦੀ। ਇਸ ਵਿਚ ਕੋਈ ਵੇਰੀਏਬਲ ਮੈਮਰੀ ਵਿੱਚ ਉਸ ਸਮੇਂ ਹੋਂਦ ਵਿਚ ਆਉਂਦਾ ਹੈ ਜਦੋਂ ਅਸੀਂ ਉਸਨੂੰ ਇੱਕ ਮੁੱਲ ਅਸਾਈਨ (Assign) ਕਰਦੇ ਹਾਂ। ਵੇਰੀਏਬਲ ਨੂੰ ਦਿਤੇ ਗਏ ਮੁੱਲ ਦੀ ਡਾਟਾ ਟਾਈਪ ਦੇ ਅਧਾਰ ਤੇ ਇੰਟਰਪ੍ਰੈਟਰ ਉਸ ਨੂੰ ਮੈਮਰੀ ਐਲੋਕੇਟ (Allocate) ਕਰਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ:

x = 12

ਇਸ ਉਦਾਹਰਣ ਵਿਚ ਵੈਰੀਏਬਲ 'x' ਨੂੰ ਇੰਟੀਜ਼ਰ (ਪੂਰਨ ਅੰਕ) ਵਜੋਂ ਡਿਕਲੇਅਰ ਕੀਤਾ ਗਿਆ ਹੈ (ਇੱਕ ਇੰਟੀਜ਼ਰ ਮੁੱਲ ਅਸਾਈਨ ਕਰਕੇ)। ਅਸੀਂ ਇਸ ਵੇਰੀਏਬਲ x ਨੂੰ ਕਿਸੇ ਹੋਰ ਕਿਸਮ ਵਿੱਚ ਵੀ ਬਦਲ ਸਕਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ ਇੱਕ ਸਟ੍ਰਿੰਗ ਵਿਚ (ਇੱਕ ਸਟ੍ਰਿੰਗ ਮੁੱਲ ਅਸਾਈਨ ਕਰਕੇ), ਜਿਵੇਂ ਕਿ:

x = 'Hello Students'

ਸਧਾਰਨ ਸ਼ਬਦਾਂ ਵਿੱਚ ਅਸੀਂ ਕਹਿ ਸਕਦੇ ਹਾਂ ਕਿ ਇੱਕ ਪਾਈਥਨ ਵੇਰੀਏਬਲ ਮੁੱਲ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਇੱਕ ਰਿਜ਼ਰਵਡ ਮੈਮਰੀ ਲੋਕੇਸ਼ਨ ਹੁੰਦੀ ਹੈ। ਜਦੋਂ ਅਸੀਂ ਵੇਰੀਏਬਲ ਨੂੰ ਮੁੱਲ ਅਸਾਈਨ ਕਰਦੇ ਹਾਂ ਤਾਂ ਵੇਰੀਏਬਲ ਡਿਕਲੇਰੇਸ਼ਨ ਆਪਣੇ ਆਪ ਹੋ ਜਾਂਦੀ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਕਿਸੇ ਵੇਰੀਏਬਲ ਨੂੰ ਵਰਤੇ ਜਾਣ ਤੋਂ ਪਹਿਲਾਂ ਹਮੇਸ਼ਾਂ ਮੁੱਲ ਅਸਾਈਨ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ, ਨਹੀਂ ਤਾਂ ਪਾਈਥਨ ਐਰਰ (Error) ਦਰਸਾਵੇਗਾ। ਪਾਈਥਨ ਵਿੱਚ ਅਸੀਂ ਨਵੇਂ ਵੇਰੀਏਬਲ ਬਣਾਉਣ ਲਈ ਅਸਾਈਨਮੈਂਟ ਓਪਰੇਟਰ (=) ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਖਾਸ ਮੁੱਲ ਅਸਾਈਨ ਕਰ ਸਕਦੇ ਹਾਂ।

ਵੇਰੀਏਬਲਜ਼ ਉਹ ਆਇਡੈਂਟੀਫਾਇਰਜ਼ ਹੁੰਦੇ ਹਨ ਜਿਹਨਾਂ ਨੂੰ ਮੁੱਲ ਸਟੋਰ ਕਰਨ ਲਈ (to store values) ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਅਸੀਂ ਰਨਟਾਈਮ ਦੌਰਾਨ ਇਹਨਾਂ ਦਾ ਮੁੱਲ ਬਦਲ ਵੀ ਸਕਦੇ ਹਾਂ। ਵੇਰੀਏਬਲ ਨਾਮ ਅਰਥਪੂਰਨ ਅਤੇ ਛੋਟਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।ਆਮ ਤੌਰ 'ਤੇ ਇਹਨਾਂ ਨੂੰ ਛੋਟੇ ਅੱਖਰਾਂ ਵਿੱਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ।

2.5.3.1 ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਮੁੱਲ ਅਸਾਈਨ ਕਰਨਾ (Assigning Values to Variables):

ਉਦਾਹਰਣ ਲਈ:

```
age =50                    # An integer assignment
distance =100.50           # A floating point assignment
student_name ="Param"      # A string assignment
```

ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ (=) ਦਾ ਖੱਬਾ ਪਾਸਾ ਵੇਰੀਏਬਲ ਦੇ ਨਾਮ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਅਤੇ ਸੱਜਾ ਪਾਸਾ ਉਸ ਮੁੱਲ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਜੋ ਵੇਰੀਏਬਲ ਵਿੱਚ ਸਟੋਰ ਕੀਤਾ ਜਾਵੇਗਾ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ 50, 100.50 ਅਤੇ "Param" ਮੁੱਲ ਕ੍ਰਮਵਾਰ age, distance ਅਤੇ student_name ਵੇਰੀਏਬਲਾਂ ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਗਿਆ ਹੈ।

2.5.3.2 ਮਲਟੀਪਲ ਅਸਾਈਨਮੈਂਟਸ (Multiple Assignments):

ਪਾਈਥਨ ਸਾਨੂੰ ਇੱਕੋ ਸਮੇਂ ਕਈ ਵੇਰੀਏਬਲਾਂ ਲਈ ਇੱਕ ਸਿੰਗਲ ਮੁੱਲ ਸਟੋਰ ਕਰਨ ਦੀ ਇਜਾਜ਼ਤ ਦਿੰਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ:

```
x = y = z = 100
```

ਇਸ ਉਦਾਹਰਣ ਵਿਚ 100 ਮੁੱਲ ਨਾਲ ਇੱਕ ਇੰਟੀਜ਼ਰ (ਪੂਰਨ ਅੰਕ) ਆਬਜੈਕਟ ਬਣਾਇਆ ਗਿਆ ਹੈ, ਅਤੇ ਤਿੰਨੇ ਵੇਰੀਏਬਲ (x, y, z) ਇੱਕੋ ਮੈਮਰੀ ਲੋਕੇਸ਼ਨ ਲਈ ਨਿਰਧਾਰਿਤ ਕੀਤੇ ਗਏ ਹਨ। ਅਸੀਂ ਮਲਟੀਪਲ ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਇਕੋ ਸਮੇਂ ਵੱਖ-ਵੱਖ ਆਬਜੈਕਟਸ ਵੀ ਅਸਾਈਨ ਕਰ ਸਕਦੇ ਹਾਂ। ਉਦਾਹਰਣ ਲਈ:

```
x, y, z = 50, 100, "Param"
```

ਇਸ ਉਦਾਹਰਣ ਵਿਚ ਮੁੱਲ 50 ਅਤੇ 100 ਵਾਲੇ ਦੋ ਇੰਟੀਜ਼ਰ ਆਬਜੈਕਟ ਕ੍ਰਮਵਾਰ x ਅਤੇ y ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਅਸਾਈਨ ਕੀਤੇ ਗਏ ਹਨ, ਅਤੇ ਮੁੱਲ "Param" ਵਾਲਾ ਇੱਕ ਸਟ੍ਰਿੰਗ ਆਬਜੈਕਟ ਵੇਰੀਏਬਲ z ਨੂੰ ਅਸਾਈਨ ਕੀਤਾ ਗਿਆ ਹੈ।

ਆਓ ਹੁਣ ਵੇਰੀਏਬਲ ਡਿਕਲੇਅਰ ਕਰਕੇ ਉਹਨਾਂ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਦਰਸਾਉਣ ਸਬੰਧੀ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰੀਏ:

ਚਿੱਤਰ 2.8: ਵੇਰੀਏਬਲ ਡਿਕਲੇਅਰ ਕਰਕੇ ਉਹਨਾਂ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਦਰਸਾਉਣ ਸਬੰਧੀ ਪ੍ਰੋਗਰਾਮ

ਹੁਣ F5 ਕੀਅ ਪ੍ਰੈਸ ਕਰਕੇ ਸਕ੍ਰਿਪਟ ਨੂੰ ਚਲਾਓ। ਇਹ ਅੱਗੇ ਦਿਖਾਏ ਅਨੁਸਾਰ ਆਊਟਪੁੱਟ ਦਰਸਾਏਗਾ:

ਆਉਟਪੁੱਟ:
```
101
Param Kansal
```

2.5 TEST Yourself — **ਹੇਠਾਂ ਦਿੱਤੇ ਗਏ ਸਟੇਟਮੈਂਟਸ ਅਨੁਸਾਰ ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟਸ ਲਿਖੋ:**

- shape ਵੇਰੀਏਬਲ ਨੂੰ 'circle' ਮੁੱਲ ਅਸਾਈਨ ਕਰੋ _____
- pi ਵੇਰੀਏਬਲ ਨੂੰ 3.14 ਮੁੱਲ ਅਸਾਈਨ ਕਰੋ _____
- length ਵੇਰੀਏਬਲ ਨੂੰ 7 ਅਤੇ breadth ਵੇਰੀਏਬਲ ਨੂੰ 5 ਮੁੱਲ ਅਸਾਈਨ ਕਰੋ _____
- num1 ਵੇਰੀਏਬਲ ਨੂੰ 2 ਅਤੇ num2 ਵੇਰੀਏਬਲ ਨੂੰ num+5 ਮੁੱਲ ਅਸਾਈਨ ਕਰੋ _____
- ਮਲਟੀਪਲ ਅਸਾਈਨਮੈਂਟ ਨਾਲ num=5 ਅਤੇ side=10 ਮੁੱਲ ਅਸਾਈਨ ਕਰੋ _____

2.5.4 ਪਾਈਥਨ ਵਿੱਚ ਕਮੈਂਟਸ ਦੀ ਵਰਤੋਂ (Use of Comments in Python):

ਕਮੈਂਟਸ ਪ੍ਰੋਗਰਾਮ ਦਾ ਅਨਖਿੜਵਾਂ ਅੰਗ (Integral part) ਹੁੰਦੇ ਹਨ। ਕਮੈਂਟ ਅਸਲ ਵਿੱਚ ਉਹ ਟੈਕਸਟ ਹੁੰਦਾ ਹੈ ਜੋ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਕੋਡ ਦੀ ਵਿਆਖਿਆ ਕਰਦਾ ਹੈ। ਕੰਪਾਈਲਰ ਅਤੇ ਇੰਟਰਪ੍ਰੈਟਰ ਇਹਨਾਂ ਕਮੈਂਟਸ ਨੂੰ ਨਜ਼ਰਅੰਦਾਜ਼ ਕਰਦੇ ਹਨ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਲਾਗੂ (Execute) ਨਹੀਂ ਕਰਦੇ।

ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਲਿਖੇ ਕੋਡ ਨੂੰ ਪੜ੍ਹਦੇ ਹੋਏ ਸਮਝਣ ਲਈ ਕਮੈਂਟਸ ਲਿਖਣੇ ਜ਼ਰੂਰੀ ਹੁੰਦੇ ਹਨ। ਇਹ ਕੋਡ ਨੂੰ ਸਮਝਣਾ ਸੌਖਾ ਬਣਾਉਂਦੇ ਹਨ। ਕਮੈਂਟਸ ਪ੍ਰੋਗਰਾਮਰ ਲਈ ਕੋਡ ਵਿੱਚ ਲਿਖੀਆਂ ਗੁੰਝਲਦਾਰ (Complex) ਚੀਜ਼ਾਂ ਨੂੰ ਯਾਦ ਰੱਖਣਾ ਆਸਾਨ ਬਣਾਉਂਦੇ ਹਨ। ਉਹਨਾਂ ਨੂੰ ਕੋਡ ਦੇ ਇੱਕ ਖਾਸ ਬਲਾਕ ਦੇ ਕੰਮ ਦੀ ਵਿਆਖਿਆ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ।

ਹਰੇਕ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਵਿੱਚ ਕਮੈਂਟ ਲਿਖਣ ਦੇ ਵੱਖ-ਵੱਖ ਤਰੀਕੇ ਹੁੰਦੇ ਹਨ। ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਕਮੈਂਟਸ ਲਿਖਣ ਦੇ ਮੁੱਖ ਤੌਰ 'ਤੇ ਦੋ ਤਰੀਕੇ ਹਨ, ਜੋ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਹਨ:

- **ਸਿੰਗਲ ਲਾਈਨ ਕਮੈਂਟਸ (Single Line Comments):** ਇਹ ਉਹ ਕਮੈਂਟਸ ਹੁੰਦੇ ਹਨ ਜਿਨ੍ਹਾਂ ਵਿੱਚ ਕਮੈਂਟ ਦਾ ਟੈਕਸਟ (Text of the comment) ਇੱਕ ਲਾਈਨ ਵਿੱਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਸਿੰਗਲ-ਲਾਈਨ ਕਮੈਂਟ ਲਿਖਣ ਲਈ ਸਾਨੂੰ ਕਮੈਂਟ ਤੋਂ ਪਹਿਲਾਂ # ਚਿੰਨ੍ਹ ਲਗਾਉਣਾ ਪੈਂਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ:

  # This is an example of a single comment

  ਅਜਿਹੇ ਕਮੈਂਟਸ ਨੂੰ ਇਕ ਨਵੀਂ ਲਾਈਨ ਵਿੱਚ ਲਿਖਣਾ ਜ਼ਰੂਰੀ ਨਹੀਂ ਹੁੰਦਾ। ਅਸੀਂ ਆਪਣੇ ਕੋਡ ਵਿੱਚ ਸਟੇਟਮੈਂਟ ਤੋਂ ਬਾਅਦ ਵੀ ਅਜਿਹੇ ਕਮੈਂਟਸ ਦਾਖਲ ਕਰ ਸਕਦੇ ਹਾਂ। ਉਦਾਹਰਣ ਲਈ:

  roll_no=101 ***#Variable holding Integer Literal***

  # ਚਿੰਨ੍ਹ ਤੋਂ ਬਾਅਦ ਲਿਖੀ ਗਈ ਕੋਈ ਵੀ ਚੀਜ਼ ਪਾਈਥਨ ਦੁਆਰਾ ਨਜ਼ਰਅੰਦਾਜ਼ (Ignore) ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸ ਨੂੰ ਲਾਗੂ (Execute) ਨਹੀਂ ਕੀਤਾ ਜਾਂਦਾ।
- **ਮਲਟੀਲਾਈਨ ਜਾਂ ਬਲਾਕ ਕਮੈਂਟਸ (Multiline or Block Comments):** ਇਹ ਉਹ ਕਮੈਂਟਸ ਹੁੰਦੇ

ਹਨ ਜਿਨ੍ਹਾਂ ਵਿੱਚ ਕਮੈਂਟ ਦਾ ਟੈਕਸਟ ਕਈ-ਲਾਈਨਾਂ ਵਿੱਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਕਮੈਂਟਸ ਉਸ ਸਮੇਂ ਲਾਭਦਾਇਕ ਹੁੰਦੇ ਹਨ ਜਦੋਂ ਕਮੈਂਟ ਟੈਕਸਟ ਇੱਕ ਲਾਈਨ ਵਿੱਚ ਫਿੱਟ ਨਹੀਂ ਹੁੰਦਾ ਜਿਸ ਕਾਰਨ ਕਮੈਂਟ ਨੂੰ ਇੱਕ ਤੋਂ ਵੱਧ ਲਾਈਨਾਂ ਵਿੱਚ ਲਿਖਣ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਮਲਟੀ-ਲਾਈਨ ਕਮੈਂਟਸ ਲਈ ਅਸੀਂ ਜਾਂ ਤਾਂ ਡਾਕ-ਸਟ੍ਰਿੰਗਸ (doc-strings) ਜਾਂ ਸਿੰਗਲ-ਲਾਈਨ ਕਮੈਂਟ ਦੇ # ਚਿੰਨ੍ਹ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

```
# This is a comment.
# This is a comment, too.
```

ਅਸੀਂ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ # ਚਿੰਨ੍ਹ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕਈ ਲਾਈਨਾਂ 'ਤੇ ਕਮੈਂਟ ਕਰਦੇ ਹੋਏ ਮਲਟੀਲਾਈਨ ਕਮੈਂਟ ਬਣਾ ਸਕਦੇ ਹਾਂ:

ਜੇਕਰ ਅਸੀਂ ਹਰੇਕ ਲਾਈਨ ਦੇ ਸ਼ੁਰੂ ਵਿੱਚ # ਚਿੰਨ੍ਹ ਲਿਖੇ ਬਿਨ੍ਹਾਂ ਮਲਟੀਲਾਈਨ ਕਮੈਂਟ ਬਨਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਅਸੀਂ ਤਿੰਨ ਡਬਲ ਕੋਟੇਸ਼ਨ ਚਿੰਨ੍ਹਾਂ (“““ ”””) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਡਾਕ-ਸਟ੍ਰਿੰਗ (doc-string) ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

```
"""This is a multiline
comment."""
```

ਕਮੈਂਟਸ (Comments) ਅਸਲ ਵਿੱਚ ਉਹ ਟੈਕਸਟ ਹੁੰਦਾ ਹੈ ਜੋ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਕੋਡ ਦਾ ਵਰਨਣ ਕਰਦਾ ਹੈ। ਕੰਪਾਈਲਰ ਅਤੇ ਇੰਟਰਪ੍ਰੈਟਰ ਇਹਨਾਂ ਕਮੈਂਟਸ ਨੂੰ ਨਜ਼ਰਅੰਦਾਜ਼ ਕਰਦੇ ਹਨ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਲਾਗੂ (Execute) ਨਹੀਂ ਕਰਦੇ।

2.5.5 ਪਾਈਥਨ ਵਿਚ ਇਨਪੁੱਟ ਅਤੇ ਆਉਟਪੁੱਟ ਸਟੇਟਮੈਂਟਸ (Input and Output Statements):

ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਤਾਂ ਹੀ ਉਪਯੋਗੀ ਸਾਬਿਤ ਹੁੰਦਾ ਹੈ ਜੇਕਰ ਉਹ ਬਾਹਰੀ ਸੰਸਾਰ ਨਾਲ ਸੰਚਾਰ (Communication with the outside world) ਕਰ ਸਕਣਯੋਗ ਹੋਵੇ। ਇਸ ਮੰਤਵ ਲਈ ਪ੍ਰੋਗਰਾਮ ਆਮ ਤੌਰ 'ਤੇ ਯੂਜ਼ਰ ਤੋਂ ਇਨਪੁਟ ਡਾਟਾ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਉਸ ਡਾਟਾ ਉਪਰ ਕੰਮ ਕਰਕੇ ਨਤੀਜਾ ਯੂਜ਼ਰ ਨੂੰ ਵਾਪਸ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਨਪੁੱਟ ਸਿੱਧਾ ਕੀਬੋਰਡ ਰਾਹੀਂ ਜਾਂ ਬਾਹਰੀ ਸਰੋਤਾਂ (External sources) ਜਿਵੇਂ ਕਿ ਫਾਈਲਾਂ ਜਾਂ ਡਾਟਾਬੇਸ ਤੋਂ ਪ੍ਰਾਪਤ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ ਅਤੇ ਆਉਟਪੁੱਟ ਨੂੰ ਸਿੱਧੇ ਮਾਨੀਟਰ ਸਕ੍ਰੀਨ 'ਤੇ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜਾਂ ਇਸਨੂੰ ਬਾਹਰੀ ਸਰੋਤਾਂ ਜਿਵੇਂ ਕਿ ਫਾਈਲਾਂ ਜਾਂ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਕੀਬੋਰਡ ਦੁਆਰਾ ਯੂਜ਼ਰ ਤੋਂ ਇਨਪੁੱਟ ਕਿਵੇਂ ਪ੍ਰਾਪਤ ਕਰਨੀ ਹੈ ਅਤੇ ਮਾਨੀਟਰ ਸਕ੍ਰੀਨ ਦੁਆਰਾ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਆਉਟਪੁੱਟ ਕਿਵੇਂ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨੀ ਹੈ, ਇਸ ਸੰਬੰਧੀ ਜਾਣਕਾਰੀ ਹੇਠਾਂ ਦਿਤੀ ਗਈ ਹੈ:

2.5.5.1 ਮਾਨੀਟਰ ਸਕਰੀਨ 'ਤੇ ਆਉਟਪੁੱਟ ਦਿਖਾਉਣਾ (Output on Monitor Screen):

ਯੂਜ਼ਰ ਤੋਂ ਡਾਟਾ ਪ੍ਰਾਪਤ ਕਰਨ ਤੋਂ ਇਲਾਵਾ, ਪ੍ਰੋਗਰਾਮ ਦੁਆਰਾ ਯੂਜ਼ਰ ਨੂੰ ਡਾਟਾ ਦਰਸਾਉਣ ਦੀ ਜ਼ਰੂਰਤ ਵੀ ਪੈਂਦੀ ਹੈ। ਅਸੀਂ print() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪਾਈਥਨ ਵਿੱਚ ਕਨਸੋਲ (Console) ਸਕ੍ਰੀਨ ਉਪਰ ਪ੍ਰੋਗਰਾਮ ਡਾਟਾ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰ ਸਕਦੇ ਹਾਂ। ਕਨਸੋਲ ਸਕ੍ਰੀਨ ਉਪਰ ਡਾਟਾ/ਆਬਜੈਕਟ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ print() ਫੰਕਸ਼ਨ ਵਿਚ ਕੌਮਿਆਂ ਨਾਲ ਵੱਖ ਕੀਤੇ ਗਏ ਆਰਗੂਮੈਂਟਾਂ ਦੀ ਇੱਕ ਸੂਚੀ (comma-separated list of arguments) ਪਾਸ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਕੁੱਝ ਆਮ ਵਰਤੇ ਜਾਂਦੇ ਆਰਗੂਮੈਂਟਸ ਨਾਲ print() ਫੰਕਸ਼ਨ ਦਾ ਸਿੰਟੈਕਸ ਹੇਠਾਂ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

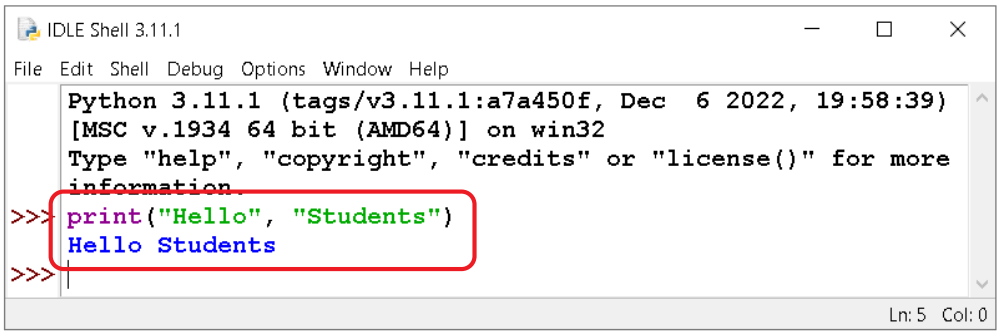
```
print(objects, sep, end)
```

ਇੱਥੇ, **objects** ਆਰਗੂਮੈਂਟ ਪ੍ਰਿੰਟ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲਾਂ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।

sep ਆਪਸ਼ਨਲ ਆਰਗੂਮੈਂਟ ਹੈ ਜੋ print() ਸਟੇਟਮੈਂਟ ਦੇ ਅੰਦਰ ਆਬਜੈਕਟਸ ਨੂੰ ਵੱਖ ਕਰਦਾ ਹੈ।

end ਆਪਸ਼ਨਲ ਆਰਗੂਮੈਂਟ ਹੈ ਜੋ ਵਿਸ਼ੇਸ਼ ਮੁੱਲ, ਜਿਵੇਂ ਕਿ: ਨਵੀਂ ਲਾਈਨ "\n", ਟੈਬ "\t" ਆਦਿ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।

ਉਦਾਹਰਣ ਲਈ:



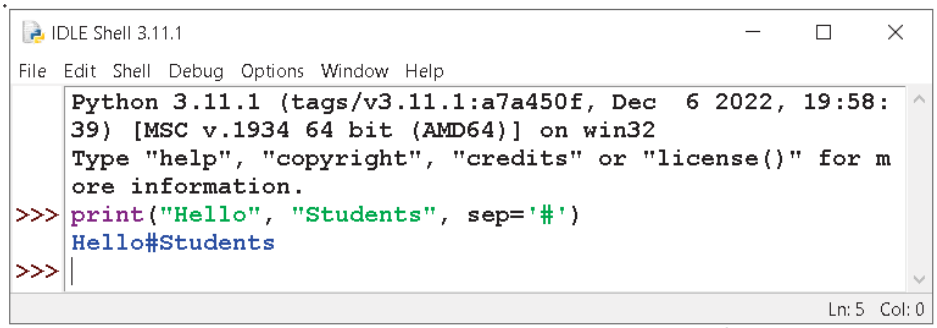
ਚਿੱਤਰ 2.9: ਪਾਈਥਨ ਵਿੱਚ print() ਸਟੇਟਮੈਂਟ ਦੀ ਸਧਾਰਨ ਉਦਾਹਰਣ

ਮੂਲ ਰੂਪ ਵਿੱਚ print() ਫੰਕਸ਼ਨ ਆਬਜੈਕਟਸ/ਮੁੱਲਾਂ ਨੂੰ ਸਿੰਗਲ ਸਪੇਸ ਦੁਆਰਾ ਵੱਖ ਕਰਕੇ ਦਰਸਾਉਂਦਾ ਹੈ ਅਤੇ ਆਉਟਪੁੱਟ ਦੇ ਅੰਤ ਵਿੱਚ ਇੱਕ ਨਵੀਂ ਲਈਨ ਦਾਖਲ ਕਰਦਾ ਹੈ। ਅਸੀਂ print() ਫੰਕਸ਼ਨ ਵਿਚ ਆਰਗੂਮੈਂਟ ਦੇ ਤੌਰ ਤੇ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੇ ਆਬਜੈਕਟ/ਮੁੱਲ ਨੂੰ ਪਾਸ ਕਰ ਸਕਦੇ ਹਾਂ। ਜੇਕਰ ਕੋਈ ਆਬਜੈਕਟ/ਮੁੱਲ ਸਟ੍ਰਿੰਗ ਕਿਸਮ ਦਾ ਨਹੀਂ ਹੈ, ਤਾਂ print() ਫੰਕਸ਼ਨ ਉਸਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਢੁਕਵੇਂ ਸਟ੍ਰਿੰਗ ਵਿੱਚ ਬਦਲਦਾ ਹੈ।

ਅਸੀਂ print() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪਾਈਥਨ ਵਿੱਚ ਕਨਸੋਲ (Console) ਸਕ੍ਰੀਨ ਉਪਰ ਪ੍ਰੋਗਰਾਮ ਡਾਟਾ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰ ਸਕਦੇ ਹਾਂ।

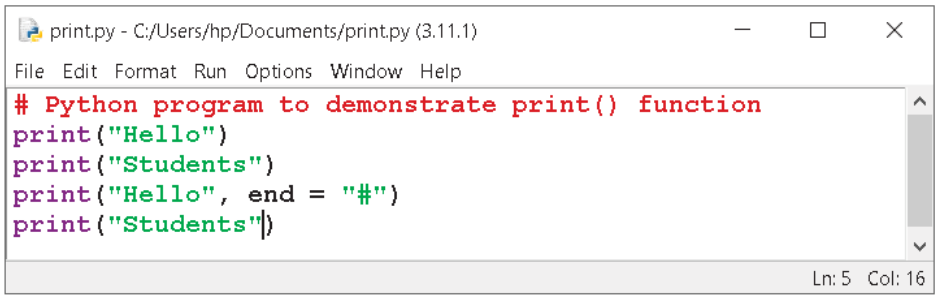
ਇੱਕ ਫੰਕਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਕਿਸੇ ਖਾਸ ਕੰਮ ਨੂੰ ਕਰਨ ਲਈ ਪਰਿਭਾਸ਼ਿਤ ਕੋਡ ਦਾ ਇੱਕ ਬਲਾਕ ਹੁੰਦਾ ਹੈ। ਇੱਕ ਫੰਕਸ਼ਨ ਉਦੋਂ ਚਲਦਾ ਹੈ ਜਦੋਂ ਇਸਨੂੰ ਬੁਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਅਸੀਂ ਡੇਟਾ ਨੂੰ ਇੱਕ ਫੰਕਸ਼ਨ ਵਿੱਚ ਪਾਸ ਕਰ ਸਕਦੇ ਹਾਂ, ਜਿਸਨੂੰ ਪੈਰਾਮੀਟਰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਫੰਕਸ਼ਨ ਨਤੀਜੇ ਵਜੋਂ ਡੇਟਾ ਵਾਪਸ ਕਰ ਸਕਦਾ ਹੈ।
ਉਦਾਹਰਣ ਲਈ : print ("hello");
ਇਸ ਉਦਾਹਰਣ ਵਿਚ print ਇਕ ਫੰਕਸ਼ਨ ਹੈ ਜੋ ਇਸ ਵਿਚ ਪੈਰਾਮੀਟਰ (i.e. "Hello") ਵੱਜੋਂ ਦਿੱਤੇ ਮੁੱਲ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ।

ਪ੍ਰਿੰਟ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲਾਂ ਨੂੰ ਵੱਖ ਕਰਨਾ (Seperating Printed Values): ਅਸੀਂ **sep** ਕੀਵਰਡ ਦੀ ਵਰਤੋਂ ਨਾਲ ਕਿਸੇ ਵੀ ਆਰਬਿਟਰਰੀ (Arbitrary) ਸਟ੍ਰਿੰਗ ਨੂੰ ਸੈਪਰੇਟਰ ਵਜੋਂ ਸੈੱਟ ਕਰ ਸਕਦੇ ਹਾਂ। **sep** ਕੀਵਰਡ ਦੀ ਵਰਤੋਂ ਇਹ ਦੱਸਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ print ਸਟੇਟਮੈਂਟ ਕਿਸ ਤਰ੍ਹਾਂ ਆਬਜੈਕਟਸ/ਮੁੱਲਾਂ ਨੂੰ ਸੈਪਰੇਟ (separate) ਕਰਕੇ ਦਿਖਾਵੇ। ਮੂਲ ਰੂਪ (By default) ਵਿੱਚ **sep="** ਹੁੰਦਾ ਹੈ ਜੋ ਆਬਜੈਕਟਸ/ਮੁੱਲਾਂ ਨੂੰ ਸਿੰਗਲ ਸਪੇਸ ਨਾਲ ਸੈਪਰੇਟ ਕਰਦਾ ਹੈ। ਪ੍ਰਿੰਟ ਸਟੇਟਮੈਂਟ ਵਿਚ ਆਰਗੂਮੈਂਟ **sep=<str>** ਲਿਖਣ ਨਾਲ ਪਾਈਥਨ ਡਿਫਾਲਟ ਸਿੰਗਲ ਸਪੇਸ ਦੀ ਬਜਾਏ ਦਿੱਤੇ ਗਏ **<str>** ਨਾਲ ਆਬਜੈਕਟਸ/ਮੁੱਲਾਂ ਨੂੰ ਸੈਪਰੇਟ ਕਰਕੇ ਦਿਖਾਵੇਗਾ। ਇਸਦੀ ਵਰਤੋਂ ਸੰਬੰਧੀ ਉਦਾਹਰਣ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:



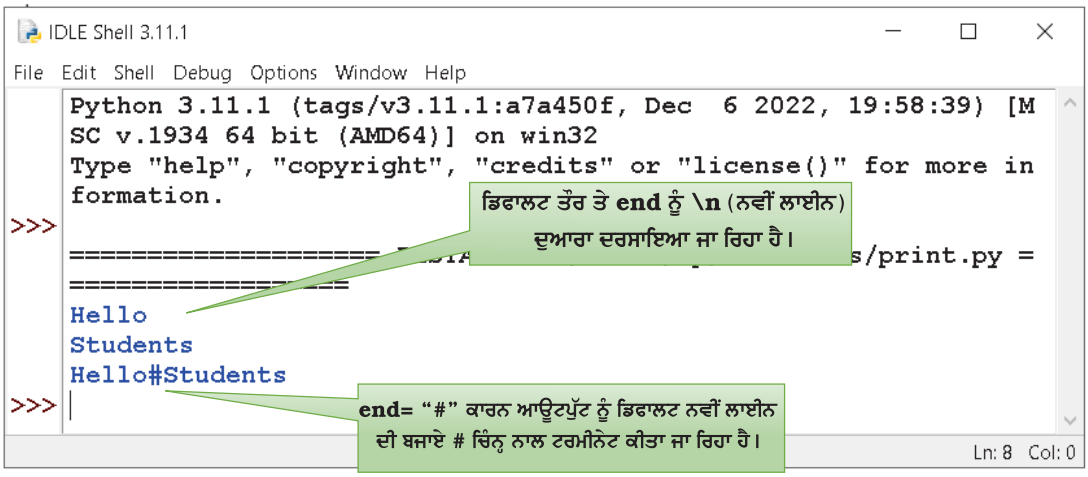
ਚਿੱਤਰ 2.10: ਪਾਈਥਨ ਵਿੱਚ sep ਕੀਵਰਡ ਨਾਲ print() ਸਟੇਟਮੈਂਟ ਦੀ ਉਦਾਹਰਨ

ਨਿਊਲਾਈਨ ਕਰੈਕਟਰ ਨੂੰ ਕੰਟਰੋਲ ਕਰਨਾ (Controlling Newline Character): **end** ਆਰਗੂਮੈਂਟ ਦੀ ਵਰਤੋਂ print ਸਟੇਟਮੈਂਟ ਦੁਆਰਾ ਸਾਰੇ ਮੁੱਲਾਂ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਤੋਂ ਬਾਅਦ ਕਿਸੇ ਖਾਸ ਚੀਜ਼ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਮੂਲ ਰੂਪ (By default) ਵਿੱਚ, **end** ਦਾ ਮੁੱਲ \n ਹੁੰਦਾ ਹੈ, ਜੋ ਹਰੇਕ print() ਸਟੇਟਮੈਂਟ ਤੋਂ ਬਾਅਦ ਇੱਕ ਨਵੀਂ ਲਾਈਨ ਦਾਖਲ ਕਰਦਾ ਹੈ। ਪ੍ਰਿੰਟ ਸਟੇਟਮੈਂਟ ਵਿਚ ਆਰਗੂਮੈਂਟ **end=<str>** ਲਿਖਣ ਨਾਲ ਪਾਈਥਨ ਡਿਫਾਲਟ ਨਵੀਂ ਲਾਈਨ ਦੀ ਬਜਾਏ **<str>** ਦੁਆਰਾ ਆਉਟਪੁੱਟ ਨੂੰ ਖਤਮ ਕਰੇਗਾ। ਇਸਦੀ ਵਰਤੋਂ ਸੰਬੰਧੀ ਉਦਾਹਰਨ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:



ਚਿੱਤਰ 2.11: end ਕੀਵਰਡ ਨਾਲ print() ਸਟੇਟਮੈਂਟ ਦੀ ਉਦਾਹਰਨ (print.py)

ਸਕ੍ਰਿਪਟ ਫਾਈਲ (print.py) ਨੂੰ ਚਲਾਉਣ ਤੋਂ ਬਾਅਦ ਯੂਜ਼ਰ ਨੂੰ ਅੱਗੇ ਦਿੱਤੀ ਗਈ ਆਉਟਪੁੱਟ ਦਿਖਾਈ ਦੇਵੇਗੀ ਜੋ print() ਸਟੇਟਮੈਂਟ ਵਿੱਚ end ਕੀਵਰਡ ਦੀ ਵਰਤੋਂ ਦੇ ਮੰਤਵ ਨੂੰ ਸਪਸ਼ਟ ਰੂਪ ਵਿਚ ਬਿਆਨ ਕਰਦੀ ਹੈ।



ਚਿੱਤਰ 2.12: ਸਕ੍ਰਿਪਟ ਫਾਈਲ (print.py) ਦਾ ਆਉਟਪੁੱਟ

TEST yourself

2.6

ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦੀ ਆਉਟਪੁੱਟ ਲਿਖੋ:

a) a = 4
b = 10
print (a, b)

ਆਉਟਪੁੱਟ (a):

b) x, y = 2, 6
x, y = y, x
print (x, y)

ਆਉਟਪੁੱਟ (b):

2.5.5.2 ਕੀਬੋਰਡ ਤੋਂ ਇਨਪੁੱਟ ਪ੍ਰਾਪਤ ਕਰਨਾ (Getting Input from the Keyboard):

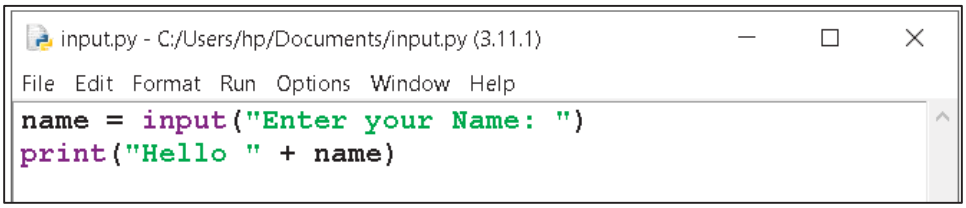
ਪ੍ਰੋਗਰਾਮਾਂ ਵਿਚ ਆਮ ਤੌਰ 'ਤੇ ਕੀਬੋਰਡ ਰਾਹੀਂ ਯੂਜ਼ਰ ਤੋਂ ਡਾਟਾ ਪ੍ਰਾਪਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਕੰਮ ਲਈ ਪਾਈਥਨ ਵਿੱਚ input() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਜਿਸਦਾ ਸਿੰਟੈਕਸ ਇਸ ਪ੍ਰਕਾਰ ਹੈ:

```
input("prompt")
```

ਇੱਥੇ prompt ਆਪਸ਼ਨਲ ਆਰਗੂਮੈਂਟ ਹੈ। ਇਹ ਇੱਕ ਮੈਸੇਜ਼ (message) ਹੁੰਦਾ ਹੈ ਜੋ ਅਸੀਂ ਯੂਜ਼ਰ ਤੋਂ ਇਨਪੁੱਟ ਲੈਂਦੇ ਸਮੇਂ ਯੂਜ਼ਰ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨਾ ਚਾਹੁੰਦਾ ਹਾਂ।

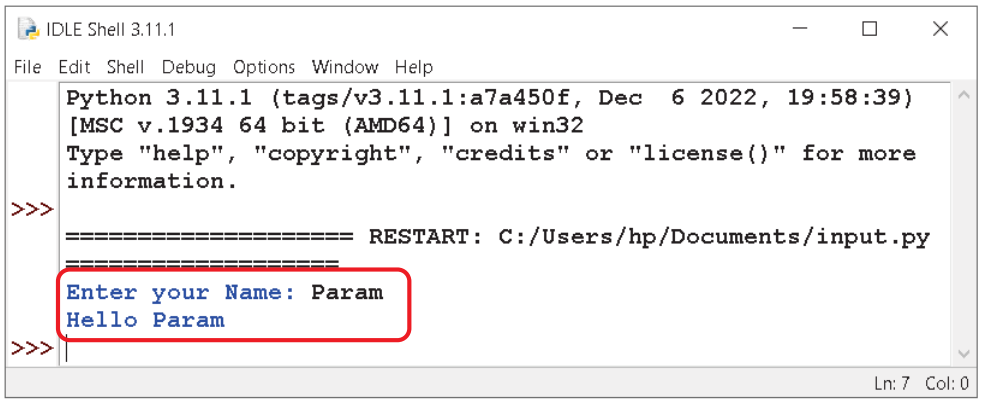
ਪਾਈਥਨ ਵਿੱਚ, ਕੀਬੋਰਡ ਰਾਹੀਂ ਯੂਜ਼ਰ ਤੋਂ ਪ੍ਰੋਗਰਾਮ ਡਾਟਾ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ input() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਫੰਕਸ਼ਨ ਇਨਪੁਟ ਕੀਤੇ ਗਏ ਡਾਟਾ ਨੂੰ ਸਟ੍ਰਿੰਗ ਰੂਪ ਵਿਚ ਵਾਪਸ (return) ਕਰਦਾ ਹੈ।

input() ਫੰਕਸ਼ਨ ਯੂਜ਼ਰ ਨੂੰ ਕੀਬੋਰਡ ਰਾਹੀਂ ਸਿੰਗਲ ਲਾਈਨ ਵਿਚ ਇਨਪੁੱਟ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ਐਗਜ਼ੀਕਿਊਸ਼ਨ (Execution) ਨੂੰ ਰੋਕਦਾ (Pause) ਹੈ। ਇੱਕ ਵਾਰ ਯੂਜ਼ਰ ਦੁਆਰਾ ਐਂਟਰ ਕੀਅ ਦਬਾਉਣ ਤੋਂ ਬਾਅਦ, ਟਾਈਪ ਕੀਤੇ ਸਾਰੇ ਅੱਖਰ ਇਕ ਸਟ੍ਰਿੰਗ ਦੇ ਰੂਪ ਵਿਚ ਪ੍ਰੋਗਰਾਮ ਦੁਆਰਾ ਪੜ੍ਹ (read) ਲਏ ਜਾਂਦੇ ਹਨ। ਉਦਾਹਰਨ ਲਈ: IDLE ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਇੱਕ ਪਾਈਥਨ ਸਕ੍ਰਿਪਟ (script) ਫਾਈਲ ਤਿਆਰ ਕਰੋ ਅਤੇ ਹੇਠ ਲਿਖਿਆ ਕੋਡ (code) ਟਾਈਪ ਕਰੋ:



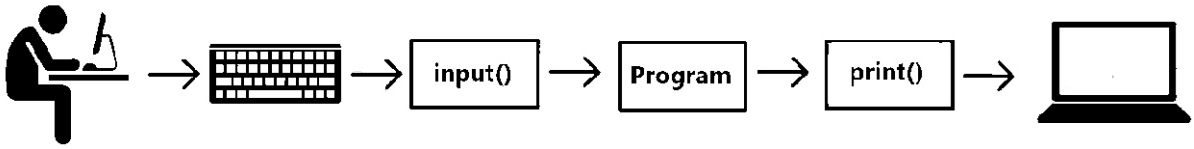
ਚਿੱਤਰ 2.13: ਪਾਈਥਨ ਸਕ੍ਰਿਪਟ ਫਾਈਲ (input.py) ਵਿਚ input() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ

ਹੁਣ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਨੂੰ ਚਲਾਉਣ ਲਈ F5 ਕੀਅ ਦਬਾਓ। ਇਹ ਯੂਜ਼ਰ ਨੂੰ ਹੇਠਾਂ ਦਰਸਾਏ ਅਨੁਸਾਰ ਉਸਦਾ ਨਾਮ ਇਨਪੁੱਟ ਕਰਨ ਲਈ ਪੁੱਛੇਗਾ। ਯੂਜ਼ਰ ਨੂੰ ਹੇਠਾਂ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਦਿਖਾਈ ਦੇਵੇਗੀ:



ਚਿੱਤਰ 2.14: ਪਾਈਥਨ ਸਕ੍ਰਿਪਟ ਫਾਈਲ (input.py) ਦਾ ਆਉਟਪੁੱਟ

input() ਫੰਕਸ਼ਨ ਹਮੇਸ਼ਾ ਯੂਜ਼ਰ ਤੋਂ ਇਨਪੁੱਟ ਪ੍ਰਾਪਤ ਕਰਕੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸਟ੍ਰਿੰਗ ਰੂਪ ਵਿਚ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਇੱਕ ਸੰਖਿਆਤਮਕ (numeric) ਰੂਪ ਵਿਚ ਇਨਪੁੱਟ ਪ੍ਰਾਪਤ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਸਾਨੂੰ ਬਿਲਟ-ਇਨ ਫੰਕਸ਼ਨਾਂ int(), float(), ਜਾਂ complex() ਆਦਿ ਫੰਕਸ਼ਨਾਂ ਦੀ ਮਦਦ ਨਾਲ ਇਨਪੁੱਟ ਸਟ੍ਰਿੰਗ ਨੂੰ ਉਚਿਤ ਕਿਸਮ ਵਿੱਚ ਬਦਲਣ ਦੀ ਜ਼ਰੂਰਤ ਪੈਂਦੀ ਹੈ।



ਚਿੱਤਰ 2.15: ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਿੱਚ ਇਨਪੁੱਟ ਅਤੇ ਆਊਟਪੁੱਟ ਸਟੇਟਮੈਂਟਾਂ ਦੀ ਵਰਤੋਂ ਕੀ ਧਾਰਨਾ

2.7 Test Yourself

ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜੀਆਂ ਇਨਪੁੱਟ ਸਟੇਟਮੈਂਟਸ ਸਹੀ ਹਨ ?

i. a = input ( ) _____

ii. b = input ("Enter a number") _____

iii. c = input (Enter your name) _____

2.5.6 ਮਲਟੀਪਲ ਲਾਈਨ ਸਟੇਟਮੈਂਟਸ (Multiple Line Statements):

ਪਾਈਥਨ ਵਿੱਚ ਸਟੇਟਮੈਂਟਾਂ ਆਮ ਤੌਰ 'ਤੇ ਇੱਕ ਨਵੀਂ ਲਾਈਨ ਸ਼ੁਰੂ ਕਰਨ ਨਾਲ ਖਤਮ ਹੁੰਦੀਆਂ ਹਨ। ਜੇਕਰ ਕੁੱਝ ਸਟੇਟਮੈਂਟਸ ਜ਼ਿਆਦਾ ਲੰਮੀਆਂ ਹੋਣ ਅਤੇ ਅਸੀਂ ਉਹਨਾਂ ਨੂੰ ਇੱਕ ਲਾਈਨ ਵਿੱਚ ਠੀਕ ਤਰ੍ਹਾਂ ਫਿੱਟ ਨਹੀਂ ਕਰ ਸਕਦੇ ਹੋਈਏ, ਤਾਂ ਮਲਟੀਪਲ-ਲਾਈਨ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਨਿਰੰਤਰਤਾ ਅੱਖਰ (Continuation Character) ਬੈਕਸਲੈਸ਼ (\) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਅਸੀਂ ਇਕ ਲੰਬੀ ਸਟੇਟਮੈਂਟ ਨੂੰ ਕਈ ਲਾਈਨਾਂ ਵਿੱਚ ਵੰਡ ਕੇ ਲਿਖ ਸਕਦੇ ਹਾਂ। ਲਾਈਨ ਨਿਰੰਤਰਤਾ ਅੱਖਰ (\) (line continuation character) ਦੀ ਵਰਤੋਂ ਇਹ ਦਰਸਾਉਂਦੀ ਹੈ ਕਿ ਮੌਜੂਦਾ ਸਟੇਟਮੈਂਟ ਅਗਲੀ ਲਾਈਨ ਵਿਚ ਵੀ ਜਾਰੀ ਰਹੇਗੀ। ਹੇਠਾਂ ਇਸਦੀ ਵਰਤੋਂ ਸੰਬੰਧੀ ਉਦਾਹਰਨ ਦਿੱਤੀ ਗਈ ਹੈ:

```
>>> print("This is a long statement that \
... will be written in multiple lines, because \
... it doesn't fit in one single line. ")
...

This is a long statement that will be written in multiple lines, because it
doesn't fit in one single line.
```

Continuation character backslash (\) ਨਾਲ ਮਲਟੀਪਲ ਲਾਈਨ ਸਟੇਟਮੈਂਟ

ਮਲਟੀਪਲ-ਲਾਈਨ ਸਟੇਟਮੈਂਟ ਦੀ ਆਊਟਪੁੱਟ

ਚਿੱਤਰ 2.16: ਮਲਟੀਪਲ ਲਾਈਨ ਸਟੇਟਮੈਂਟ ਦੀ ਉਦਾਹਰਨ

ਅਸੀਂ ਸੈਮੀਕੋਲਨ (;) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਇੱਕ ਲਾਈਨ ਵਿੱਚ ਕਈ ਸਟੇਟਮੈਂਟਾਂ ਵੀ ਲਿਖ ਸਕਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

```
a =1;b = 2;c =3
```

[ ], {}, ਜਾਂ ( ) ਬਰੈਕਟਾਂ ਵਿੱਚ ਸ਼ਾਮਲ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਲਾਈਨ ਨਿਰੰਤਰਤਾ ਅੱਖਰ (line continuation character (\)) ਦੀ ਵਰਤੋਂ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਪੈਂਦੀ। ਉਦਾਹਰਣ ਲਈ:

```
days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday']
```

2.5.7 ਪਾਈਥਨ ਵਿੱਚ ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹ (Quotation in Python):

ਪਾਈਥਨ ਵਿੱਚ ਸਟਰਿੰਗ ਲਿਟਰਲ (Literal) ਲਈ ਅੱਖਰਾਂ ਦੇ ਕ੍ਰਮ (Sequence of Characters) ਨੂੰ ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹਾਂ (Quotatoin Marks) ਵਿਚਕਾਰ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਵਿਚ ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹਾਂ ਚਿੰਨ੍ਹਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸਟ੍ਰਿੰਗ ਲਿਟਰਲ ਲਿਖਣ ਦੇ ਤਿੰਨ ਤਰੀਕੇ ਹਨ:

- ਇਕਹਿਰੇ ਕੁਟੇਸ਼ਨ (Single Quotation) ਚਿੰਨ੍ਹ (‘), ਉਦਾਹਰਣ ਲਈ: ‘hello’
- ਦੁਹਰੇ ਕੁਟੇਸ਼ਨ (Double Quotation) ਚਿੰਨ੍ਹ (“), ਉਦਾਹਰਣ ਲਈ: “hello”
- ਤੀਹਰੇ ਕੁਟੇਸ਼ਨ (Triple Quotation) ਚਿੰਨ੍ਹ (‘‘‘ or """), ਉਦਾਹਰਣ ਲਈ: """hello"""

ਅਸੀਂ ਪਹਿਲਾਂ ਹੀ ਪਾਈਥਨ ਵਿੱਚ ਕਮੈਂਟਸ (Comments) ਦੀ ਵਰਤੋਂ ਵਾਲੇ ਸੈਕਸ਼ਨ ਵਿੱਚ ਤੀਹਰੇ ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹਾਂ ਦੀ ਵਰਤੋਂ ਵੇਖ ਚੁੱਕੇ ਹਾਂ। ਟ੍ਰਿਪਲ ਕੋਟਸ ਆਮ ਤੌਰ 'ਤੇ ਕਈ ਲਾਈਨਾਂ ਵਿੱਚ ਸਟ੍ਰਿੰਗ ਨੂੰ ਲਿਖਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਕਹਿਰੇ ਅਤੇ ਦੋਹਰੇ ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹ ਸਿੰਗਲ-ਲਾਈਨ ਵਿੱਚ ਸਟ੍ਰਿੰਗ ਲਿਖਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇੱਥੇ ਇੱਕ ਗੱਲ ਧਿਆਨ ਦੇਣ ਯੋਗ ਇਹ ਹੈ ਕਿ ਸਟਰਿੰਗ ਲਿਟਰਲ ਲਈ ਓਪਨਿੰਗ (Opening) ਅਤੇ ਕਲੋਜ਼ਿੰਗ (Closing) ਕੋਟੇਸ਼ਨ ਚਿੰਨ੍ਹ ਇੱਕੋ ਕਿਸਮ (same type) ਦੇ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ।

ਉਦਾਹਰਣ ਲਈ:

```
str1 = 'Hello'
str2 = "Hello Students! How are you?"
str3 = """It is a paragraph.
        It is made up of multiple lines and sentences,"""
```

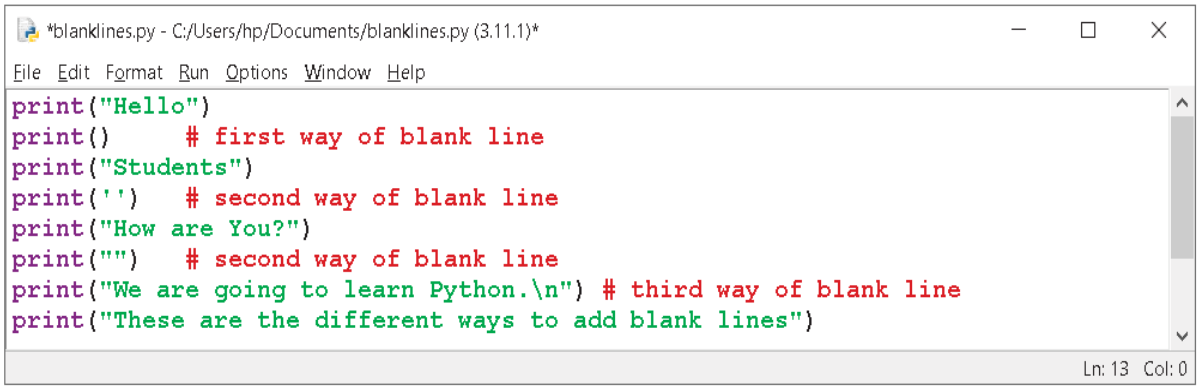
ਪਾਈਥਨ ਵਿਚ ਸਟ੍ਰਿੰਗ ਲਿਟਰਲਜ਼ ਲਿਖਣ ਲਈ ਸਿੰਗਲ (' '), ਡਬਲ (" ") ਅਤੇ ਟ੍ਰਿਪਲ (""" """ """) ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

2.5.8 ਪਾਈਥਨ ਵਿੱਚ ਖਾਲੀ ਲਾਈਨਾਂ (Blank Lines in Python):

ਸਿਰਫ਼ ਖਾਲੀ ਥਾਂ (only whitespace) ਵਾਲੀ ਲਾਈਨ ਨੂੰ ਖਾਲੀ ਲਾਈਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਖਾਲੀ ਲਾਈਨਾਂ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਦੇ ਤਿੰਨ ਤਰੀਕੇ ਹਨ:

- ਪਹਿਲਾ ਅਤੇ ਸਰਲ ਤਰੀਕਾ ਹੈ ਖਾਲੀ ਪ੍ਰਿੰਟ ਸਟੇਟਮੈਂਟ (blank print statement) ਦੀ ਵਰਤੋਂ ਕਰਨਾ।
- ਦੂਜਾ ਤਰੀਕਾ ਹੈ ਪ੍ਰਿੰਟ ਸਟੇਟਮੈਂਟ ਵਿੱਚ ਖਾਲੀ ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹ ਲਗਾਉਣਾ – ਸਿੰਗਲ ਜਾਂ ਡਬਲ।
- ਤੀਜਾ ਤਰੀਕਾ ਹੈ ਇੱਕ ਨਿਊ ਲਾਈਨ ਕਰੈਕਟਰ (\n) ਦੀ ਵਰਤੋਂ ਕਰਨਾ।

ਹੇਠ ਦਿੱਤਾ ਪਾਈਥਨ ਕੋਡ ਖਾਲੀ ਲਾਈਨਾਂ ਸੰਬੰਧੀ ਉਦਾਹਰਣਾਂ ਨੂੰ ਦਰਸਾ ਰਿਹਾ ਹੈ



ਚਿੱਤਰ 2.17: ਆਉਟਪੁੱਟ ਵਿੱਚ ਖਾਲੀ ਲਾਈਨਾਂ ਪਾਉਣ ਲਈ ਸਕ੍ਰਿਪਟ (blanklines.py) ਫਾਈਲ

ਉਪਰੋਕਤ ਸਕ੍ਰਿਪਟ (blanklines.py) ਫਾਈਲ ਨੂੰ ਚਲਾਉਣ ਤੋਂ ਬਾਅਦ ਹੇਠਾਂ ਦਿੱਤੀ ਆਉਪੁੱਟ ਦਿਖਾਈ ਦੇਵੇਗੀ:

```
Hello

Students

How are You?

We are going to learn Python.

These are the different ways to add blank lines
```

ਅਭਿਆਸ

ੳ. ਬਹੁਪਸੰਦੀ ਪ੍ਰਸ਼ਨ

1. _____ ਇੱਕ ਡਿਫਾਲਟ ਐਡੀਟਰ ਹੈ ਜੋ ਪਾਈਥਨ ਨਾਲ ਆਉਂਦਾ ਹੈ।

ੳ. IDLE
ਅ. IPLE
ੲ. Text Editor
ਸ. Notepad

2. _____ ਸ਼ੈੱਲ ਪ੍ਰੋਂਪਟ ਹੈ ਜਿੱਥੇ ਅਸੀਂ ਆਪਣੀਆਂ ਕਮਾਂਡਾਂ ਟਾਈਪ ਕਰਦੇ ਹਾਂ।

ੳ. < <
ਅ. > >
ੲ. > > >
ਸ. < < <

3. IDE ਦਾ ਪੂਰਾ ਨਾਂ ਹੈ _____

ੳ. ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਾਇਰੈਕਟ ਇਨਵਾਇਰਨਮੈਂਟ
ਅ. ਇੰਟੀਗ੍ਰੇਟਿਡ ਡੇਵਲਪਮੈਂਟ ਇਨਵਾਇਰਨਮੈਂਟ
ੲ. ਇਨਫਾਰਮੇਸ਼ਨ ਡੇਵਲਪਮੈਂਟ ਇਨਵਾਇਰਨਮੈਂਟ
ਸ. ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ

4. ਪਾਈਥਨ ਵਿੱਚ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਕੋਡ ਚਲਾਉਣ ਲਈ ਅਸੀਂ _____ ਸ਼ਾਰਟਕੱਟ ਕੀਅ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

ੳ. F1
ਅ. F2
ੲ. F5
ਸ. F7

5. _____ ਅੰਗਰੇਜ਼ੀ ਭਾਸ਼ਾ ਦੇ ਸ਼ਬਦਾਂ ਅਤੇ ਵਿਰਾਮ ਚਿੰਨ੍ਹਾਂ ਵਾਂਗ ਹੁੰਦੇ ਹਨ।

ੳ. ਲਿਟਰਲਜ਼
ਅ. ਆਇਡੈਂਟੀਫਾਇਰਜ਼
ੲ. ਵੇਰੀਏਬਲਜ਼
ਸ. ਟੋਕਨਜ਼

6. _____ ਪ੍ਰੋਗਰਾਮ ਐਲੀਮੈਂਟਸ (ਜਿਵੇਂ ਕਿ: ਵੇਰੀਏਬਲ, ਫੰਕਸ਼ਨ, ਲਿਸਟ, ਟਪਲਸ, ਆਦਿ) ਨੂੰ ਉਹਨਾਂ ਦੀ ਪਛਾਣ ਲਈ ਦਿੱਤੇ ਗਏ ਨਾਮ ਹੁੰਦੇ ਹਨ।

ੳ. ਲਿਟਰਲਜ਼
ਅ. ਆਇਡੈਂਟੀਫਾਇਰਜ਼
ੲ. ਵੇਰੀਏਬਲਜ਼
ਸ. ਟੋਕਨਜ਼

7. ਲਿਟਰਲਜ਼ ਸੋਰਸ ਕੋਡ ਵਿੱਚ ਵਰਤੇ ਗਏ _____ ਮੁੱਲ ਹੁੰਦੇ ਹਨ।

ੳ. ਸਥਿਰ
ਅ. ਬੂਲੀਅਨ
ੲ. ਸਟ੍ਰਿੰਗ
ਸ. ਫਲੋਟ

8. _____ ਉਹ ਆਈਡੈਂਟੀਫਾਇਰ ਹੁੰਦੇ ਹਨ ਜੋ ਮੁੱਲਾਂ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਅਸੀਂ ਰਨ ਟਾਈਮ ਦੌਰਾਨ ਉਹਨਾਂ ਦੇ ਮੁੱਲ ਨੂੰ ਬਦਲ ਵੀ ਸਕਦੇ ਹਾਂ।

ੳ. ਕਾਂਸਟੈਂਟ
ਅ. ਵੇਰੀਏਬਲ
ੲ. ਲਿਸਟ
ਸ. ਟੋਕਨਜ਼

9. ਕੁਮੈਂਟ ਮੂਲ ਰੂਪ ਵਿੱਚ ਇੱਕ ਟੈਕਸਟ ਹੁੰਦਾ ਹੈ ਜੋ ਪ੍ਰੋਗਰਾਮ ਕੋਡ ਦੀ _____ ਕਰਦੇ ਹਨ।

ੳ. ਐਗਜ਼ੀਕਿਊਸ਼ਨ
ਅ. ਕੰਪਾਈਲੇਸ਼ਨ
ੲ. ਵਿਆਖਿਆ
ਸ. ਉਪਰੋਕਤ ਬਾਰੇ

10. ਅਸੀਂ _____ ਫੰਕਸ਼ਨ ਦੇ ਨਾਲ ਪਾਈਥਨ ਵਿੱਚ ਕਨਸੋਲ ਉੱਪਰ ਪ੍ਰੋਗਰਾਮ ਡਾਟਾ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰ ਸਕਦੇ ਹਾਂ।

ੳ. input()　　ਅ. print()

ੲ. output()　　ਸ. show()

ਅ. ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

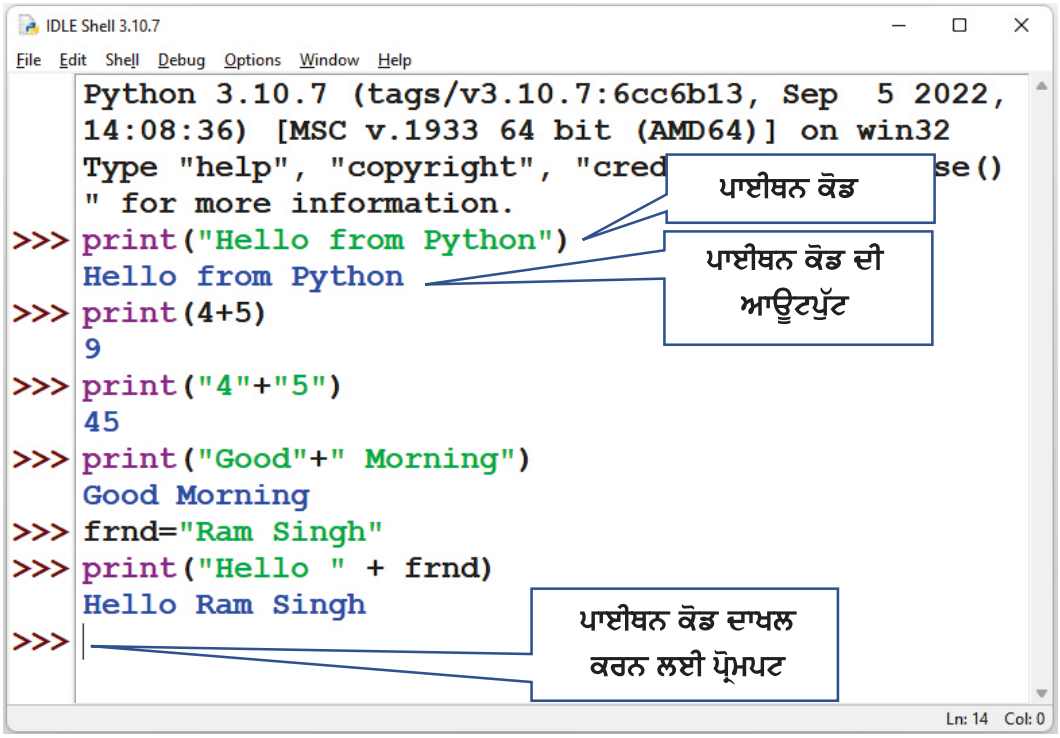
1. ਤੁਸੀਂ ਪਾਈਥਨ (Pyton) ਬਾਰੇ ਕੀ ਜਾਣਦੇ ਹੋ ?
2. IDE ਕੀ ਹੈ ?
3. ਤੁਸੀਂ ਪਾਈਥਨ ਸ਼ੈੱਲ ਦੀ ਵਰਤੋਂ ਨਾਲ ਕੀਵਰਡਜ਼ (Keywords) ਦੀ ਲਿਸਟ ਕਿਸ ਤਰ੍ਹਾਂ ਦੇਖ ਸਕਦੇ ਹੋ ? ਕੀਵਰਡਜ਼ ਦੀਆਂ ਕੁੱਝ ਉਦਾਹਰਣਾਂ ਲਿਖੋ।
4. ਤੁਸੀਂ ਕਿਸ ਤਰ੍ਹਾਂ ਪਾਈਥਨ ਵਿੱਚ ਵੇਰੀਏਬਲਜ਼ (Variables) ਨੂੰ ਡਿਕਲੇਅਰ ਕਰੋਗੇ ? ਉਦਾਹਰਣਾਂ ਦਿਓ।
5. ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਜ਼ ਵਿਚ print() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਿਉਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ?

ੲ. ਵੱਡੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

1. ਟੋਕਨਜ਼ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋ ? ਪਾਈਥਨ ਵਿਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਵੱਖ-ਵੱਖ ਟੋਕਨਜ਼ ਦਾ ਵਰਨਣ ਕਰੋ।
2. ਆਈਡੈਂਟੀਫਾਇਰਜ਼ (Identifers) ਕੀ ਹੁੰਦੇ ਹਨ ? ਇਹਨਾਂ ਦੇ ਵੱਖ-ਵੱਖ ਨੇਮਿੰਗ (Naming) ਰੂਲਜ਼ ਲਿਖੋ।
3. ਲਿਟਰਲਜ਼ ਕੀ ਹੁੰਦੇ ਹਨ ? ਪਾਈਥਨ ਵਿਚ ਵਰਤੇ ਜਾਂਦੇ ਵੱਖ-ਵੱਖ ਤਰ੍ਹਾਂ ਦੇ ਲਿਟਰਲਜ਼ ਨੂੰ ਦਰਸਾਉਂਦਾ ਇਕ ਚਾਰਟ ਤਿਆਰ ਕਰੋ।
4. ਕਮੈਂਟਸ (Comments) ਕੀ ਹੁੰਦੇ ਹਨ ? ਪਾਈਥਨ ਵਿਚ ਕਮੈਂਟਸ ਲਿਖਣ ਦੇ ਵੱਖ-ਵੱਖ ਤਰੀਕੇ ਲਿਖੋ।

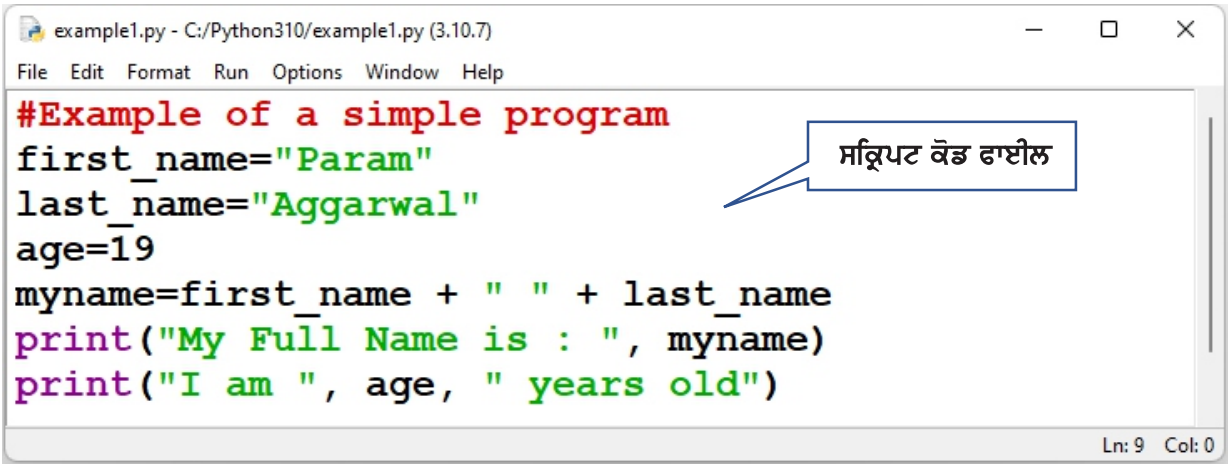
ਲੈਬ ਐਕਟੀਵਿਟੀ

2.1 IDLE ਖੋਲੋ ਅਤੇ ਹੇਠਾਂ ਦਰਸਾਏ ਗਏ ਨਿਰਦੇਸ਼ਾਂ ਨੂੰ ਲਿਖ ਕੇ ਇੰਟਰਐਕਟਿਵ ਮੋਡ ਵਿੱਚ ਅਭਿਆਸ ਕਰੋ:



''ਪਾਇਥਨ ਨਾਲ ਜਾਣ-ਪਛਾਣ'' ਪਾਠ ਵਿਚ ਕੀਤੀ ਗਈ ਚਰਚਾ ਅਨੁਸਾਰ ਇਹਨਾਂ ਕੋਡਜ਼ ਦੀ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਅਤੇ ਕੰਮਾਂ ਨੂੰ ਸਮਝਣ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰੋ।

2.2 IDLE ਦੀ ਵਰਤੋਂ ਨਾਲ ਨਵੀਂ ਫਾਈਲ ਤਿਆਰ ਕਰਕੇ ਸਕ੍ਰਿਪ ਮੋਡ ਵਿੱਚ ਪਾਈਥਨ ਕੋਡ ਲਿਖੋ ਅਤੇ ਚਲਾਓ

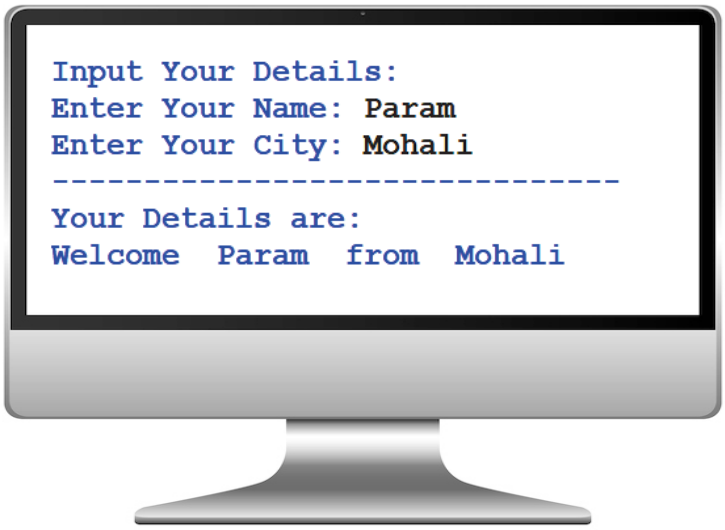


ਹੁਣ ਇਸ ਫਾਈਲ ਨੂੰ ਚਲਾਉਣ ਲਈ ਕੀਬੋਰਡ ਤੋਂ F5 ਕੀਅ ਪ੍ਰੈਸ ਕਰੋ, ਇਹ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ ਆਉਟਪੁਟ ਦਿਖਾਏਗਾ:

```
My Full Name is :  Param Aggarwal
I am  19  years old
```

2.3 ਇੰਟਰਐਕਟਿਵ/ਸਕ੍ਰਿਪਟ ਮੋਡਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਪਾਈਥਨ IDLE ਵਿੱਚ ਹੇਠ ਲਿਖੀਆਂ ਹਦਾਇਤਾਂ ਨੂੰ ਲਾਗੂ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰੋ:

1. IDLE (ਇੰਟਰਐਕਟਿਵ ਮੋਡ) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੇਠਾਂ ਦਿਤੀਆਂ ਕੈਲਕੁਲੇਸ਼ਨਾਂ ਦਾ ਨਤੀਜਾ ਪਤਾ ਕਰੋ।
   ੳ) 7+4
   ਅ) 6+9*10
   ੲ) (6+9)*10
   ਸ) print(7–2+3–2)
   ਹ) print(7/2)
2. IDLE ਵਿਚ ਪਾਈਥਨ ਕੀਵਰਡਜ਼ ਦੀ ਲਿਸਟ ਦੇਖਣ ਲਈ ਸਟੇਟਮੈਂਟ ਲਿਖੋ।
3. ਪਾਈਥਨ ਵਿਚ ਸਕ੍ਰਿਪਟ ਮੋਡ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਤਿਆਰ ਕਰੋ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਆਉਟਪੁੱਟ ਨੂੰ ਨੋਟ ਕਰੋ:
   ੳ)
   ```
   msg = “Hellow Friends”
   print(msg)
   ```
   ਅ)
   ```
   msg1 = “Python” + “Programming”
   msg2 = “Hello from” + msg1
   print(msg2)
   ```
   ੲ)
   ```
   str1 = input(“Enter name of your friend:”)
   str2 = “Freind:”
   print(“Hellow My”, str2, str1)
   ```
4. ਇੱਕ ਅਜਿਹਾ ਪਾਈਥਨ ਕੋਡ ਲਿਖੋ ਜੋ ਤੁਹਾਡਾ ਪੂਰਾ ਨਾਮ ਅਤੇ ਤੁਹਾਡੇ ਜਨਮਦਿਨ ਨੂੰ ਵੱਖਰੀਆਂ ਸਟ੍ਰਿੰਗਜ਼ ਵਜੋਂ ਸਕ੍ਰਿਨ ਉਪਰ ਪ੍ਰਿੰਟ ਕਰੋ।
5. ਇੱਕ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜੋ ਦੋ ਲੋਕਾਂ ਦਾ ਨਾਮ ਪੁੱਛਦਾ ਹੋਵੇ ਅਤੇ ਉਹਨਾਂ ਨੂੰ name1 ਅਤੇ name2 ਨਾਮਕ ਵੇਰੀਏਬਲ ਵਿੱਚ ਸਟੋਰ ਕਰਦਾ ਹੋਵੇ, ਫਿਰ ਉਹਨਾਂ ਦੋਵਾਂ ਨਾਵਾਂ ਨੂੰ Hello ਕਹਿੰਦਾ ਮੈਸੇਜ਼ ਸਕ੍ਰਿਨ ਉਪਰ ਦਰਸਾਉਂਦਾ ਹੋਵੇ।
6. ਹੇਠਾਂ ਦਿਖਾਈ ਗਈ ਆਉਟਪੁੱਟ ਨੂੰ ਦਰਸਾਉਂਦਾ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰੋ। ਜ਼ਰੂਰਤ ਅਨੁਸਾਰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਢੁਕਵੇਂ ਕਮੈਂਟਸ ਵੀ ਦਾਖਲ ਕਰੋ।



ਪਾਈਥਨ ਵਿਚ ਡਾਟਾ ਟਾਈਪਸ, ਆਪਰੇਟਰਜ਼ ਅਤੇ ਐਕਸਪ੍ਰੈਸ਼ਨਜ਼

ਡਾਟਾ ਟਾਈਪ ਇੱਕ ਕਿਸਮ ਦੇ ਮੁੱਲ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ ਅਤੇ ਇਹ ਨਿਰਧਾਰਤ ਕਰਦੀ ਹੈ ਕਿ ਉਸ ਉਪਰ ਕਿਹੜੇ ਓਪਰੇਸ਼ਨ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਨੁਮੈਰਿੱਕ, ਨਾਨ-ਨੁਮੈਰਿੱਕ ਅਤੇ ਬੁਲੀਅਨ (ਸਹੀ/ਗਲਤ) ਕਿਸਮ ਦੀਆਂ ਡਾਟਾ ਟਾਈਪਸ ਸਭ ਤੋਂ ਵੱਧ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਡਾਟਾ ਟਾਈਪਸ ਹਨ। ਹਾਲਾਂਕਿ, ਹਰੇਕ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਵਿਚ ਡਾਟਾ ਟਾਇਸ ਦਾ ਆਪਣਾ ਵਰਗੀਕਰਨ ਹੁੰਦਾ ਹੈ ਜੋ ਉਸ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੀ ਫਿਲਾਸਫੀ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।

ਪਾਠ ਦਾ ਉਦੇਸ਼

- ✓ ਡਾਟਾ ਟਾਈਪਸ : ਪਾਈਥਨ ਵਿਚ ਸਟੈਂਡਰਡ ਡਾਟਾ ਟਾਈਪਸ-ਨੁਮੇਰਿੱਕ, ਬੂਲੀਅਨ, ਸਿਕੁਐਂਸ, ਸੈੱਟ, ਮੈਪਿੰਗ, ਨਨ, ਮਿਊਟੇਬਲ ਅਤੇ ਇਮਮਿਊਟੇਬਲ ਟਾਈਪਸ
- ✓ ਪਾਈਥਨ ਵਿਚ ਆਪਰੇਟਰਜ਼ : ਅਰਿੱਥਮੈਟਿਕ, ਰਿਲੇਸ਼ਨਲ, ਲਾਜੀਕਲ, ਅਸਾਈਨਮੈਂਟ, ਮੈਂਬਰਸ਼ਿੱਪ ਆਪਰੇਟਰਜ਼, ਐਕਸਪ੍ਰੈਸ਼ਨਜ਼ ਅਤੇ ਇਹਨਾਂ ਦਾ ਮੁਲਾਂਕਣ: ਆਪਰੇਟਰ ਪ੍ਰੈਸੀਡੇਂਸ; ਟਾਈਪ ਕਨਵਰਜ਼ਨ: ਇੰਪਲੀਸਿੱਟ ਅਤੇ ਐਕਸਪਲੀਸਿੱਟ ਕਨਵਰਜ਼ਨ

ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ✓ ਵਿਦਿਆਰਥੀ ਪਾਈਥਨ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਡਾਟਾ ਟਾਈਪਸ ਦੀ ਧਾਰਨਾ ਅਤੇ ਵਰਤੋਂ ਨੂੰ ਸਮਝਣਗੇ।
- ✓ ਉਹ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਆਪਰੇਟਰਾਂ ਦੀ ਮਹੱਤਤਾ ਅਤੇ ਵਰਤੋਂ ਨੂੰ ਸਮਝਣ ਯੋਗ ਹੋਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀ ਓਪਰੇਟਰਾਂ ਦੀ ਤਰਜੀਹ (ਪ੍ਰੈਸੀਡੇਂਸ) ਦੇ ਆਧਾਰ 'ਤੇ ਐਕਸਪ੍ਰੈਸ਼ਨਜ਼ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਯੋਗ ਹੋਣਗੇ।
- ✓ ਉਹ ਸਧਾਰਨ ਗਣਨਾਵਾਂ ਨੂੰ ਕਰਨ ਵਾਲੀਆਂ ਛੋਟੀਆਂ ਸਕ੍ਰਿਪਟਾਂ ਨੂੰ ਡਿਵਲਪ ਕਰਨ ਯੋਗ ਹੋਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਆਮ ਗਲਤੀਆਂ ਦੀ ਪਛਾਣ ਕਰਨ ਯੋਗ ਹੋਣਗੇ।

03 ਪਾਠ

ਪਾਈਥਨ ਵਿੱਚ ਡਾਟਾ ਟਾਈਪਸ, ਆਪਰੇਟਰਜ਼ ਅਤੇ ਐਕਸਪ੍ਰੈਸ਼ਨਜ਼

ਹੁਣ ਤੱਕ ਅਸੀਂ ਇਸ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰ ਚੁੱਕੇ ਹਾਂ ਕਿ ਪਾਇਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਨਾਲ ਕਿਵੇਂ ਕੰਮ ਕਰਨਾ ਹੈ ਅਤੇ ਪਾਈਥਨ ਕੋਡ ਨੂੰ ਕਿਵੇਂ ਚਲਾਉਣਾ ਹੈ। ਹੁਣ ਅਸੀਂ ਪਾਈਥਨ ਭਾਸ਼ਾ ਦੇ ਹੋਰ ਵੇਰਵਿਆਂ ਬਾਰੇ ਡਿਟੇਲ ਵਿਚ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਾਂਗੇ। ਇਸ ਪਾਠ ਵਿੱਚ ਅਸੀਂ ਪਾਇਥਨ ਵਿੱਚ ਮੌਜੂਦ ਮੁੱਢਲੀਆਂ ਡਾਟਾ ਟਾਈਪਸ, ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਆਪਰੇਟਰਾਂ, ਐਕਸਪ੍ਰੈਸ਼ਨਜ਼ ਅਤੇ ਵੱਖ-ਵੱਖ ਤਰ੍ਹਾਂ ਦੀਆਂ ਡਾਟਾ ਕਨਵਰਜ਼ਨ ਤਕਨੀਕਾਂ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਾਂਗੇ। ਪਾਈਥਨ ਡਾਟਾ ਟਾਈਪਸ ਅਤੇ ਆਪਰੇਟਰਾਂ ਨੂੰ ਬਿਹਤਰ ਤਰੀਕੇ ਨਾਲ ਸਮਝਣ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਇਸ ਭਾਸ਼ਾ ਦੇ ਡਿਜ਼ਾਈਨ ਅਤੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਹੋਰ ਪ੍ਰਭਾਵੀ ਢੰਗ ਨਾਲ ਸਮਝਣਯੋਗ ਬਣ ਸਕਾਂਗੇ।

3.0 ਡਾਟਾ ਟਾਈਪਸ (DATA TYPES)

ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਿੱਚ ਡਾਟਾ ਟਾਈਪ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਸੰਕਲਪ (Concept) ਹੈ। ਵੇਰੀਏਬਲਾਂ ਵਿਚ ਮੁੱਲ, ਸਟੋਰ ਕੀਤੇ ਜਾਂਦੇ ਹਨ, ਅਤੇ ਹਰੇਕ ਮੁੱਲ ਦਾ ਇੱਕ ਡਾਟਾ-ਟਾਈਪ ਹੁੰਦਾ ਹੈ। ਡਾਟਾ-ਟਾਈਪ ਡਾਟਾ ਦੀ ਕਿਸਮ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਜੋ ਇੱਕ ਵੇਰੀਏਬਲ ਦੇ ਅੰਦਰ ਸਟੋਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਇੰਟੀਜ਼ਰ ਅੰਕ (+ve ਜਾਂ -ve ਚਿੰਨ੍ਹ ਦੇ ਨਾਲ ਜਾਂ ਇਹਨਾਂ ਤੋਂ ਬਿਨਾਂ ਸੰਪੂਰਨ ਸੰਖਿਆਵਾਂ (Whole Numbers)), ਸਟ੍ਰਿੰਗ (ਕੁਟੇਸ਼ਨ ਚਿੰਨ੍ਹਾਂ ਵਿੱਚ ਬੰਦ ਅੱਖਰਾਂ ਦਾ ਕ੍ਰਮ) ਅਤੇ ਰੀਅਲ (Real Values) ਮੁੱਲ (ਦਸ਼ਮਲਵ ਬਿੰਦੂ (Decimal Point) ਦੇ ਨਾਲ ਨੁਮੇਰਿੱਕ ਮੁੱਲ) ਆਮ ਵਰਤੇ ਜਾਂਦੇ ਮੁੱਲ ਹਨ ਜੋ ਅਸੀਂ ਵੇਰੀਏਬਲਾਂ ਵਿੱਚ ਸਟੋਰ ਕਰਨ ਲਈ ਵਰਤਦੇ ਹਾਂ।

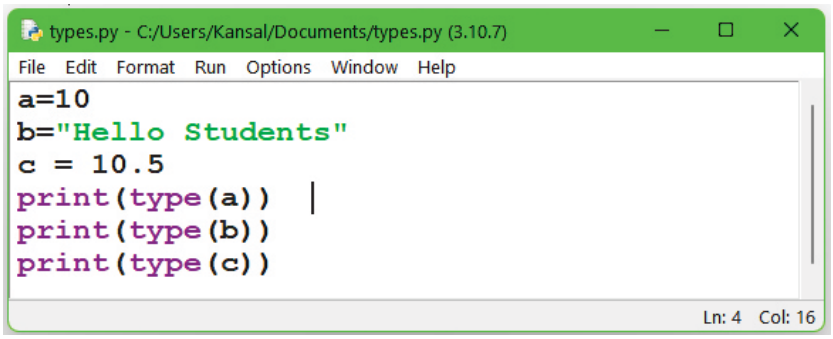
ਪਾਈਥਨ ਇੱਕ ਡਾਇਨਾਮਿੱਕਲੀ ਟਾਈਪਡ (Dynamically Typed) ਭਾਸ਼ਾ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਸਾਨੂੰ ਵੇਰੀਏਬਲ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਦੇ ਸਮੇਂ ਉਸ ਦੀ ਡਾਟਾ-ਟਾਈਪ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਪੈਂਦੀ। ਇੰਟਰਪ੍ਰੈਟਰ ਆਪਣੇ ਆਪ ਮੁੱਲ ਦੀ ਟਾਈਪ ਅਨੁਸਾਰ ਵੇਰੀਏਬਲ ਨੂੰ ਉਸਦੀ ਟਾਈਪ ਨਾਲ ਜੋੜ ਦਿੰਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ:

z = 10.5

ਇੱਸ ਉਦਾਹਰਣ ਵਿੱਚ ਵੇਰੀਏਬਲ z ਵਿੱਚ ਫਲੋਟ ਮੁੱਲ 10.5 ਸਟੋਰ ਕੀਤਾ ਜਾ ਰਿਹਾ ਹੈ ਅਤੇ ਅਸੀਂ ਇਸਦੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਨਹੀਂ ਕੀਤਾ ਗਿਆ। ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਆਪਣੇ ਆਪ ਵੇਰੀਏਬਲ z ਨੂੰ ਫਲੋਟਿੰਗ ਟਾਈਪ ਦੇ ਤੌਰ 'ਤੇ ਇੰਟਰਪ੍ਰੈਟ (Interpret) ਕਰੇਗਾ।

ਅਸੀਂ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਵਰਤੇ ਗਏ ਵੇਰੀਏਬਲ ਦੇ ਡਾਟਾ-ਟਾਈਪ ਦੀ ਜਾਂਚ (Check) ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ। ਪਾਈਥਨ ਸਾਨੂੰ ਇਸ ਮਕਸਦ ਲਈ **type()** ਫੰਕਸ਼ਨ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। **type()** ਫੰਕਸ਼ਨ ਇੱਕ ਆਰਗੂਮੈਂਟ ਦੇ ਰੂਪ ਵਿੱਚ ਇਸ ਵਿੱਚ ਪਾਸ ਕੀਤੇ ਗਏ ਵੇਰੀਏਬਲ ਦੇ ਡਾਟਾ-ਟਾਈਪ ਨੂੰ ਵਾਪਸ (return) ਕਰਦਾ ਹੈ।

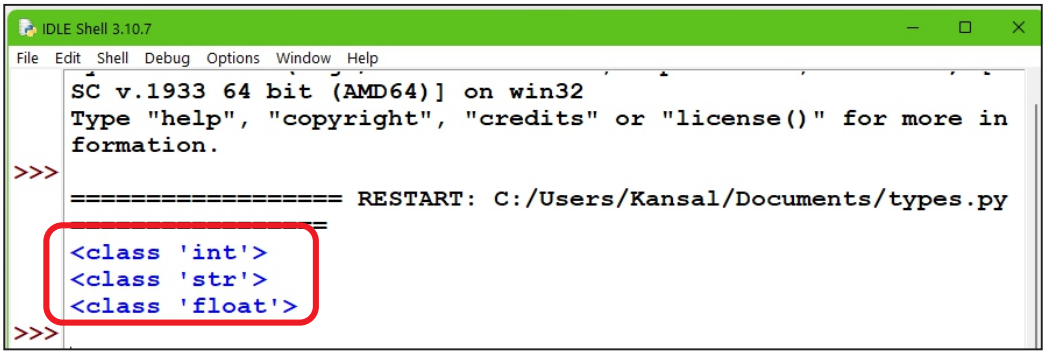
ਇਸ ਫੰਕਸ਼ਨ ਨਾਲ ਸੰਬੰਧਤ ਅੱਗੇ ਉਦਾਹਰਣ ਪ੍ਰੋਗਰਾਮ ਦਿੱਤਾ ਗਿਆ ਹੈ ਜਿਸ ਵਿੱਚ ਅਸੀਂ ਵੱਖ-ਵੱਖ ਡਾਟਾ-ਟਾਈਪਸ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰ ਰਹੇ ਹਾਂ ਅਤੇ ਫਿਰ type() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਨਾਲ ਉਹਨਾਂ ਦੇ ਡਾਟਾ-ਟਾਈਪਸ ਦੀ ਜਾਂਚ ਕੀਤੀ ਗਈ ਹੈ:



ਚਿੱਤਰ 3.1: type() ਫੰਕਸ਼ਨ ਨਾਲ ਵੇਰੀਏਬਲਾਂ ਦੇ ਡਾਟਾ-ਟਾਈਪ ਦੀ ਜਾਂਚ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ (types.py)

ਡਾਟਾ-ਟਾਈਪ ਵੇਰੀਏਬਲਜ਼ ਅੰਦਰ ਸਟੋਰ ਕੀਤੇ ਜਾ ਸਕਦੇ ਡਾਟਾ ਦੀ ਕਿਸਮ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਪਾਈਥਨ ਇੱਕ ਡਾਇਨਾਮਿੱਕਲੀ ਟਾਈਪਡ **(Dynamically Typed)** ਭਾਸ਼ਾ ਹੈ, ਜਿਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਸਾਨੂੰ ਵੇਰੀਏਬਲਜ਼ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਦੇ ਸਮੇਂ ਉਹਨਾਂ ਦੀ ਡਾਟਾ-ਟਾਈਪ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਪੈਂਦੀ। ਇੰਟਰਪ੍ਰੈਟਰ ਆਪਣੇ ਆਪ ਮੁੱਲ ਦੀ ਟਾਈਪ ਅਨੁਸਾਰ ਵੇਰੀਏਬਲ ਨੂੰ ਉਸਦੀ ਟਾਇਪ ਨਾਲ ਜੋੜ ਦਿੰਦਾ ਹੈ।

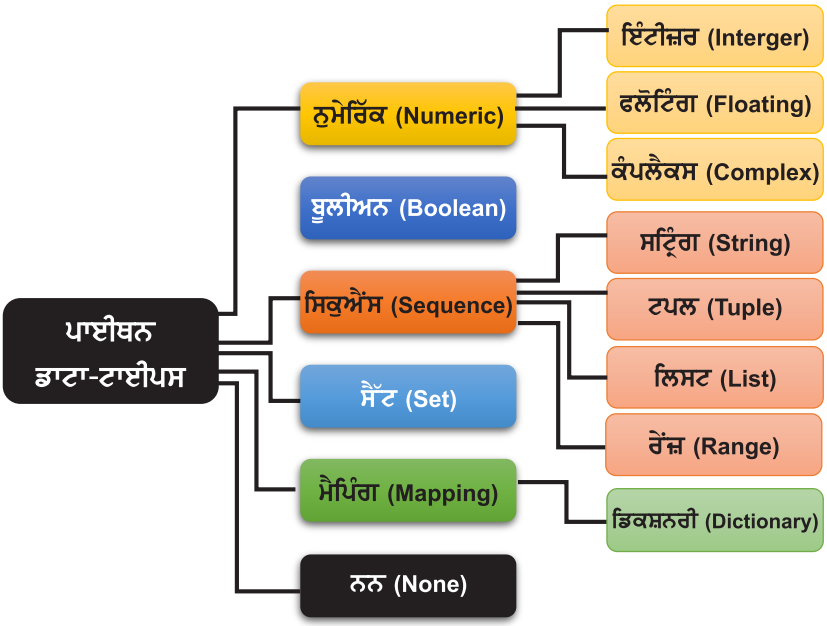
ਪ੍ਰੋਗਰਾਮ (types.py) ਨੂੰ ਚਲਾਉਣ ਤੋਂ ਬਾਅਦ, ਸਾਨੂੰ ਹੇਠਾਂ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਮਿਲੇਗੀ:



ਚਿੱਤਰ 3.2: ਪ੍ਰੋਗਰਾਮ **types.py** ਦਾ ਆਉਟਪੁੱਟ

3.1.1 ਪਾਈਥਨ ਵਿੱਚ ਸਟੈਂਡਰਡ ਡਾਟਾ ਟਾਈਪਸ (Standard Data Types in Python):

ਸਟੈਂਡਰਡ ਡਾਟਾ ਟਾਈਪਸ ਨੂੰ ਬਿਲਟ-ਇਨ ਡਾਟਾ ਟਾਈਪਸ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਡਾਟਾ ਟਾਈਪਸ ਨੂੰ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਵਿੱਚ ਬਣਾਇਆ ਗਿਆ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਕਈ ਸਟੈਂਡਰਡ ਡਾਟਾ ਟਾਈਪਸ ਹਨ ਅਤੇ ਹਰੇਕ ਟਾਈਪ ਲਈ ਓਪਰੇਸ਼ਨਾਂ ਅਤੇ ਸਟੋਰੇਜ ਵਿਧੀਆਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਗਿਆ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਪਰਿਭਾਸ਼ਿਤ ਕੁੱਝ ਆਮ ਡਾਟਾ ਟਾਈਪਸ ਨੂੰ ਹੇਠਾਂ ਦਿੱਤੇ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:



ਚਿੱਤਰ 3.3 : ਪਾਈਥਨ ਵਿਚ ਆਮ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਸਟੈਂਡਰਡ ਡਾਟਾ ਟਾਈਪਸ

ਸਟੈਂਡਰਡ ਡਾਟਾ ਟਾਈਪਸ ਨੂੰ ਬਿਲਟ-ਇਨ ਡਾਟਾ ਟਾਈਪਸ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਡਾਟਾ ਟਾਈਪਸ ਨੂੰ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਵਿੱਚ ਬਣਾਇਆ ਗਿਆ ਹੈ।

3.1.1.1 ਪਾਈਥਨ ਵਿਚ ਨੁਮੇਰਿੱਕ ਡਾਟਾ ਟਾਈਪਸ (Numeric Data Types in Python):

ਨੁਮੇਰਿੱਕ ਡਾਟਾ ਟਾਈਪਸ ਉਹ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਜੋ ਸੰਖਿਆਤਮਕ ਮੁੱਲਾਂ (Numeric Values) ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਨੁਮੇਰਿੱਕ ਟਾਈਪਸ ਵਿੱਚ ਇੰਟੀਜ਼ਰਸ (ਪੂਰਨ ਅੰਕ), ਫਲੋਟਿੰਗ ਟਾਈਪ ਨੰਬਰਸ (ਜਾਂ ਫਲੋਟਸ), ਅਤੇ ਕੰਪਲੈਕਸ ਨੰਬਰ ਆਉਂਦੇ ਹਨ:

- **ਇੰਟੀਜ਼ਰ (Integers)** ਸੰਪੂਰਨ ਸੰਖਿਆਵਾਂ (Whole Numbers) (ਭਿੰਨਾਂ (fractions) ਜਾਂ ਦਸ਼ਮਲਵ ਤੋਂ ਬਿਨਾਂ) ਹੁੰਦੇ ਹਨ ਜੋ ਨਕਾਰਾਤਮਕ (Negative), ਜ਼ੀਰੋ ਜਾਂ ਸਕਾਰਾਤਮਕ (Positive) ਹੋ ਸਕਦੇ ਹਨ। ਉਦਾਹਰਨ ਲਈ: 44, -56, 0, +77 ਆਦਿ। ਪਾਈਥਨ ਵਿੱਚ, ਇਸ ਗੱਲ ਦੀ ਕੋਈ ਸੀਮਾ ਨਹੀਂ (no limit) ਹੈ ਕਿ ਇੱਕ ਪੂਰਨ ਅੰਕ ਦਾ ਮੁੱਲ ਕਿੰਨਾ ਵੱਡਾ ਹੋ ਸਕਦਾ ਹੈ। int() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਫਲੋਟਿੰਗ ਨੰਬਰ ਜਾਂ ਸਟ੍ਰਿੰਗ ਨੂੰ ਇਸਦੇ ਬਰਾਬਰ ਦੇ ਇੰਟੀਜ਼ਰ ਅੰਕ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵੀ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਉਦਾਹਰਨ ਲਈ: int(6.7), ਇਹ ਫਲੋਟਿੰਗ ਨੰਬਰ 6.7 ਨੂੰ ਇੰਟੀਜ਼ਰ ਨੰਬਰ 6 ਵਿੱਚ ਬਦਲ ਦੇਵੇਗਾ।
- **ਫਲੋਟਿੰਗ ਨੰਬਰਸ (Floating Numbers)** ਰੀਅਲ ਨੰਬਰ (Real Numbers) ਹੁੰਦੇ ਹਨ ਜੋ ਦਸ਼ਮਲਵ ਬਿੰਦੂ (Decimal Point) ਨਾਲ ਲਿਖੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਦਸ਼ਮਲਵ ਬਿੰਦੂ ਰੀਅਲ ਨੰਬਰ ਨੂੰ ਦੋ ਭਾਗਾਂ ਵਿੱਚ ਵੰਡਦਾ ਹੈ: ਇੰਟੀਜ਼ਰ (Integer) ਅਤੇ ਫਰੈਕਸ਼ਨਲ (Fractional) ਭਾਗ। ਉਦਾਹਰਨ ਲਈ: 56.5, +69.63, 0.56, 4.977 ਆਦਿ। ਫਲੋਟਿੰਗ-ਪੁਆਇੰਟ ਵੇਰੀਏਬਲ ਵਿੱਚ ਵੱਧ ਤੋਂ ਵੱਧ ਮੁੱਲ ਲਗਭਗ $1.8x10^{308}$ ਹੈ ਸਕਦਾ ਹੈ। ਜੇਕਰ ਇਨਪੁੱਟ ਕੀਤਾ ਮੁੱਲ ਇਸ ਅਧਿਕਤਮ ਮੁੱਲ ($1.8x10^{308}$) ਤੋਂ ਵੱਧ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਇੱਕ ਐਰਰ (inf (infinity)) ਦਰਸਾਵੇਗਾ। ਫਲੋਟਿੰਗ ਪੁਆਇੰਟ ਨੰਬਰਾਂ ਨੂੰ ਵਿਗਿਆਨਕ (Scientific) ਨੋਟੇਸ਼ਨ ਵਿੱਚ ਦਰਸਾਉਣ ਸਮੇਂ ਐਕਸਪੋਨੈਂਟ (Exponent) ਭਾਗ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਅਸੀਂ ਅੱਖਰ e ਜਾਂ E ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਉਦਾਹਰਨ ਲਈ: 3.23e-18 (ਜੋ ਕਿ 3.23×10^{-18} ਹੈ), 3.25e+205 (ਜੋ ਕਿ $3.25x10^{+205}$ ਹੈ)। ਅਸੀਂ ਦਿੱਤੇ ਮੁੱਲ ਨੂੰ ਫਲੋਟਿੰਗ ਪੁਆਇੰਟ ਮੁੱਲ ਵਿੱਚ ਬਦਲਣ ਲਈ float() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ। ਉਦਾਹਰਨ ਲਈ: float(45), ਇਹ 45 ਨੂੰ ਫਲੋਟਿੰਗ ਨੰਬਰ 45.0 ਵਿੱਚ ਬਦਲ ਦੇਵੇਗਾ।
- **ਕੰਪਲੈਕਸ (Complex)** ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਦੋ ਮੁੱਲ ਹੁੰਦੇ ਹਨ: ਪਹਿਲਾ ਕੰਪਲੈਕਸ ਸੰਖਿਆ ਦਾ ਰੀਅਲ ਭਾਗ (Real Part) ਹੁੰਦਾ ਹੈ, ਅਤੇ ਦੂਜਾ ਇਮੇਜ਼ਨਰੀ ਭਾਗ (Imaginary Part) ਹੁੰਦਾ ਹੈ। ਪਾਈਥਨ ਇੱਕ ਕੰਪਲੈਕਸ ਨੰਬਰ ਨੂੰ ਦਰਸਾਉਣ ਲਈ A+Bj ਨੋਟੇਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਕੰਪਲੈਕਸ ਨੰਬਰ ਦੇ ਦੋਵੇਂ ਭਾਗਾਂ ਨੂੰ ਰੀਅਲ ਨੰਬਰਾਂ ਵਜੋਂ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। -2+3j ਕੰਪਲੈਕਸ ਨੰਬਰ ਦੀ ਉਦਾਹਰਨ ਹੈ। ਪਾਈਥਨ Complex() ਫੰਕਸ਼ਨ ਦਿੱਤੇ ਗਏ ਪੈਰਾਮੀਟਰਾਂ ਨੂੰ ਕੰਪਲੈਕਸ ਨੰਬਰਾਂ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: complex(2,5), ਇਹ ਰੀਅਲ ਨੰਬਰਜ਼ 2 ਅਤੇ 5 ਨੂੰ ਕੰਪਲੈਕਸ ਨੰਬਰ 2+5j ਵਿੱਚ ਬਦਲ ਦੇਵਗਾ।

ਨੁਮੇਰਿੱਕ ਡਾਟਾ ਟਾਈਪਸ ਉਹ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਜੋ ਸੰਖਿਆਤਮਕ ਮੁੱਲਾਂ (Numeric Values) ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਨੁਮੇਰਿੱਕ ਟਾਈਪਸ ਵਿੱਚ ਇੰਟੀਜ਼ਰਸ (ਪੂਰਨ ਅੰਕ), ਫਲੋਟਿੰਗ ਟਾਈਪ ਨੰਬਰਸ (ਜਾਂ ਫਲੋਟਸ), ਅਤੇ ਕੰਪਲੈਕਸ ਨੰਬਰ ਆਉਂਦੇ ਹਨ।

3.1.1.2 ਪਾਈਥਨ ਵਿਚ ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ (Boolean Data Type in Python):

ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ ਦੀ ਵਰਤੋਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਦੀ ਟਰੁੱਥ ਵੈਲੀਊ (Truth Value) ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਉਦਾਹਰਨ ਲਈ, ਐਕਸਪ੍ਰੈਸ਼ਨ 5 < = 10 ਸਹੀ (True) ਹੈ, ਜਦੋਂ ਕਿ ਐਕਸਪ੍ਰੈਸ਼ਨ 5 > 10 ਗਲਤ (False) ਹੈ। ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ ਦੇ ਸਿਰਫ਼ ਦੋ ਸੰਭਵ ਮੁੱਲ ਹੁੰਦੇ ਹਨ: ਸਹੀ (True) ਅਤੇ ਗਲਤ (False)। ਬੂਲੀਅਨ ਟਾਈਪ ਲਈ ਕੋਈ ਹੋਰ ਮੁੱਲ ਸੰਭਵ ਨਹੀਂ ਹੈ। ਪਾਈਥਨ ਦੇ bool() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦਿਤੇ ਗਏ ਮੁੱਲ ਨੂੰ ਬੂਲੀਅਨ ਮੁੱਲ ਵਿੱਚ ਬਦਲਣ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਕੋਈ ਵੀ ਇੰਟੀਜ਼ਰ, ਫਲੋਟਿੰਗ-ਪੁਆਇੰਟ ਨੰਬਰ, ਜਾਂ ਕੰਪਲੈਕਸ ਨੰਬਰ ਜਿਸਦਾ ਮੁੱਲ ਜ਼ੀਰੋ ਹੋਵੇ, ਨੂੰ ਗਲਤ (False) ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ, ਜਦੋਂ ਕਿ ਕਿਸੇ ਵੀ ਸਕਾਰਾਤਮਕ (Positive) ਜਾਂ ਨਕਾਰਾਤਮਕ (Negative) ਨੰਬਰ ਨੂੰ ਸਹੀ (True) ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: bool(4.5) ਸਹੀ (True) ਮੁੱਲ ਵਾਪਸ ਕਰੇਗਾ, ਜਦੋਂ ਕਿ bool(0+0j) ਗਲਤ (False) ਮੁੱਲ ਵਾਪਸ ਕਰੇਗਾ।

ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ ਦੀ ਵਰਤੋਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਦੀ ਟਰੁੱਥ ਵੈਲੀਊ (**Truth Value**) ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ ਦੇ ਸਿਰਫ਼ ਦੋ ਸੰਭਵ ਮੁੱਲ ਹੁੰਦੇ ਹਨ: ਸਹੀ (**True**) ਅਤੇ ਗਲਤ (**False**)

TEST YOURSELF 3.1

ਹੇਠਾਂ ਦਿੱਤੇ ਗਏ ਮੁੱਲਾਂ ਦੀਆਂ ਕਿਸਮਾਂ ਲਿਖੋ:

ੳ) 9174 _____ ਸ) -4+5j _____

ਅ) True _____ ਹ) 3.23e-18 _____

ੲ) 3.14 _____ ਕ) False _____

3.1.1.3 ਪਾਈਥਨ ਵਿੱਚ ਸਿਕੁਐਂਸ ਡਾਟਾ ਟਾਈਪ (Sequence Data Types in Python):

ਸਿਕੁਐਂਸ (Sequence) ਇਕੋ (Similar) ਟਾਈਪ ਜਾਂ ਵੱਖ-ਵੱਖ ਟਾਈਪ ਦੋ ਮੁੱਲਾਂ ਦਾ ਇੱਕ ਕ੍ਰਮਬੱਧ ਸੰਗ੍ਰਹਿ (Ordered Collection) ਹੁੰਦਾ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਕਈ ਸਿਕੁਐਂਸ ਡਾਟਾ ਟਾਈਪਸ ਹਨ। ਪਾਈਥਨ ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਸਿਕੁਐਂਸ ਟਾਈਪਸ ਦਾ ਵਰਨਣ ਹੇਠਾਂ ਕੀਤਾ ਗਿਆ ਹੈ:

- **ਲਿਸਟ (List) :** ਲਿਸਟ ਇੱਕ ਜਾਂ ਇੱਕ ਤੋਂ ਵੱਧ ਡਾਟਾ ਆਈਟਮਾਂ (ਤੱਤਾਂ) ਦਾ ਕ੍ਰਮਬੱਧ ਸੰਗ੍ਰਹਿ (Ordered Collection) ਹੁੰਦੀ ਹੈ, ਇਹ ਜ਼ਰੂਰੀ ਨਹੀਂ ਕਿ ਲਿਸਟ ਦੀਆਂ ਸਾਰੀਆਂ ਆਈਟਮਾਂ ਇੱਕੋ ਟਾਈਪ ਦੀਆਂ ਹੋਣ। ਇੱਕ ਲਿਸਟ ਬਣਾਉਣ ਲਈ ਅਸੀਂ ਡਾਟਾ ਆਈਟਮਾਂ ਨੂੰ ਸਕੇਅਰ ਬਰੈਕਟਾਂ ([ ]) ਵਿੱਚ ਲਿਖਦੇ ਹਾਂ। ਸਾਰੀਆਂ ਆਈਟਮਾਂ ਨੂੰ ਕਾਮਿਆਂ (Commas) ਨਾਲ ਵੱਖ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: [12, 5.6, 29, "Hello"] । ਪਾਈਥਨ ਲਿਸਟਾਂ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਤੱਤ (Elements) ਸ਼ਾਮਲ ਹੋ ਸਕਦੇ ਹਨ। ਲਿਸਟਾਂ ਸੋਧਣਯੋਗ (Modifiable) ਹੁੰਦੀਆਂ ਹਨ; ਭਾਵ ਲਿਸਟ ਵਿਚ ਤੱਤਾਂ ਨੂੰ ਅਸਾਈਨ ਕਰਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਇਸਦੇ ਤੱਤਾਂ ਵਿਚ ਬਦਲਾਵ ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਟਪਲ (Tuple) :** ਟਪਲ ਇੱਕ ਜਾਂ ਇੱਕ ਤੋਂ ਵੱਧ ਡਾਟਾ ਆਈਟਮਾਂ (ਤੱਤਾਂ) ਦਾ ਕ੍ਰਮਬੱਧ ਸੰਗ੍ਰਹਿ (Ordered Collection) ਹੁੰਦਾ ਹੈ, ਇਹ ਜ਼ਰੂਰੀ ਨਹੀਂ ਕਿ ਟਪਲ ਦੀਆਂ ਸਾਰੀਆਂ ਆਈਟਮਾਂ ਇੱਕੋ ਟਾਈਪ ਦੀਆਂ ਹੋਣ। ਇੱਕ ਟਪਲ ਬਣਾਉਣ ਲਈ ਅਸੀਂ ਡਾਟਾ ਆਈਟਮਾਂ ਨੂੰ ਗੋਲ ਬਰੈਕਟਾਂ ਵਿੱਚ ਲਿਖਦੇ ਹਾਂ। ਸਾਰੀਆਂ ਆਈਟਮਾਂ ਨੂੰ ਕਾਮਿਆਂ (Commas) ਨਾਲ ਵੱਖ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: (12, 5.6, 29, "Hello")। ਇੱਕ ਟਪਲ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਮੁੱਲ ਵੀ ਹੋ ਸਕਦੇ ਹਨ। ਟਪਲਸ ਰਾਈਟ-ਪ੍ਰੋਟੈਕਟਡ (Write-Protected) ਹੁੰਦੇ ਹਨ, ਭਾਵ ਟਪਲ ਵਿਚ ਤੱਤਾਂ ਨੂੰ ਅਸਾਈਨ ਕਰਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਇਸਦੇ ਤੱਤਾਂ ਵਿਚ ਬਦਲਾਵ ਨਹੀਂ ਕਰ ਸਕਦੇ।
- **ਰੇਂਜ (Range) :** ਰੇਂਜ ਇੱਕ ਅਜਿਹੀ ਡਾਟਾ ਟਾਈਪ ਹੈ ਜੋ ਮੁੱਖ ਤੌਰ 'ਤੇ ਉਸ ਸਮੇਂ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਜਦੋਂ ਅਸੀਂ ਲੂਪ ਦੀ ਵਰਤੋਂ ਕਰ ਰਹੇ ਹੁੰਦੇ ਹਾਂ। ਇਹ ਸੰਖਿਆਵਾਂ ਦੇ ਇੱਕ ਗੈਰ-ਸੋਧਣਯੋਗ ਕ੍ਰਮ (Non-Modifiable Sequence) ਨੂੰ ਜੈਨਰੇਟ ਕਰਦਾ ਹੈ। range() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਇਸਦੇ ਵਿੱਚ ਦਿੱਤੇ ਗਏ ਆਰਗੂਮੈਂਟਾਂ ਅਨੁਸਾਰ ਸੰਖਿਆਵਾਂ ਦਾ ਕ੍ਰਮ ਬਣਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। range() ਫੰਕਸ਼ਨ ਡਿਫਾਲਟ ਰੂਪ ਵਿੱਚ 0 ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦੇ ਹੋਏ ਸੰਖਿਆਵਾਂ ਦਾ ਇੱਕ ਕ੍ਰਮ (Sequence of Numbers) ਵਾਪਸ (return) ਕਰਦਾ ਹੈ; ਇਹ ਕ੍ਰਮ ਮੂਲ ਰੂਪ ਵਿੱਚ 1-1 ਅੰਕ ਦੁਆਰਾ ਵਧਦਾ ਹੈ ਅਤੇ ਇੱਕ ਨਿਰਧਾਰਤ ਸੰਖਿਆ ਤੋਂ ਪਹਿਲਾਂ ਇਹ ਕ੍ਰਮ ਰੁਕ ਜਾਂਦਾ ਹੈ। range() ਫੰਕਸ਼ਨ ਦਾ ਸਿੰਟੈਕਸ ਹੈ:

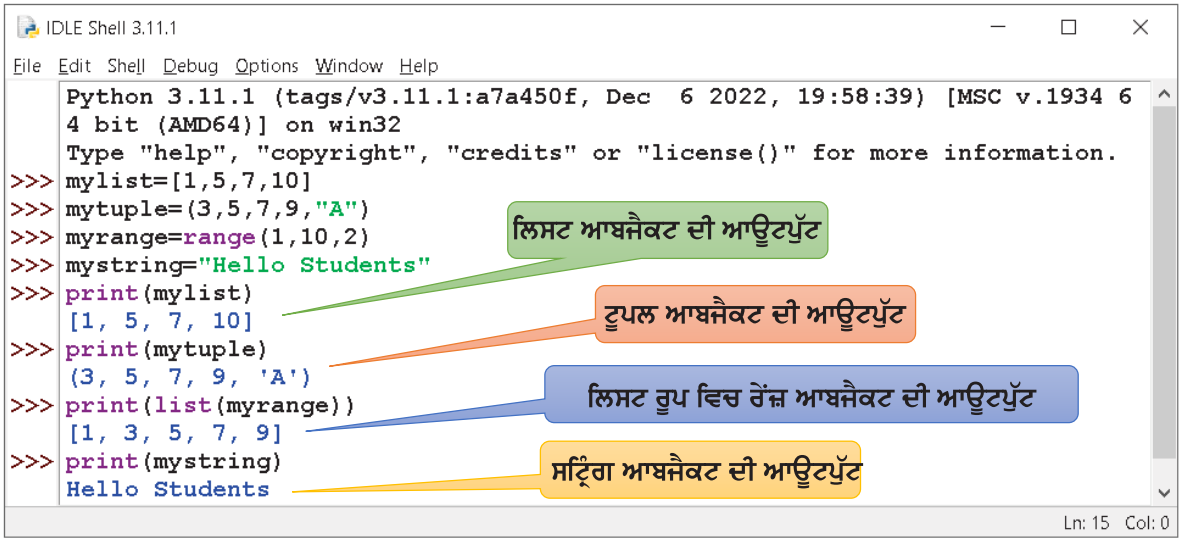
  range(start, stop, step).

  ਇੱਥੇ start ਇੱਕ ਆਪਸ਼ਨਲ ਪੈਰਾਮੀਟਰ ਹੈ ਜੋ ਸੰਖਿਆਵਾਂ ਦੇ ਕ੍ਰਮ ਦੀ ਸ਼ੁਰੂਆਤ ਨੂੰ ਤੈਅ ਕਰਦਾ ਹੈ, stop ਪੈਰਾਮੀਟਰ ਇਹ ਨਿਰਧਾਰਿਤ ਕਰਦਾ ਹੈ ਕਿ ਕਿਸ ਸੰਖਿਆ ਤੱਕ ਪਹੁੰਚਣ ਉਪਰ ਕ੍ਰਮ ਨੂੰ ਰੋਕਣਾ ਹੈ (stop ਮੁੱਲ

ਤਿਆਰ ਕੀਤੇ ਗਏ ਕ੍ਰਮ ਵਿੱਚ ਸ਼ਾਮਲ ਨਹੀਂ ਹੁੰਦਾ), ਅਤੇ step ਵੀ ਇੱਕ ਆਪਸ਼ਨਲ ਪੈਰਾਮੀਟਰ ਹੈ ਜੋ range() ਫੰਕਸ਼ਨ ਦੁਆਰਾ ਤਿਆਰ ਕ੍ਰਮ ਵਿੱਚ ਕਿਸੇ ਵੀ ਦੋ ਲਗਾਤਾਰ ਸੰਖਿਆਵਾਂ ਵਿੱਚ ਅੰਤਰ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ:

- **range(10):** ਇਸ ਦੁਆਰਾ ਅੰਕਾਂ ਦਾ ਇਹ ਕ੍ਰਮ ਜੈਨਰੇਟ ਕੀਤਾ ਜਾਵੇਗਾ: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- **range(5,10):** ਇਸ ਦੁਆਰਾ ਅੰਕਾਂ ਦਾ ਇਹ ਕ੍ਰਮ ਜੈਨਰੇਟ ਕੀਤਾ ਜਾਵੇਗਾ: 5, 6, 7, 8, 9
- **range(2,15,2):** ਇਸ ਦੁਆਰਾ ਅੰਕਾਂ ਦਾ ਇਹ ਕ੍ਰਮ ਜੈਨਰੇਟ ਕੀਤਾ ਜਾਵੇਗਾ: 2, 4, 6, 8, 10, 12, 14
- **range(10,5,–1):** ਇਸ ਦੁਆਰਾ ਅੰਕਾਂ ਦਾ ਇਹ ਕ੍ਰਮ ਜੈਨਰੇਟ ਕੀਤਾ ਜਾਵੇਗਾ: 10, 9, 8, 7, 6

- ਸਟ੍ਰਿੰਗ **(String)** : ਪਾਈਥਨ ਵਿੱਚ ਸਟ੍ਰਿੰਗਸ ਨੂੰ ਯੂਨੀਕੋਡ ਕਰੈਕਟਰ (Unicode character) ਮੁੱਲਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਕਰੈਕਟਰ (Character) ਡਾਟਾ ਟਾਈਪ ਮੌਜੂਦ ਨਹੀਂ ਹੈ। ਇੱਕ ਸਿੰਗਲ ਅੱਖਰ (Character) ਨੂੰ ਵੀ ਇੱਕ ਸਟ੍ਰਿੰਗ ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਸਟ੍ਰਿੰਗ ਮੁੱਲ ਨੂੰ ਸਿੰਗਲ, ਡਬਲ ਜਾਂ ਟ੍ਰਿਪਲ ਕੋਟਸ ਦੀ ਵਰਤੋਂ ਨਾਲ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ: 'Hi', "Hello", """Welcome"""



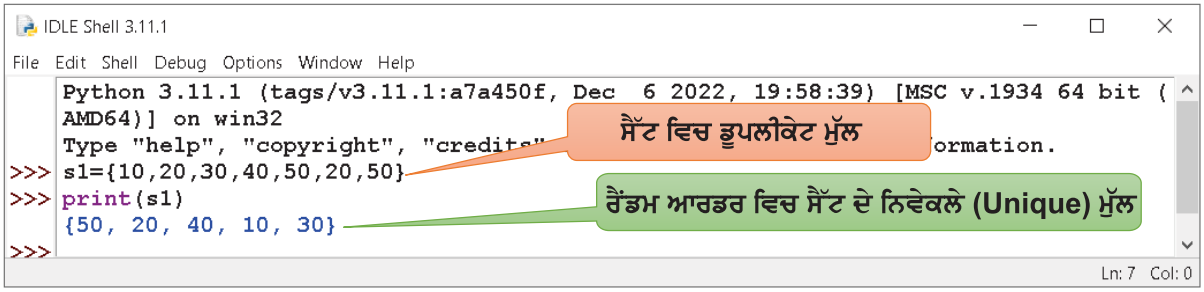
ਚਿੱਤਰ 3.4: ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਸਿਕੁਐਂਸ ਟਾਈਪਸ ਲਈ ਉਦਾਹਰਨ ਕੋਡ

ਸਿਕੁਐਂਸ (Sequence) ਇਕੋ (Similar) ਟਾਈਪ ਜਾਂ ਵੱਖ-ਵੱਖ ਟਾਈਪ ਦੇ ਮੁੱਲਾਂ ਦਾ ਇੱਕ ਕ੍ਰਮਬੱਧ ਸੰਗ੍ਰਹਿ (Ordered Collection) ਹੁੰਦਾ ਹੈ। ਲਿਸਟ, ਟਪਲ, ਰੇਂਜ਼ ਅਤੇ ਸਟ੍ਰਿੰਜ਼ ਪਾਈਥਨ ਵਿਚ ਆਮ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਸਿਕੁਐਂਸ ਟਾਈਪਸ ਹਨ।

3.1.1.4 ਪਾਈਥਨ ਵਿਚ ਸੈੱਟ ਡਾਟਾ ਟਾਈਪ (Set Data Type in Python):

ਸੈੱਟ ਤੱਤਾਂ ਦਾ ਇੱਕ ਅਨਆਰਡਰਡ ਸੰਗ੍ਰਹਿ (Unordered Collection) ਹੁੰਦਾ ਹੈ। ਸੈੱਟ ਦੇ ਐਲੀਮੈਂਟਸ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਸਾਨੂੰ ਕਰਲੀ ਬਰੈਕਟਾਂ ({}) ਵਿੱਚ ਕਾਮਿਆਂ (Commas) ਨਾਲ ਵੱਖ ਕੀਤੇ ਮੁੱਲ ਲਿਖਣੇ ਪੈਂਦੇ ਹਨ। ਉਦਾਹਰਨ ਲਈ: {10, 20, 30,40, 50}। ਸੈੱਟ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਮੁੱਲ ਨਹੀਂ ਹੁੰਦੇ। ਜੇਕਰ ਅਸੀਂ ਸੈੱਟ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਮੁੱਲਾਂ ਨੂੰ ਲਿਖਦੇ ਹਾਂ, ਤਾਂ ਇਹ ਕੋਈ ਐਰਰ (Error) ਨਹੀਂ ਦਿਖਾਵੇਗਾ, ਪਰੰਤੂ ਇਹ ਆਉਟਪੁੱਟ ਵਿੱਚ ਸਿਰਫ਼

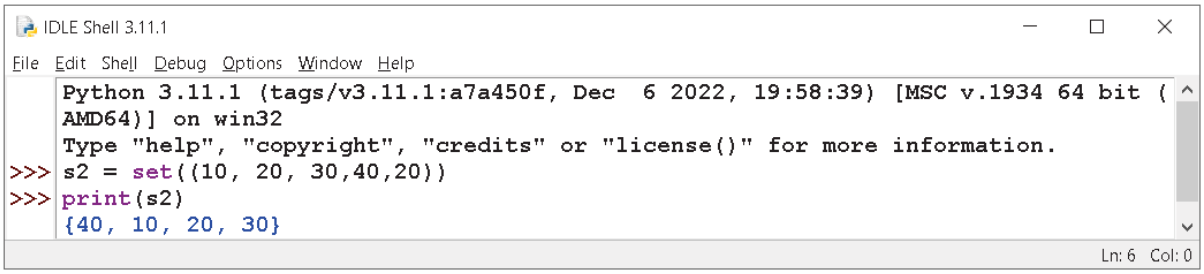
ਵੱਖਰੇ (Distinct) ਮੁੱਲ ਹੀ ਦਿਖਾਏਗਾ, ਭਾਵ ਸੈੱਟ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਮੁੱਲਾਂ ਨੂੰ ਨਜ਼ਰਅੰਦਾਜ਼ ਕੀਤਾ ਜਾਵੇਗਾ। ਕਿਉਂਕਿ ਸੈੱਟ ਵਿੱਚ ਆਈਟਮਾਂ ਅਨਆਰਡਰਡ (ਬਿਨਾਂ ਕ੍ਰਮਬੱਧ) ਹੁੰਦੀਆਂ ਹਨ, ਇਸਲਈ ਸੈੱਟ ਦੀਆਂ ਆਈਟਮਾਂ ਬੇਤਰਤੀਬ ਕ੍ਰਮ (Random order) ਵਿੱਚ ਦਿਖਾਈ ਦੇਣਗੀਆਂ। ਸੈੱਟ ਦਾ ਸਾਧਾਰਣ ਪ੍ਰੋਗਰਾਮ ਕੋਡ ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਹੈ:



ਚਿੱਤਰ 3.5: ਸੈੱਟ ਦਾ ਉਦਾਹਰਨ ਕੋਡ ਅਤੇ ਆਉਟਪੁੱਟ

ਇਹ ਜ਼ਰੂਰੀ ਨਹੀਂ ਹੈ ਕਿ ਇੱਕ ਸੈੱਟ ਦੇ ਸਾਰੇ ਤੱਤ ਇੱਕੋ ਡਾਟਾ ਟਾਈਪ ਦੇ ਹੋਣ। ਸੈੱਟ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਮਿਕਸਡ (mixed) ਡਾਟਾ ਟਾਈਪਸ ਦੇ ਮੁੱਲ ਵੀ ਲਿਖੇ ਜਾ ਸਕਦੇ ਹਨ। ਬਿਲਟ-ਇਨ set() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵੀ ਸੈੱਟ ਬਣਾਏ ਜਾ ਸਕਦੇ ਹਨ।

ਉਦਾਹਰਣ ਲਈ:



ਚਿੱਤਰ 3.6: set() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਸੈੱਟ ਬਨਾਉਣਾ

ਇੱਕ ਸੈੱਟ ਤੱਤਾਂ ਦਾ ਇੱਕ ਅਨਆਰਡਰਡ ਸੰਗ੍ਰਹਿ (Collection) ਹੁੰਦਾ ਹੈ। ਸੈੱਟ ਦੇ ਐਲੀਮੈਂਟਸ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਸਾਨੂੰ ਕਰਲੀ ਬਰੈਕਟਾਂ ({ }) ਵਿੱਚ ਕਾਮਿਆਂ (Commas) ਨਾਲ ਵੱਖ ਕੀਤੇ ਮੁੱਲ ਲਿਖਣੇ ਪੈਂਦੇ ਹਨ। ਸੈੱਟ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਮੁਲਾਂ ਨੂੰ ਨਜ਼ਰਅੰਦਾਜ਼ ਕੀਤਾ ਜਾਵੇਗਾ। ਕਿਉਂਕਿ ਸੈੱਟ ਵਿੱਚ ਆਈਟਮਾਂ ਅਨਆਰਡਰਡ ਹੁੰਦੀਆਂ ਹਨ, ਇਸਲਈ ਸੈੱਟ ਦੀਆਂ ਆਈਟਮਾਂ ਬੇਤਰਤੀਬ (Random) ਕ੍ਰਮ ਵਿੱਚ ਦਿਖਾਈ ਦੇਣਗੀਆਂ। ਇਹ ਜ਼ਰੂਰੀ ਨਹੀਂ ਹੈ ਕਿ ਇੱਕ ਸੈੱਟ ਦੇ ਸਾਰੇ ਤੱਤ ਇੱਕੋ ਡਾਟਾ ਟਾਈਪ ਦੇ ਹੋਣ।

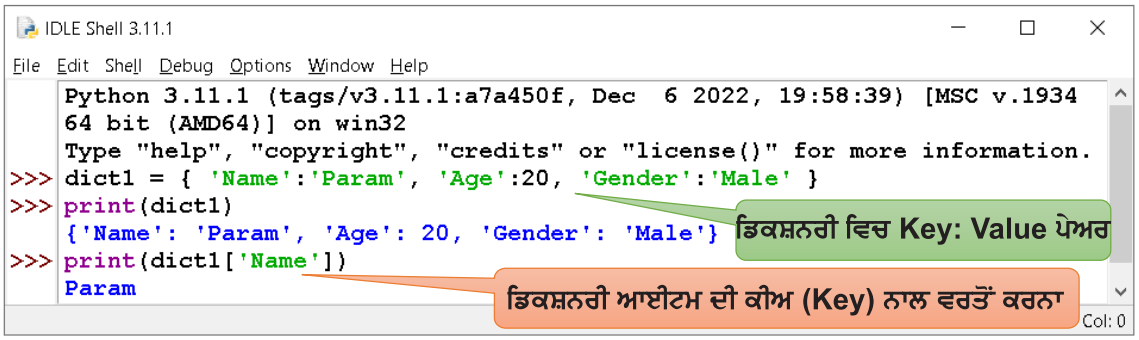
3.1.1.5 ਪਾਈਥਨ ਵਿਚ ਮੈਪਿੰਗ ਡਾਟਾ ਟਾਈਪ (Mappings Data Types in Python):

ਮੈਪਿੰਗ ਡਾਟਾ ਟਾਈਪਸ ਉਹ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਜਿਨ੍ਹਾਂ ਵਿੱਚ ਕੀਅਜ਼ (Keys) ਨੂੰ ਆਰਬਿਟਰੇਰੀ (Arbitrary) ਮੁੱਲਾਂ/ਆਬਜੈਕਟਾਂ ਨਾਲ ਮੈਪ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਦੂਜੇ ਸ਼ਬਦਾਂ ਵਿੱਚ ਅਸੀਂ ਕਹਿ ਸਕਦੇ ਹਾਂ ਕਿ ਮੈਪਿੰਗ ਟਾਈਪਸ ਵਿੱਚ ਤੱਤਾਂ (Elements) ਨੂੰ **Key: Value** ਜੋੜਿਆਂ (Pairs) ਦੇ ਰੂਪ ਵਿੱਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਸਿਰਫ ਡਿਕਸ਼ਨਰੀ ਹੀ ਮੈਪਿੰਗ ਕਿਸਮ ਦੀ ਡਾਟਾ-ਟਾਈਪ ਹੈ:

ਡਿਕਸ਼ਨਰੀ (Dictionary) ਜੋੜਿਆਂ ਦੇ ਰੂਪ ਵਿੱਚ ਡਾਟਾ ਆਈਟਮਾਂ ਦਾ ਇੱਕ ਅਨਆਰਡਰਡ ਕ੍ਰਮ ਹੁੰਦਾ ਹੈ। ਅਸੀਂ Key-value ਜੋੜਿਆਂ ਨੂੰ ਵੱਖ ਕਰਨ ਲਈ ਉਹਨਾਂ ਵਿਚਕਾਰ ਇੱਕ ਕੌਲਨ (:) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ **Key:value**। ਇਹਨਾਂ keys ਨੂੰ ਡਿਕਸ਼ਨਰੀ ਵਿੱਚਲੀਆਂ ਆਈਟਮਾਂ ਨੂੰ ਅਸੈਸ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਕਿਉਂਕਿ ਅਸੀਂ ਆਈਟਮਾਂ ਨੂੰ ਅਸੈਸ ਕਰਨ ਲਈ keys ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ, ਇਸਲਈ ਇਹ keys ਡੁਪਲੀਕੇਟ ਨਹੀਂ ਹੋ ਸਕਦੀਆਂ, ਪਰ values ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਆਈਟਮਾਂ ਹੋ ਸਕਦੀਆਂ ਹਨ। ਡਿਕਸ਼ਨਰੀ ਦੀਆਂ ਆਈਟਮਾਂ ਕਰਲੀ ਬਰੈਕਟਾਂ ਅੰਦਰ ਲਿਖੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। **'Key'** ਭਾਗ ਲਈ ਨੰਬਰ ਜਾਂ ਸਟ੍ਰਿੰਗ ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ, ਜਦੋਂ ਕਿ **'Value'** ਭਾਗ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦਾ ਹੋ ਸਕਦਾ ਹੈ।

ਮੈਪਿੰਗ ਟਾਈਪਸ ਵਿੱਚ ਤੱਤਾਂ **(Elements)** ਨੂੰ Keys: Value ਜੋੜਿਆਂ ਦੇ ਰੂਪ ਵਿੱਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਸਿਰਫ ਡਿਕਸ਼ਨਰੀ ਹੀ ਮੈਪਿੰਗ ਕਿਸਮ ਦੀ ਡਾਟਾ-ਟਾਈਪ ਹੈ ਜੋ ਡਾਟਾ ਆਈਟਮਾਂ ਦੇ ਇੱਕ ਅਨਆਰਡਰਡ ਕ੍ਰਮ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ।

ਹੇਠਾਂ ਦਿੱਤਾ ਪ੍ਰੋਗਰਾਮ ਡਿਕਸ਼ਨਰੀ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਅਤੇ ਇਸਦੇ ਮੁੱਲ ਨੂੰ ਦਿਖਾਉਣ ਸਬੰਧੀ ਉਦਾਹਰਣ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ:



ਚਿੱਤਰ: 3.7 ਡਿਕਸ਼ਨਰੀ - ਮੈਪਿੰਗ ਟਾਈਪ ਲਈ ਉਦਾਹਰਨ ਕੋਡ

ਨੋਟ: ਡਿਕਸ਼ਨਰੀ ਕੁੰਜੀਆਂ **(Keys)** ਕੇਸ ਸੰਵੇਦਨਸ਼ੀਲ **(Case Sensitive)** ਹੁੰਦੀਆਂ ਹਨ, ਇੱਕੋ ਨਾਮ ਪਰ ਕੁੰਜੀ **(Key)** ਦੇ ਵੱਖੋ-ਵੱਖਰੇ ਕੇਸਾਂ ਨੂੰ ਵੱਖਰਾ ਸਮਝਿਆ ਜਾਵੇਗਾ।

3.1.1.6 ਪਾਈਥਨ ਵਿਚ None ਡਾਟਾ ਡਾਈਪ (None Data Type in Python):

ਪਾਈਥਨ ਵਿੱਚ None ਇਕ ਵਿਸ਼ੇਸ਼ ਡਾਟਾ ਟਾਈਪ ਹੈ। ਇਹ ਟਾਈਪ ਸਿਰਫ਼ ਮੁੱਲ ਦੀ ਅਣਹੋਂਦ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। None ਕੀਵਰਡ ਦੀ ਵਰਤੋਂ ਜਾਂ ਤਾਂ null ਮੁੱਲ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜਾਂ ਇਸਦੀ ਵਰਤੋਂ ਉਸ ਸਮੇਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜਦੋਂ ਸਾਡੇ ਕੋਲ ਵੇਰੀਏਬਲ ਨੂੰ ਅਸਾਈਨ ਕਰਨ ਲਈ ਕੋਈ ਮੁੱਲ ਨਾ ਹੋਵੇ। None ਮੁੱਲ 0, False, ਜਾਂ ਖਾਲੀ ਸਟ੍ਰਿੰਗ ਦੀ ਤਰ੍ਹਾਂ ਨਹੀਂ ਹੁੰਦਾ। None ਆਪਣੇ ਆਪ ਵਿਚ ਡਾਟਾ ਦੀ ਇਕ ਕਿਸਮ (None Type) ਹੈ। ਅਸੀਂ ਕਿਸੇ ਵੀ ਵੇਰੀਏਬਲ ਨੂੰ None ਮੁੱਲ ਅਸਾਈਨ ਕਰ ਸਕਦੇ ਹਾਂ।

TEST Yourself

3.2

ਹੇਠਾਂ ਦਿਖਾਏ ਡਾਟਾ ਮੁੱਲਾਂ ਲਈ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਢੁਕਵੀਆਂ ਡਾਟਾ ਟਾਈਪਸ ਲਿਖੋ।

ੳ) ਇਕ ਸਾਲ ਦੇ ਮਹੀਨਿਆਂ ਦੀ ਗਿਣਤੀ ਲਈ _____

ਅ) ਪੰਜਾਬ ਦਾ ਵਸਨੀਕ ਹੈ ਜਾਂ ਨਹੀਂ _____

ੲ) ਮੋਬਾਈਲ ਨੰਬਰ _____

ਸ) ਚੱਕਰ ਦਾ ਖੇਤਰ _____

ਹ) ਵਿਦਿਆਰਥੀ ਦਾ ਨਾ _____

ਕ) ਵਿਦਿਆਰਥੀ ਦਾ ਪਤਾ _____

ਖ) **[12, 5.6, 29, "Hello"]** _____

ਗ) **(12, 5.6. 29, "Hello")** _____

ਘ) **(10, 20, 30, 40, 50)** _____

ਙ) **{'RollNo':101, 'Marks':495, 'Result':'Pass'}** _____

3.1.2 ਮਿਊਟੇਬਲ ਅਤੇ ਇਮਮਿਊਟੇਬਲ ਟਾਈਪਸ (Mutable and Immutable Types):

ਜਿਹਨਾਂ ਡਾਟਾ ਆਬਜੈਕਟਸ ਦੀ ਵਿਆਖਿਆ ਅਸੀਂ ਹੁਣ ਤੱਕ ਕੀਤੀ ਹੈ ਉਹਨਾਂ ਨੂੰ ਪ੍ਰੋਸੈਸਿੰਗ ਲਈ ਕੰਪਿਊਟਰ ਦੀ ਮੈਮਰੀ ਵਿੱਚ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਵਿੱਚੋਂ ਕੁੱਝ ਮੁੱਲਾਂ ਵਿੱਚ ਪ੍ਰੋਸੈਸਿੰਗ ਦੌਰਾਨ ਬਦਲਾਵ (modify) ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ, ਜਦੋਂ ਕਿ ਕੁਝ ਹੋਰਾਂ ਨੂੰ ਮੈਮਰੀ ਵਿੱਚ ਸਟੋਰ ਹੋਣ ਤੋਂ ਬਾਅਦ ਬਦਲਿਆ ਨਹੀਂ ਜਾ ਸਕਦਾ:

- **ਇਮਮਿਊਟੇਬਲ ਟਾਈਪਸ (Immutable Types):** ਇਮਮਿਊਟੇਬਲ ਟਾਈਪਸ ਉਹ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਜਿਨ੍ਹਾਂ ਦੇ ਕੰਟੈਂਟਸ ਨੂੰ ਬਨਾਉਣ (creation) ਤੋਂ ਬਾਅਦ ਬਦਲਣ ਦੀ ਆਗਿਆ ਨਹੀਂ ਹੁੰਦੀ। ਉਦਾਹਰਨ ਲਈ: ਨੰਬਰ, ਸਟ੍ਰਿੰਗਜ਼, ਅਤੇ ਟਪਲਸ ਇਮਿਊਟੇਬਲ ਟਾਈਪਸ ਦੀਆਂ ਚੰਗੀਆਂ ਉਦਾਹਰਣਾਂ ਹਨ ਕਿਉਂਕਿ ਅਸੀਂ ਇਹਨਾਂ ਨੂੰ ਬਨਾਉਣ ਤੋਂ ਬਾਅਦ ਇਹਨਾਂ ਡਾਟਾ ਆਬਜੈਕਟਸ ਦੇ ਕੰਟੈਂਟਸ ਨੂੰ ਬਦਲ (modify) ਨਹੀਂ ਸਕਦੇ। ਇਮਮਿਊਟੇਬਲ ਟਾਈਪਸ ਵਿੱਚ ਡਾਟਾ-ਆਈਟਮਾਂ ਨੂੰ ਜੋੜਨਾ (add), ਮਿਟਾਉਣਾ (delete), ਇਨਸਰਟ (insert) ਕਰਨਾ ਅਤੇ ਮੁੜ-ਵਿਵਸਥਿਤ (rearrange) ਕਰਨਾ ਸੰਭਵ ਨਹੀਂ ਹੈ।
- **ਮਿਊਟੇਬਲ ਟਾਈਪਸ (Mutable Types) :** ਮਿਊਟੇਬਲ ਟਾਈਪਸ ਉਹ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਜਿਨ੍ਹਾਂ ਦੇ ਕੰਟੈਂਟਸ ਨੂੰ ਬਨਾਉਣ ਤੋਂ ਬਾਅਦ (creation) ਬਦਲਣ (change) ਦੀ ਇਜਾਜ਼ਤ ਹੁੰਦੀ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਲਿਸਟ ਅਤੇ ਡਿਕਸ਼ਨਰੀ ਮਿਊਟੇਬਲ ਟਾਈਪਸ ਦੀਆਂ ਚੰਗੀਆਂ ਉਦਾਹਰਣਾਂ ਹਨ, ਜਿਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਅਸੀਂ ਉਹਨਾਂ ਦੇ ਤੱਤਾਂ (ਡਾਟਾ-ਆਈਟਮਾਂ) ਨੂੰ ਬਨਾਉਣ ਤੋਂ ਬਾਅਦ ਬਦਲ ਸਕਦੇ ਹਾਂ। ਮਿਊਟੇਬਲ ਟਾਈਪਸ ਵਿੱਚ ਡਾਟਾ-ਆਈਟਮਾਂ ਨੂੰ ਜੋੜਨਾ (add), ਮਿਟਾਉਣਾ (delete), ਇਨਸਰਟ (insert) ਕਰਨਾ ਅਤੇ ਮੁੜ-ਵਿਵਸਥਿਤ (rearrange) ਕਰਨਾ ਸੰਭਵ ਹੁੰਦਾ ਹੈ।

3.3 TEST YOURSELF

ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਡਾਟਾ-ਟਾਈਪਸ ਲਈ ਮਿਊਟੇਬੂਲ ਜਾਂ ਇਮਮਿਊਟੇਬਲ ਟਾਈਪ ਲਿਖੋ।

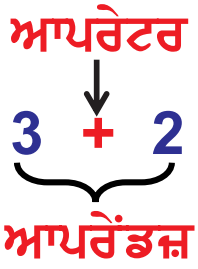
ੳ) ਨੰਬਰ _____ ਸ) ਡਿਕਸ਼ਨਰੀ _____

ਅ) ਸਟ੍ਰਿੰਗ _____ ਹ) ਟਪਲ _____

ੲ) ਲਿਸਟ _____ ਕ) ਸੈੱਟ _____

3.2 ਪਾਈਥਨ ਵਿੱਚ ਆਪਰੇਟਰਜ਼ (OPERATORS IN PYTHON)

ਆਪਰੇਟਰ ਕਿਸੇ ਵੀ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੀ ਬੁਨਿਆਦ ਹੁੰਦੇ ਹਨ। ਇਹ ਉਹ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਹੁੰਦੇ ਹਨ ਜੋ ਆਪਰੇਡਾਂ ਉਪਰ ਸਾਧਾਰਣ ਆਪਰੇਸ਼ਨ (ਪ੍ਰੋਸੈਸਿੰਗ) ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਆਪਰੇਟਰ ਆਮ ਤੌਰ 'ਤੇ ਅੰਕਗਣਿਤ (arithmetic) ਜਾਂ ਲਾਜ਼ੀਕਲ (logical) ਆਪਰੇਸ਼ਨ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਕੱਲਾ ਆਪਰੇਟਰ ਆਪਣੇ ਆਪ ਵਿਚ ਕੋਈ ਕੰਮ ਨਹੀਂ ਕਰ ਸਕਦਾ। ਇਸਨੂੰ ਆਪਣਾ ਕੰਮ ਕਰਨ ਲਈ ਇੱਕ ਜਾਂ ਇੱਕ ਤੋਂ ਵੱਧ ਆਪਰੇਂਡਾਂ ਦੀ ਜ਼ਰੂਰਤ ਪੈਂਦੀ ਹੈ। ਆਪਰੇਂਡ ਉਹ ਮੁੱਲ ਹੁੰਦੇ ਹਨ ਜਿਨ੍ਹਾਂ ਉਪਰ ਆਪਰੇਟਰ ਆਪਣਾ ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰਦੇ ਹਨ। ਉਦਾਹਰਣ ਲਈ:



ਚਿੱਤਰ 3.8: ਆਪਰੇਟਰਜ਼ ਅਤੇ ਆਪਰੇਂਡਜ਼ ਦੀ ਧਾਰਨਾ

ਉਪਰੋਕਤ ਉਦਾਹਰਨ ਵਿੱਚ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ + ਇੱਕ **ਆਪਰੇਟਰ** ਹੈ ਜੋ 3 ਅਤੇ 2 ਮੁੱਲਾਂ 'ਤੇ ਆਪਣਾ ਕੰਮ ਕਰ ਰਿਹਾ ਹੈ। ਇਹਨਾਂ ਮੁੱਲਾਂ (3 ਅਤੇ 2) ਨੂੰ **ਆਪਰੇਂਡ** ਕਿਹਾ ਜਾਵੇਗਾ। ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਡਾਂ ਦੇ ਅਜਿਹੇ ਵੈਧ ਸੁਮੇਲ (Valid Combination) ਨੂੰ **ਐਕਸਪ੍ਰੈਸ਼ਨ** (Expression) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਵੀ ਅਸੀਂ ਮੁੱਲਾਂ/ਆਪਰੇਂਡਾਂ 'ਤੇ ਕੋਈ ਕਾਰਵਾਈ ਕਰਦੇ ਹਾਂ, ਸਾਨੂੰ ਇੱਕ ਨਵਾਂ ਮੁੱਲ ਮਿਲਦਾ ਹੈ (ਜਿਸ ਨੂੰ ਆਪਰੇਸ਼ਨ ਦਾ ਨਤੀਜਾ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ)। ਉਪਰੋਕਤ ਉਦਾਹਰਨ ਵਿੱਚ ਮੁੱਲਾਂ 'ਤੇ ਜੋੜ (Addition) ਦਾ ਕੰਮ ਕਰਨ ਤੋਂ ਬਾਅਦ ਸਾਨੂੰ 5 ਮੁੱਲ ਮਿਲੇਗਾ।

ਆਪਰੇਟਰ ਉਹ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਹੁੰਦੇ ਹਨ ਜੋ ਆਪਰੇਂਡਜ਼ ਉਪਰ ਸਧਾਰਣ ਆਪਰੇਸ਼ਨ (ਪ੍ਰੋਸੈਸਿੰਗ) ਲਾਗੂ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਆਪਰੇਂਡਜ਼ ਉਹ ਮੁੱਲ ਹੁੰਦੇ ਹਨ ਜਿਨ੍ਹਾਂ ਉਪਰ ਆਪਰੇਟਰ ਆਪਣਾ ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰਦੇ ਹਨ।

ਪਾਈਥਨ ਵਿਚ ਬਹੁਤ ਸਾਰੇ ਆਪਰੇਟਰ ਮੌਜੂਦ ਹਨ। ਆਮ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਆਪਰੇਟਰਾਂ ਨੂੰ ਅੱਗੇ ਲਿਖੀਆਂ ਸ਼੍ਰੇਣੀਆਂ ਵਿਚ ਵੰਡਿਆ ਜਾ ਸਕਦਾ ਹੈ:

- ਅਰਿੱਥਮੈਟਿਕ ਆਪਰਟੇਰਜ਼ (Arithmetic Operators)
- ਰਿਲੇਸ਼ਨਲ (ਤੁਲਨਾਤਮਕ) ਆਪਰੇਟਰਜ਼ (Relational (Comparison) Operators)
- ਲਾਜੀਕਲ ਆਪਰੇਟਰਜ਼ (Logical Operators)
- ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰਜ਼ (Assignment Operators)
- ਮੈਂਬਰਸ਼ਿਪ ਆਪਰੇਟਰਜ਼ (Membership Operators)

ਇਹਨਾਂ ਆਪਰੇਟਰਾਂ ਨੂੰ ਹੇਠਾਂ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਸਮਝਾਇਆ ਗਿਆ ਹੈ:

3.2.1 ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਜ਼

ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਾਂ ਦੀ ਵਰਤੋਂ ਨੂਮੇਰਿੱਕ ਮੁੱਲਾਂ (numerical values) ਉਪਰ ਅਰਿੱਥਮੈਟਿਕ (ਗਣਿਤਿਕ/mathematical) ਆਪਰੇਸ਼ਨ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤਾ ਟੇਬਲ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਾਂ ਨੂੰ ਉਹਨਾਂ ਦੇ ਆਪਰੇਸ਼ਨਾਂ ਅਤੇ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਦਰਸਾ ਰਿਹਾ ਹੈ:

ਆਪਰੇਟਰ ਦਾ ਚਿੰਨ੍ਹ	ਆਪਰੇਟਰ ਦੀ ਵਿਆਖਿਆ	ਉਦਾਹਰਣ
+	ਜੋੜ ਕਰਨ ਲਈ (Addition)	5 + 10 = 15
-	ਘਟਾਓ ਕਰਨ ਲਈ (Subtraction)	5 – 10 = -5
*	ਗੁਣਾ ਕਰਨ ਲਈ (Multiplication)	5 * 10 = 50
/	ਰੀਅਲ ਡਿਵੀਜ਼ਨ ਕਰਨ ਲਈ (Real Division)	19 / 5 = 3.8
//	ਇੰਟੀਜ਼ਰ ਡਿਵੀਜ਼ਨ ਕਰਨ ਲਈ (Integer Division)	19 // 5 = 3
%	ਮਾਡੂਲਸ (ਸ਼ੇਸ਼ਫਲ) ਪਤਾ ਕਰਨ ਲਈ (Modulus)	19 % 5 = 4
**	ਐਕਸਪੋਨੈਂਟ ਪਤਾ ਕਰਨ ਲਈ (Exponent)	5 ** 2 = 25

ਟੇਬਲ 3.1: ਪਾਈਥਨ ਵਿਚ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਜ਼

3.4 ਹੇਠਾਂ ਦਿਤੀਆਂ ਸਟੇਟਮੈਂਟ ਦੀ ਆਊਟਪੁਟ ਲਿਖੋ।

ੳ)
```
a = 3 ** 2 + 10
b = 5 * 6
c = 7.0/8.0
print("Values:",a,b,c)
```
ੳ) ਆਊਟਪੁੱਟ _____

ਅ)
```
num1,num2 = 2, 3
num3,num2 = num1, num2*2
num1 = num1 ** 2
print(num1, num2, num3)
```
ਅ) ਆਊਟਪੁੱਟ _____

3.2.2 ਰਿਲੇਸ਼ਨਲ (ਤੁਲਨਾਤਮਕ) ਆਪਰੇਟਰਜ਼ (Relational (Comparison) Opertors) :

ਰਿਲੇਸ਼ਨਲ ਆਪਰੇਟਰਾਂ ਨੂੰ ਤੁਲਨਾਤਮਕ (Comparison) ਆਪਰੇਟਰ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਕਿਉਂਕਿ ਇਹ ਆਪਰੇਟਰ ਮੁੱਲਾਂ ਦੀ ਤੁਲਨਾ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਤੁਲਨਾ ਦੌਰਾਨ, ਖੱਬੇ ਆਪਰੇਂਡ ਦੇ ਮੁੱਲ ਦੀ ਤੁਲਨਾ ਸੱਜੇ ਆਪਰੇਂਡ ਦੇ ਮੁੱਲ ਨਾਲ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਤੁਲਨਾ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਇਹ ਓਪਰੇਟਰ ਸਹੀ (True) ਜਾਂ ਗਲਤ (False) ਮੁੱਲ ਵਾਪਸ ਕਰਦੇ ਹਨ। ਇਹ ਆਪਰੇਟਰ ਆਮ ਤੌਰ 'ਤੇ ਵੱਖ-ਵੱਖ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਟੈਸਟ-ਕੰਡੀਸ਼ਨਾਂ (Test-Conditions) ਵਿੱਚ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। (ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਾਂ ਬਾਰੇ ਇਸ ਕਿਤਾਬ ਦੇ ਅਗਲੇ ਪਾਠ ਵਿੱਚ ਵਿਸਥਾਰ ਵਿੱਚ ਚਰਚਾ ਕੀਤੀ ਜਾਵੇਗੀ।) ਹੇਠਾਂ ਦਿੱਤਾ ਟੇਬਲ ਰਿਲੇਸ਼ਨਲ ਆਪਰੇਟਰਾਂ ਨੂੰ ਉਹਨਾਂ ਦੇ ਆਪਰੇਸ਼ਨਾਂ ਅਤੇ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਦਰਸਾ ਰਿਹਾ ਹੈ:

ਆਪਰੇਟਰ	ਆਪਰੇਟਰ ਦੀ ਵਿਆਖਿਆ	ਉਦਾਹਰਣ
==	ਬਰਾਬਰ ਹੈ (Equal)	4 = = 5 ਗਲਤ (False) ਹੈ।
!=	ਬਰਾਬਰ ਨਹੀਂ ਹੈ (Not Equal)	4! = 5 ਸਹੀ (True) ਹੈ।
>	ਵੱਡਾ ਹੈ (Greater Than)	4 > 5 ਗਲਤ (False) ਹੈ।
<	ਛੋਟਾ ਹੈ (Less Than)	4 < 5 ਸਹੀ (True) ਹੈ।
>=	ਵੱਡਾ ਹੈ ਜਾਂ ਬਰਾਬਰ ਹੈ (Greater than or Equal to)	4 > = 5 ਗਲਤ (False) ਹੈ।
<=	ਛੋਟਾ ਹੈ ਜਾਂ ਬਰਾਬਰ ਹੈ (Less than or Equal to)	4 < = 5 ਸਹੀ (True) ਹੈ।

ਟੇਬਲ 3.2: ਪਾਈਥਨ ਵਿਚ ਰਿਲੇਸ਼ਨਲ ਆਪਰੇਟਰਜ਼

3.5

TEST YOURSELF

ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦੀ ਆਊਟਪੁੱਟ ਲਿਖੋ।

ੳ) `print('Hello' == 'HELLO')` _____

ਅ) `print(10 < 16)` _____

ੲ) `print(10 == 60)` _____

ਸ) `print(50>=50)` _____

3.2.3 ਲਾਜੀਕਲ (ਬੂਲੀਅਨ) ਆਪਰੇਟਰਜ਼ (Logical Boolean) Operators) :

ਲਾਜ਼ੀਕਲ ਆਪਰੇਟਰਾਂ ਨੂੰ ਬੂਲੀਅਨ ਆਪਰੇਟਰ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਕਿਉਂਕਿ ਇਹ ਆਪਰੇਟਰ ਆਪਰੇਡਾਂ 'ਤੇ ਲਾਜ਼ੀਕਲ ਆਪਰੇਸ਼ਨ ਕਰਵਾਉਣ ਤੋਂ ਬਾਅਦ ਨਤੀਜੇ ਦੇ ਅਧਾਰ 'ਤੇ ਬੂਲੀਅਨ ਮੁੱਲ ਵਾਪਸ ਕਰਦੇ ਹਨ। ਇਹਨਾਂ ਆਪਰੇਟਰਾਂ ਨੂੰ ਆਮ ਤੌਰ 'ਤੇ ਉਹਨਾਂ ਸਥਿਤੀਆਂ ਵਿੱਚ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਸਾਨੂੰ ਕਈ ਟੈਸਟ-ਕੰਡੀਸ਼ਨਾਂ (Multiple Test Conditions) ਦੀ ਤੁਲਨਾ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਪੈਂਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤਾ ਟੇਬਲ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਪਾਈਥਨ ਵਿਚ ਵਰਤੇ ਜਾਂਦੇ ਲਾਜ਼ੀਕਲ ਓਪਰੇਟਰਾਂ ਨੂੰ ਦਰਸਾ ਰਿਹਾ ਹੈ:

ਆਪਰੇਟਰ	ਆਪਰੇਟਰ ਦੀ ਵਿਆਖਿਆ	ਉਦਾਹਰਣ
and	ਇਹ ਆਪਰੇਟਰ ਉਸ ਸਮੇਂ true ਵਾਪਸ ਕਰਦਾ ਹੈ ਜਦੋਂ ਦੋਵੇਂ ਆਪਰੇਂਡ true ਜਾਂ ਨਾਨ-ਜ਼ੀਰੋ ਹੋਣ।	(5>4 and 5>2) ਸਹੀ (true) ਵਾਪਿਸ ਕਰੇਗਾ
or	ਇਹ ਆਪਰੇਟਰ ਉਸ ਸਮੇਂ true ਵਾਪਸ ਕਰਦਾ ਹੈ ਜਦੋਂ ਦੋਵਾਂ ਵਿਚੋਂ ਕੋਈ ਇਕ ਆਪਰੇਂਡ true ਜਾਂ ਨਾਨ-ਜ਼ੀਰੋ ਹੋਵੇ।	(5>4 or 5<2) ਸਹੀ (true) ਵਾਪਿਸ ਕਰੇਗਾ
not	ਇਹ ਆਪਰੇਟਰ ਆਪਣੇ ਆਪਰੇਂਡ ਦੀ ਲਾਜ਼ੀਕਲ ਸਟੇਟ ਨੂੰ ਉਲਟਾਉਂਦਾ ਹੈ।	not (5>4) ਗਲਤ (false) ਵਾਪਿਸ ਕਰੇਗਾ।

ਟੇਬਲ 3.3: ਪਾਈਥਨ ਵਿਚ ਲਾਜੀਕਲ ਆਪਰੇਟਰਜ਼

TEST YOURSELF

3.6

ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦੀ ਆਊਟਪੁੱਟ ਲਿਖੋ।

ੳ) `print((10 != 9) and (20 >= 20))` _____

ਅ) `print((10 < 50) and (29 <= 29))` _____

ੲ) `print(not((0 < 6) or (10 == 6)))` _____

3.2.4 ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰਜ਼ (Assignment Operator):

ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ ਦੀ ਵਰਤੋਂ ਵੈਰੀਏਬਲਾਂ ਨੂੰ ਮੁੱਲ ਅਸਾਈਨ (assign) ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਭਾਵ ਇਹ ਆਪਰੇਟਰ ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਐਲੋਕੇਟ (allocate) ਕੀਤੀ ਗਈ ਮੈਮਰੀ ਵਿੱਚ ਮੁੱਲਾਂ ਨੂੰ ਸਟੋਰ ਕਰਵਾਉਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਸ ਮੰਤਵ ਲਈ = ਚਿੰਨ੍ਹ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਅਸਾਈਨ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲ = ਆਪਰੇਟਰ ਦੇ ਸੱਜੇ ਪਾਸੇ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ, ਅਤੇ ਉਹ ਵੇਰੀਏਬਲ ਜਿਸ ਵਿੱਚ ਅਸੀਂ ਮੁੱਲ ਨੂੰ ਸਟੋਰ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਉਹ = ਆਪਰੇਟਰ ਦੇ ਖੱਬੇ ਪਾਸੇ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰਾਂ ਦੀਆਂ ਕੁੱਝ ਉਦਾਹਰਣਾਂ ਹੇਠਾਂ ਦਿਤੀਆਂ ਗਈਆਂ ਹਨ:

ਉਦਾਹਰਣ 1: `a = 5`

ਇੱਥੇ 'a' ਇਕ ਵੇਰੀਏਬਲ ਹੈ ਅਤੇ ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ (=) ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸ ਨੂੰ ਇੱਕ ਸਥਿਰ (Constant) ਮੁੱਲ '5' ਅਸਾਈਨ ਕੀਤਾ ਗਿਆ ਹੈ। ਅਸਾਈਨਮੈਂਟ ਓਪਰੇਟਰ ਦਾ ਸੱਜਾ ਪਾਸਾ ਇੱਕ ਸਥਿਰ ਮੁੱਲ ਹੋ ਸਕਦਾ ਹੈ ਜਾਂ ਇਹ ਇੱਕ ਵੇਰੀਏਬਲ ਜਾਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵੀ ਹੋ ਸਕਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਅਗਲੀ ਉਦਾਹਰਣ ਵਿਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

ਉਦਾਹਰਣ 2: `x = y`

ਇਸ ਉਦਾਹਰਨ ਵਿੱਚ ਅਸੀਂ ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ ਦੇ ਸੱਜੇ ਪਾਸੇ ਇੱਕ ਵੇਰੀਏਬਲ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਹੈ ਜਿਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਵੇਰੀਏਬਲ y ਦਾ ਮੁੱਲ ਵੇਰੀਏਬਲ x ਵਿੱਚ ਸਟੋਰ ਕੀਤਾ ਜਾਵੇਗਾ।

ਉਦਾਹਰਣ 3: ਪਾਈਥਨ ਵਿੱਚ ਅਸੀਂ ਮਲਟੀਪਲ-ਅਸਾਈਨਮੈਂਟ (Multiple-Assignment) ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕਈ ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਇੱਕੋ ਆਬਜੈਕਟ ਦਾ ਰੈਫਰੈਂਸ ਦੇ ਸਕਦੇ ਹਾਂ। ਇਹ ਉਸ ਸਮੇਂ ਲਾਭਦਾਇਕ **ਹੋ ਸਕਦਾ ਹੈ ਜਦੋਂ ਅਸੀਂ ਇੱਕੋ ਸ਼ੁਰੂਆਤੀ ਮੁੱਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕਈ ਸਮਾਨ ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਇਨੀਸ਼ੀਅਲਾਈਜ਼** (Initialize) **ਕਰਨਾ ਹੋਵੇ:**

`x=y=6`

ਉਦਾਹਰਣ 4: ਪਾਈਥਨ ਵਿੱਚ ਅਸੀਂ ਕਈ ਵੇਰੀਏਬਲਾਂ ਨੂੰ ਇੱਕੋ ਸਮੇਂ ਕਈ ਮੁੱਲ ਅਸਾਈਨ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਸ ਕਿਸਮ ਦੀ ਅਸਾਈਨਮੈਂਟ ਨੂੰ ਪੈਰਲਲ ਅਸਾਈਨਮੈਂਟ (Parallel Assignment) **ਕਿਹਾ ਜਾਂਦਾ ਹੈ।**

`x, y, z = 2, 4, 6`

ਇਸ ਉਦਾਹਰਣ ਵਿਚ ਅਸੀਂ ਤਿੰਨ ਮੁੱਲਾਂ 2, 4 ਅਤੇ 6 ਨੂੰ ਤਿੰਨ ਵੇਰੀਏਬਲਾਂ x, y ਅਤੇ z ਨੂੰ ਕ੍ਰਮਵਾਰ ਅਸਾਈਨ ਕੀਤਾ ਹੈ, ਭਾਵ x=2, y=4 and z=6 ਅਸਾਈਨਮੈਂਟ ਕੀਤੀ ਗਈ ਹੈ।

ਉਦਾਹਰਣ 5: `a = b + c`

ਇਸ ਉਦਾਹਰਨ ਵਿੱਚ ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟ ਦਾ ਸੱਜਾ ਪਾਸਾ ਇੱਕ ਐਕਸਪ੍ਰੈਸ਼ਨ ਹੈ। ਐਕਸਪ੍ਰੈਸ਼ਨ (b+c) ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਇਸਦਾ ਨਤੀਜਾ ਵੇਰੀਏਬਲ 'a' ਵਿੱਚ ਸਟੋਰ ਕਰ ਦਿੱਤਾ ਜਾਵੇਗਾ।

ਉਦਾਹਰਣ 6: ਅਸੀਂ ਵੇਰੀਏਬਲ ਵਿਚ ਸਟੋਰ ਪੁਰਾਣੇ ਮੁੱਲ (Previous Value) 'ਤੇ ਕੋਈ ਕੰਮ ਕਰਕੇ ਉਸ ਵੇਰੀਏਬੂਲ ਦੇ ਮੁੱਲ ਨੂੰ ਅਪਡੇਟ ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ: `x=x+5`

ਇਸ ਉਦਾਹਰਨ ਵਿੱਚ ਅਸੀਂ ਵੇਰੀਏਬਲ 'x' ਦੇ ਪੁਰਾਣੇ ਮੁੱਲ ਵਿੱਚ 5 ਜੋੜਿਆ ਹੈ ਅਤੇ ਨਤੀਜੇ ਨੂੰ ਵਾਪਸ ਵੇਰੀਏਬਲ 'x' ਨੂੰ ਹੀ ਅਸਾਈਨ ਕੀਤਾ ਹੈ। ਅਜਿਹੀਆਂ ਅਸਾਈਨਮੈਂਟਾਂ ਨੂੰ **ਸੈਲਫ-ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟਸ** (Self-Assignment Statements) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਅਜਿਹੀਆਂ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਲਿਖਣ ਦਾ ਇੱਕ ਵਿਕਲਪਿਕ (alternative) ਤਰੀਕਾ ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਹੋ ਸਕਦਾ ਹੈ: `x+=5`

ਸੈਲਫ-ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਲਿਖਣ ਦੇ ਅਜਿਹੇ ਤਰੀਕੇ ਨੂੰ ਔਗਮੈਂਟਡ ਅਸਾਈਨਮੈਂਟ **(Augmented Assignment)** ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਔਗਮੈਂਟਡ ਅਸਾਈਨਮੈਂਟ ਅਸਲ ਵਿੱਚ ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ ਦੇ ਕੰਮ ਨੂੰ ਕਿਸੇ ਹੋਰ ਆਪਰੇਟਰ ਨਾਲ ਜੋੜ ਕੇ ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟ ਨੂੰ ਹੋਰ ਸੰਖੇਪ (Concise) ਬਣਾਉਂਦਾ ਹੈ। ਉਪਰੋਕਤ ਉਦਾਹਰਣ ਵਿਚ += ਆਪਰੇਟਰ ਅਰਿੱਥਮੈਟਿਕ ਜੋੜ ਅਤੇ ਅਸਾਈਨਮੈਂਟ ਦੇ ਕੰਮ ਨੂੰ ਇਕੱਠਾ ਕਰ ਰਿਹਾ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਆਪਰੇਟਰਾਂ ਨੂੰ ਔਗਮੈਂਟੇਡ ਆਪਰੇਟਰ ਇਸ ਲਈ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਕਿਉਂਕਿ ਇਹ ਆਪਰੇਟਰ ਆਪਣੇ ਕੰਮ ਨੂੰ ਇਕੋ ਸਮੇਂ ਦੋ ਆਪਰੇਸ਼ਨਾਂ ਤੱਕ ਵਧਾ ਕੇ **(extended or augmented)** ਅੰਜਾਮ ਦਿੰਦੇ ਹਨ, ਜਿਵੇਂ ਕਿ ਉਪਰੋਕਤ ਉਦਾਹਰਣ ਵਿੱਚ ਅਸੀਂ ਜੋੜ ਕਰਨ ਦੇ ਨਾਲ-ਨਾਲ ਅਸਾਈਨਿੰਗ ਦਾ ਕੰਮ ਵੀ ਕਰਵਾ ਰਹੇ ਹਾਂ। ਇਸ ਤਰ੍ਹਾਂ ਔਗਮੈਂਟਡ ਅਸਾਈਨਮੈਂਟ ਓਪਰੇਟਰ ਬਾਈਨਰੀ ਆਪਰੇਸ਼ਨ ਕਰਨ ਦਾ ਇੱਕ ਸ਼ਾਰਟਕੱਟ ਤਰੀਕਾ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ ਅਤੇ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿਚ ਵਰਤੇ ਗਏ ਆਪਰੇਂਡ ਨੂੰ ਹੀ ਨਤੀਜਾ ਵਾਪਸ ਕਰਦੇ ਹਨ। ਔਗਮੈਂਟਡ ਆਪਰੇਟਰਾਂ ਦੀ ਵਰਤੋਂ ਕਰਨ ਦਾ ਮੂਲ ਸਿੰਟੈਕਸ ਹੇਠਾਂ ਦਿਤਾ ਗਿਆ ਹੈ:

```
variable op = expression
```

ਇੱਥੇ op ਆਮ ਓਪਰੇਟਰ ਲਈ ਇੱਕ ਪਲੇਸਹੋਲਡਰ ਹੈ, ਜਿਸਨੂੰ ਔਗਮੈਂਟਡ ਆਪਰੇਟਰ ਵਜੋਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਅਸਲ ਵਿੱਚ ਇੱਕ ਔਗਮੈਂਟਡ ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੇਟਮੈਂਟ ਦੇ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ:

```
variable = variable op expression
```

ਹੇਠਾਂ ਦਿੱਤਾ ਟੇਬਲ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਾਂ ਨਾਲ ਔਗਮੈਂਟਡ ਆਪਰੇਟਰਾਂ ਦੀ ਲਿਸਟ ਦਿਖਾ ਰਿਹਾ ਹੈ:

ਆਪਰੇਟਰ	ਆਪਰੇਟਰ ਦੀ ਵਿਆਖਿਆ	ਉਦਾਹਰਣ	ਬਰਾਬਰ ਹੈ
+=	ਜੋੜ ਅਸਾਈਨਮੈਂਟ (Addition Assignment)	a += 5	a = a + 5
-=	ਘਟਾਓ ਅਸਾਈਨਮੈਂਟ (Subtraction Assignment)	a -= 5	a = a - 5
*=	ਗੁਣਾ ਅਸਾਈਨਮੈਂਟ (Multiplication Assignment)	a *= 5	a = a * 5
/=	ਇੰਟੀਜ਼ਰ ਡਿਵੀਜ਼ਨ ਅਸਾਈਨਮੈਂਟ (Division Assignment)	a /= 5	a = a / 5
%=	ਮਾਡੂਲਸ ਅਸਾਈਨਮੈਂਟ (Modulus Assignment)	a %= 5	a = a % 5
**=	ਐਕਸਪੋਨੈਂਟ ਅਸਾਈਨਮੈਂਟ (Exponent Assignment)	a **= 2	a = a **2
//=	ਫਲੋਰ ਡਿਵੀਜ਼ਨ ਅਸਾਈਨਮੈਂਟ (Floor Division Assignment)	a //= 3	a = a // 3

ਟੇਬਲ 3.4: ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਾਂ ਨਾਲ ਔਗਮੈਂਟਡ ਅਸਾਈਨਮੈਂਟ ਆਪਰੇਟਰ

3.7

TEST YOURSELF

ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟਸ ਦੀਆਂ ਕਿਸਮਾਂ (ਸੈਲਫ/ ਮਲਟੀਪਲ / ਪੈਰਲਲ / ਔਗਮੈਂਟਡ ਅਸਾਈਨਮੈਂਟ) ਅਤੇ ਵੇਰੀਏਬਲਾਂ ਦੇ ਮੁੱਲ ਲਿਖੋ, ਜੇਕਰ num1=6 ਅਤੇ num2=3 ਹੋਵੇ।

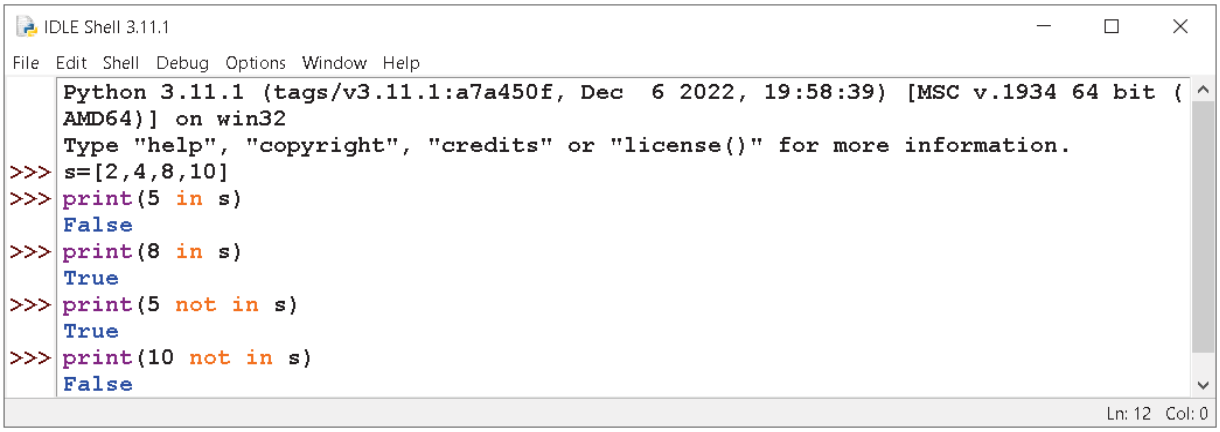
	ਸਟੇਟਮੈਂਟ	ਕਿਸਮ	ਮੁੱਲ
ੳ)	`num3, num4 = num2, num1+3`	_____	num3=___ num4= ___
ਅ)	`num5 = num4 = num1 / num2`	_____	num4=___ num5= ___
ੲ)	`num1 *= num2`	_____	num1=___ num2= ___
ਸ)	`num2 **= 2`	_____	num2=___
ਹ)	`num5 = num5 + 10`	_____	num5=___

3.2.5 ਮੈਂਬਰਸ਼ਿੱਪ ਆਪਰੇਟਰਜ਼ (Membership (Operators):

ਇਹਨਾਂ ਆਪਰੇਟਰਾਂ ਦੀ ਵਰਤੋਂ ਦਿੱਤੇ ਗਏ ਸਿਕੁਐਂਸ (Sequence) ਟਾਈਪਸ, ਜਿਵੇਂ ਕਿ ਸਟ੍ਰਿੰਗ, ਲਿਸਟਾਂ, ਜਾਂ ਟਪਲਜ਼ ਆਦਿ ਵਿੱਚ ਇੱਕ ਮੁੱਲ ਸਰਚ (search) ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਸਰਚ ਨਤੀਜੇ ਦੇ ਆਧਾਰ 'ਤੇ, ਇਹ ਆਪਰੇਟਰ ਸਹੀ (True) ਜਾਂ ਗਲਤ (False) ਮੁੱਲ ਵਾਪਸ ਕਰਦੇ ਹਨ। ਪਾਈਥਨ ਵਿੱਚ ਦੋ ਮੈਂਬਰਸ਼ਿਪ ਆਪਰੇਟਰ ਹਨ, ਜਿਵੇਂ ਕਿ: ਹੇਠਾਂ ਦਿੱਤੇ ਟੇਬਲ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

ਆਪਰੇਟਰ	ਆਪਰੇਟਰ ਦੀ ਵਿਆਖਿਆ	ਉਦਾਹਰਣ
in	ਜੇਕਰ ਦਿੱਤੇ ਗਏ ਸਿਕੁਐਂਸ ਵਿੱਚ ਨਿਰਧਾਰਤ ਮੁੱਲ ਮੌਜੂਦ ਹੁੰਦਾ ਹੈ, ਤਾਂ ਇਹ ਆਪਰੇਟਰ True ਮੁੱਲ ਵਾਪਸ ਕਰਦਾ ਹੈ ਨਹੀਂ ਤਾਂ ਇਹ False ਮੁੱਲ ਵਾਪਸ ਕਰੇਗਾ।	s=[2,4,8,10] print(8 in s) ਇਹ True ਦਿਖਾਵੇਗਾ।
not in	ਜੇਕਰ ਦਿੱਤੇ ਗਏ ਸਿਕੁਐਂਸ ਵਿੱਚ ਨਿਰਧਾਰਤ ਮੁੱਲ ਮੌਜੂਦ ਨਹੀਂ ਹੁੰਦਾ, ਤਾਂ ਇਹ ਆਪਰੇਟਰ True ਮੁੱਲ ਵਾਪਸ ਕਰਦਾ ਹੈ ਨਹੀਂ ਤਾਂ ਇਹ False ਮੁੱਲ ਵਾਪਸ ਕਰੇਗਾ।	s=[2,4,8,10] print(8 not in s) ਇਹ False ਦਿਖਾਵੇਗਾ

ਟੇਬਲ 3.5: ਪਾਈਥਨ ਵਿੱਚ ਮੈਂਬਰਸ਼ਿਪ ਆਪਰੇਟਰਜ਼



ਚਿੱਤਰ 3.9: ਮੈਂਬਰਸ਼ਿੱਪ ਆਪਰੇਟਰਾਂ ਲਈ ਪਾਈਥਨ ਕੋਡ ਦੀ ਉਦਾਹਰਨ

3.8

TEST YOURSELF

ਹੇਠਾਂ ਦਿੱਤੇ ਆਪਰੇਟਰਾਂ ਦੀ ਪਛਾਣ ਕਰਕੇ ਉਹਨਾਂ ਦੀ ਕਿਸਮ ਦਾ ਨਾਂ ਲਿਖੋ।

ੳ) `+, -, *, **, /, //, %` _____

ਅ) `and, or, not` _____

ੲ) `=, +=, -=, */, /=` _____

ਸ) `in, not in` _____

ਹ) `>, >=, <, <=, ==, !=` _____

3.3 ਐਕਸਪ੍ਰੈਸ਼ਨ ਅਤੇ ਇਸਦਾ ਮੁਲਾਂਕਣ (EXPRESSION & ITS EVALUATION)

ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਡਾਂ ਦੇ ਇੱਕ ਸਹੀ ਸੁਮੇਲ (Valid Combination) ਨੂੰ ਐਕਸਪ੍ਰੈਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਸਹੀ ਸੁਮੇਲ ਅਜਿਹਾ ਸੁਮੇਲ ਹੁੰਦਾ ਹੈ ਜੋ ਪਾਈਥਨ ਭਾਸ਼ਾ ਦੇ ਸਿੰਟੈਕਸ ਨਿਯਮਾਂ (Syntax Rules) ਦੀ ਪੁਸ਼ਟੀ ਕਰਦਾ ਹੈ। ਇੱਕ ਸਹੀ ਐਕਸਪ੍ਰੈਸ਼ਨ ਨੂੰ ਸੁਚੱਜੇ ਰੂਪ ਵਾਲੀ (Well-Formed) ਐਕਸਪ੍ਰੈਸ਼ਨ ਵੱਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਗਣਿਤ ਵਿੱਚ ਇੱਕ ਫਾਰਮੂਲੇ ਵਾਂਗ ਹੁੰਦੀ ਹੈ। ਮੁਲਾਂਕਣ (Evaluation) ਤੋਂ ਬਾਅਦ ਐਕਸਪ੍ਰੈਸ਼ਨ ਹਮੇਸ਼ਾ ਇੱਕ ਮੁੱਲ

ਵਾਪਸ ਕਰਦਾ ਹੈ। ਇਹ ਨਤੀਜਾ ਮੁੱਲ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਆਪਰੈਂਡ ਵੱਜੋਂ ਵੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਐਕਸਪ੍ਰੈਸ਼ਨ ਇੱਕ ਸਿੰਗਲ ਮੁੱਲ ਜਿੰਨੀ ਸਧਾਰਣ ਅਤੇ ਇੱਕ ਵੱਡੀ ਗਣਨਾ ਜਿੰਨੀ ਗੁੰਝਲਦਾਰ (Complex) ਵੀ ਹੋ ਸਕਦੀ ਹੈ। ਉਦਾਹਰਣ ਲਈ:

$$x = 2.9$$

ਇਹ ਇੱਕ ਸਧਾਰਨ ਐਕਸਪ੍ਰੈਸ਼ਨ (Simple Expression) ਹੈ ਜਿੱਥੇ = ਆਪਰੇਟਰ ਨੂੰ ਆਪਰੇਂਡ x ਅਤੇ 2.9 ਨਾਲ ਵਰਤਿਆ ਗਿਆ ਹੈ।

$$x = 2.9 * y + 3.6 > z - (3.4 / z)$$

ਇਹ ਕੁੱਝ ਹੱਦ ਤੱਕ ਗੁੰਝਲਦਾਰ ਐਕਸਪ੍ਰੈਸ਼ਨ (Complex Expression) ਹੈ ਜਿਸ ਵਿੱਚ ਬਹੁਤ ਸਾਰੇ ਆਪਰੇਟਰ ਅਤੇ ਆਪਰੇਂਡਜ਼ ਹਨ। ਇਸ ਉਦਾਹਰਣ ਵਿਚ =, *, +, >, – ਅਤੇ / ਆਪਰੇਟਰ ਹਨ ਅਤੇ x, 2.9, y, 3.6, 3.4 ਅਤੇ z ਆਪਰੇਂਡ ਹਨ।

ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਡਾਂ ਦੇ ਇੱਕ ਸਹੀ ਸੁਮੇਲ (Valid Combination) ਨੂੰ ਐਕਸਪ੍ਰੈਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਗਣਿਤ ਵਿੱਚ ਇੱਕ ਫਾਰਮੂਲੇ ਵਾਂਗ ਹੁੰਦੀ ਹੈ। ਇਹ ਨਤੀਜਾ ਮੁੱਲ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਇਕ ਆਪਰੈਂਡ ਵੱਜੋਂ ਵੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ।

ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਉਦਾਹਰਣ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿਚ ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਂਡਾਂ ਦੇ ਪ੍ਰਮਾਣਿਕ ਸੁਮੇਲ (Valid Combination) ਨੂੰ ਨਹੀਂ ਦਰਸਾ ਰਹੀ:

$$x + y = z;$$

ਹਾਲਾਂਕਿ ਅਸੀਂ ਇਸ ਉਦਾਹਰਨ ਵਿੱਚ ਪ੍ਰਮਾਣਿਕ (Valid) ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਡਾਂ ਦੀ ਵਰਤੋਂ ਕਰ ਰਹੇ ਹਾਂ, ਫਿਰ ਵੀ ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਡਾਂ ਦਾ ਇਹ ਸੁਮੇਲ ਇੱਕ ਪ੍ਰਮਾਣਿਕ ਐਕਸਪ੍ਰੈਸ਼ਨ (Valid Expression) ਨਹੀਂ ਬਣਾਉਂਦਾ। ਇਕ ਪ੍ਰਮਾਣਿਕ ਐਕਸਪ੍ਰੈਸ਼ਨ (Valid expression) ਬਨਾਉਣ ਲਈ ਆਪਰੇਟਰਾਂ ਅਤੇ ਆਪਰੇਂਡਾਂ ਦੇ ਸੁਮੇਲ ਨੂੰ ਪਾਈਥਨ ਭਾਸ਼ਾ ਦੇ ਸਿੰਟੈਕਸ ਨਿਯਮਾਂ ਅਨੁਸਾਰ ਲਿਖਣਾ ਜ਼ਰੂਰੀ ਹੁੰਦਾ ਹੈ। ਉਪਰੋਕਤ ਉਦਾਹਰਣ ਵਿੱਚ ਐਕਸਪ੍ਰੈਸਨ ਨੂੰ ਪ੍ਰਮਾਣਿਕ ਬਨਾਉਣ ਲਈ ਵੇਰੀਏਬਲ z ਦੇ ਮੁੱਲ ਨੂੰ ਸਟੋਰ ਕਰਵਾਉਣ ਲਈ = ਓਪਰੇਟਰ ਦੇ ਖੱਬੇ ਪਾਸੇ ਇੱਕ ਪ੍ਰਮਾਣਿਕ ਮੈਮਰੀ ਲੋਕੇਸ਼ਨ (ਆਈਡੈਂਟੀਫਾਇਰ) ਦਾ ਹੋਣਾ ਜ਼ਰੂਰੀ ਹੈ।

ਜਦੋਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿੱਚ ਇੱਕ ਤੋਂ ਵੱਧ ਆਪਰੇਟਰ ਮੌਜੂਦ ਹੁੰਦੇ ਹਨ, ਤਾਂ ਇਸਦਾ ਮੁਲਾਂਕਣ (Evaluation) ਆਪਰੇਟਰਾਂ ਦੀ ਤਰਜੀਹ/ਪ੍ਰੈਸੀਡੈਂਸ (Hierarchy or Precedence) ਅਨੁਸਾਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

3.3.1 ਆਪਰੇਟਰ ਪ੍ਰੈਸੀਡੇਂਸ (Operators Precedence):

ਆਪਰੇਟਰਾਂ ਨੂੰ ਉਹਨਾਂ ਦੇ ਮੁਲਾਂਕਣ ਕ੍ਰਮ ਅਨੁਸਾਰ ਲੜੀਵਾਰ ਰੂਪ ਵਿੱਚ (hierarchically) ਸਮੂਹਬੱਧ (grouped) ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਜਿਸਨੂੰ **ਪ੍ਰੈਸੀਡੇਂਸ (Precedence)** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਉੱਚ ਤਰਜੀਹ ਵਾਲੇ ਆਪਰੇਟਰ ਉਪਰ ਘੱਟ ਤਰਜੀਹ ਵਾਲੇ ਆਪਰੇਟਰਾਂ ਤੋਂ ਪਹਿਲਾਂ ਕੰਮ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਸਧਾਰਨ ਸ਼ਬਦਾਂ ਵਿੱਚ, ਆਪਰੇਟਰ ਪ੍ਰੈਸੀਡੈਂਸ ਉਸ ਕ੍ਰਮ ਦਾ ਵਰਣਨ ਕਰਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਇੱਕ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿੱਚਲੇ ਆਪਰੇਡਾਂ ਉੱਤੇ ਆਪਰੇਸ਼ਨ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਬਰੈਕਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੁਦਰਤੀ ਕ੍ਰਮ ਨੂੰ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿੱਚ ਇੱਕ ਸਮਾਨ ਤਰਜੀਹ ਵਾਲੇ ਆਪਰੇਟਰ ਵੀ ਮੌਜੂਦ ਹੋ ਸਕਦੇ ਹਨ। ਅਜਿਹੀ ਸਿਥਤੀ ਵਿੱਚ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿੱਚ ਇੱਕੋ ਤਰਜੀਹ ਵਾਲੇ ਆਪਰੇਟਰਾਂ ਦਾ ਖੱਬੇ-ਤੋਂ-ਸੱਜੇ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਅੱਗੇ ਦਿੱਤਾ ਟੇਬਲ ਪਾਈਥਨ ਦੇ ਆਮ ਤੌਰ 'ਤੇ ਵਰਤੇ ਜਾਂਦੇ ਆਪਰੇਟਰਾਂ ਦੀ ਉੱਚ ਤਰਜੀਹ (High Precedence) ਤੋਂ ਘੱਟ ਤਰਜੀਹ (Low Precedence) ਤੱਕ ਦੀ ਸੂਚੀ ਦਿਖਾ ਰਿਹਾ ਹੈ:

ਆਪਰੇਟਰ ਅਤੇ ਵਿਆਖਿਆ (Operator & Description)	ਪ੍ਰੈਸੀਡੇਂਸ
( ) (Parenthesis)	ਉੱਚ ਤਰਜੀਹ
** (Exponentiation)	
+, - (Unary Plus and Minus, as a +ve or –ve Number)	
*, /, %, // (Multiply, Divide, Modulo and Floor Division)	
+, - (Addition and Subtraction)	
<=, <, >, >= (Comparison Operators)	
==, != (Equality Operators)	
=, %=, /=, //=, -=, +=, *=, **= (Assignment Operators)	
in, not in (Membership Operators)	
not, or, and (Logical Operators)	ਘੱਟ ਤਰਜੀਹ

ਵੱਧ ਤੋਂ ਘੱਟ ਤਰਜੀਹ ਵੱਲ

ਟੇਬਲ 3.6: ਆਪਰੇਟਰਾਂ ਦੀ ਪ੍ਰੈਸੀਡੇਂਸ

ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿੱਚ ਓਪਰੇਟਰਾਂ ਦੇ ਮੁਲਾਂਕਣ ਦੇ ਕੁਦਰਤੀ ਕ੍ਰਮ ਨੂੰ ਬਰੈਕਟਸ ( ) ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿਚ ( ) ਦੇ ਅੰਦਰ ਲਿਖੇ ਆਪਰੇਟਰਜ਼ ਅਤੇ ਆਪਰੈਂਡਜ਼ ਦਾ ਮੁਲਾਂਕਣ ਪਹਿਲਾਂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਬਰਾਬਰ ਤਰਜੀਹ ਵਾਲੇ ਓਪਰੇਟਰਾਂ ਲਈ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿਚ ਖੱਬੇ ਤੋਂ ਸੱਜੇ ਵੱਲ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

ਹੇਠਾਂ ਇੱਕ ਉਦਾਹਰਨ ਦਿਤੀ ਗਈ ਹੈ ਜੋ ਇਹ ਦਰਸਾ ਰਹੀ ਹੈ ਕਿ ਕਿਵੇਂ ਅਰਿੱਥਮੈਟਿਕ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿਚ ਆਪਰੇਟਰਾਂ ਦੀ ਤਰਜੀਹ/ਪ੍ਰੈਸੀਡੇਂਸ ਅਨੁਸਾਰ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ:

```
a = 2 * 2 ** 3 / 2 + 5;
```

ਇਸ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿੱਚ ਐਕਸਪੋਨੈਂਟ (**) ਆਪਰੇਟਰ ਦੀ ਤਰਜੀਹ ਸਭ ਤੋਂ ਵੱਧ ਹੈ, ਇਸਲਈ ਪਹਿਲਾਂ ਇਸ ਆਪਰੇਟਰ (2 ** 3) ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਵੇਗਾ। ਇਸ ਉਪਰ ਕੰਮ ਕਰਨ ਤੋਂ ਬਅਦ ਐਕਸਪ੍ਰੈਸ਼ਨ ਇਹ ਹੋਵੇਗੀ:

```
a = 2 * 8 / 2 + 5;          (** ਆਪਰੇਟਰ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ)
```

ਹੁਣ ਪ੍ਰੈਸੀਡੈਂਸ ਅਨੁਸਾਰ ਗੁਣਾ (*) ਅਤੇ ਭਾਗ (/) ਆਪਰੇਟਰ ਆਪਣਾ ਕੰਮ ਕਰਨਗੇ। ਪਰ ਇਨ੍ਹਾਂ ਦੋਹਾਂ ਆਪਰੇਟਰਾਂ ਦੀ ਤਰਜੀਹ ਇੱਕੋ ਜਿਹੀ ਹੈ। ਜੇਕਰ ਆਪਰੇਟਰਾਂ ਦੀ ਤਰਜੀਹ ਇੱਕੋ ਜਿਹੀ ਹੋਵੇ, ਤਾਂ ਉਹਨਾਂ ਦਾ ਖੱਬੇ-ਤੋਂ-ਸੱਜੇ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਲਈ ਭਾਗ (/) ਆਪਰੇਟਰ ਤੋਂ ਪਹਿਲਾਂ ਗੁਣਾ (*) ਆਪਰੇਟਰ ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਵੇਗਾ। ਹੁਣ ਐਕਸਪ੍ਰੈਸ਼ਨ ਇਸ ਤਰ੍ਹਾਂ ਬਣ ਜਾਵੇਗੀ:

```
a = 16 / 2 + 5;             (* ਆਪਰੇਟਰ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ)
```

ਹੁਣ, ਸਾਡੇ ਕੋਲ ਸਿਰਫ਼ ਤਿੰਨ ਆਪਰੇਟਰ ਬਚੇ ਹਨ: ਭਾਗ (/), ਜੋੜ (+) ਅਤੇ ਅਸਾਈਨਮੈਂਟ (=) ਆਪਰੇਟਰ। ਪ੍ਰੈਸੀਡੈਂਸ/ਤਰਜੀਹ ਅਨੁਸਾਰ, ਭਾਗ ਪਹਿਲਾਂ ਕੀਤਾ ਜਾਵੇਗਾ, ਫਿਰ ਐਕਸਪ੍ਰੈਸ਼ਨ ਦਾ ਅੰਤਿਮ ਮੁੱਲ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਜੋੜ (+) ਕੀਤਾ ਜਾਵੇਗਾ, ਭਾਵ:

```
a = 8 + 5;                  (/ ਆਪਰੇਟਰ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ)
a = 13;                     (+ ਆਪਰੇਟਰ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ)
```

ਐਕਸਪ੍ਰੈਸ਼ਨ ਦਾ ਇਹ ਅੰਤਿਮ ਮੁੱਲ (13) ਹੁਣ ਅਸਾਈਨਮੈਂਟ (=) ਆਪਰੇਟਰ ਦੀ ਸਭ ਤੋਂ ਘੱਟ ਤਰਜੀਹ/ਪ੍ਰੈਸੀਡੇਂਸ ਕਾਰਨ ਵੇਰੀਏਬਰ 'a' ਵਿੱਚ ਸਟੋਰ ਕੀਤਾ ਜਾਵੇਗਾ।

ਆਉ ਹੁਣ ਆਪਰੇਟਰਾਂ ਦੀ ਪ੍ਰੈਸੀਡੇਂਸ ਦੀ ਬਿਹਤਰ ਸਮਝ ਲਈ ਕੁੱਝ ਹੋਰ ਉਦਾਹਰਣਾਂ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ:

ਉਦਾਹਰਣ: 1 ਐਕਸਪ੍ਰੈਸ਼ਨ: 20 - 10 * 40

ਹੱਲ:	20 - 10 * 40	#Step 1 (ਅਸਲ ਐਕਸਪ੍ਰੈਸ਼ਨ)
	20 - 400	#Step 2 (ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)
	-380	#Step 3 (ਅੰਤਿਮ ਨਤੀਜੇ ਲਈ 20-400 ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)

ਉਦਾਹਰਣ: 2 ਐਕਸਪ੍ਰੈਸ਼ਨ: (20 – 10) * 40

ਹੱਲ:	(20 – 10) * 40	#Step 1 (ਅਸਲ ਐਕਸਪ੍ਰੈਸ਼ਨ)
	10 * 40	#Step 2 ((20-10) ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)
	400	#Step 3 (ਅੰਤਿਮ ਨਤੀਜੇ ਲਈ 1040 ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)

ਉਦਾਹਰਣ: 3 ਐਕਸਪ੍ਰੈਸ਼ਨ 17.0 / 4 + (6 + 3.0)

ਹੱਲ:	17.0 / 4 + (6 + 3.0)	#Step 1 (ਅਸਲ ਐਕਸਪ੍ਰੈਸ਼ਨ)
	17.0 / 4 + 9.0	#Step 2 (ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)
	4.25 + 9.0	#Step 3 (ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)
	13.25	#Step 4 (ਅੰਤਿਮ ਨਤੀਜੇ ਲਈ 4.25+9.0 ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਗਿਆ)

TEST YOURSELF 3.9

ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਐਕਸਪ੍ਰੈਸ਼ਨਜ਼ ਦੀ ਆਊਟਪੁੱਟ ਲਿਖੋ:

ੳ) `print(4.00/(1.0+ 3.0))` _____

ਅ) `print(4.00/1.0+ 3.0)` _____

ੲ) `print(50 + 2 * 3 ** 2 != 9//2 - 5)` _____

ਸ) `print(25%10+5< 50 and 39 <= 39)` _____

ਹ) `print(not(10+5<50) and (39<=39))` _____

3.4 ਟਾਈਪ ਕਨਵਰਜ਼ਨ (TYPE CONVERSION)

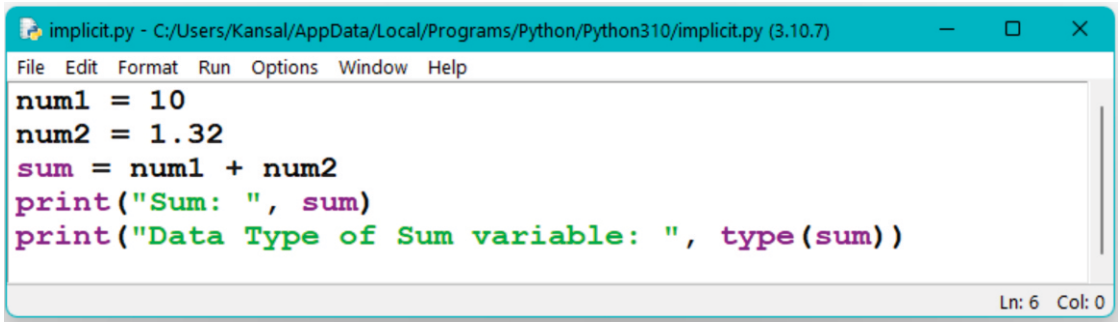
ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਿੱਚ ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਇੱਕ ਕਿਸਮ ਦੇ ਡਾਟਾ ਨੂੰ ਦੂਜੀ ਕਿਸਮ ਵਿੱਚ ਬਦਲਣ ਦੀ ਪ੍ਰਕਿਰਿਆ ਹੁੰਦੀ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਇੰਟੀਜ਼ਰ ਮੁੱਲ ਨੂੰ ਸਟ੍ਰਿੰਗ ਵਿੱਚ ਕਨਵਰਟ (ਰੂਪਾਂਤਰਨ) ਕਰਨਾ, ਫਲੋਟ ਮੁੱਲ ਨੂੰ ਇੰਟੀਜ਼ਰ ਮੁੱਲ ਵਿੱਚ ਤਬਦੀਲ ਕਰਨਾ ਆਦਿ। ਐਕਸਪ੍ਰੈਸ਼ਨ ਦਾ ਅੰਤਿਮ ਮੁੱਲ ਕੈਲਕੁਲੇਟ ਹੋਣ ਤੋਂ ਪਹਿਲਾਂ ਵੱਖੋ-ਵੱਖਰੇ ਕਿਸਮਾਂ ਦੇ ਆਪਰੇਡਾਂ ਵਿਚਕਾਰ ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਦੇ ਦੋ ਤਰੀਕੇ ਹਨ:

1. ਇੰਪਲੀਸਿੱਟ (ਆਟੋਮੈਟਿੱਕ) ਕਨਵਰਜ਼ਨ (Implicit (Automatic) Conversion)
2. ਐਕਸਪਲੀਸਿੱਟ (ਟਾਈਪ ਕਾਸਟਿੰਗ) ਕਨਵਰਜ਼ਨ (Explicit (Type Casting) Conversion)

3.4.1 ਇੰਪਲੀਸਿੱਟ ਕਨਵਰਜ਼ਨ (Implicit Conversion):

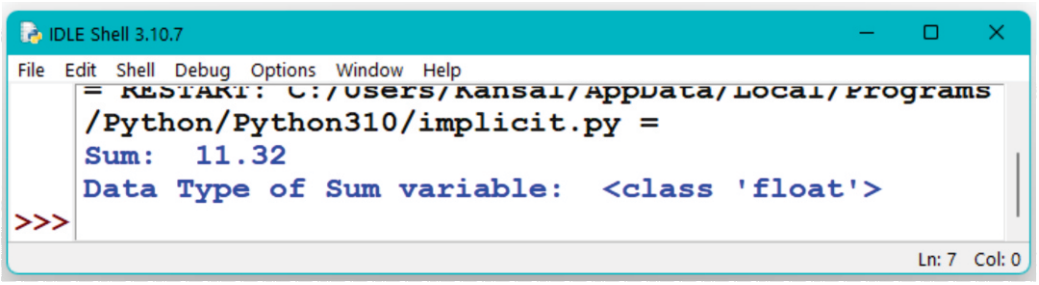
ਕੁੱਝ ਸਥਿਤੀਆਂ ਵਿੱਚ ਪਾਈਥਨ ਆਪਣੇ ਆਪ ਇੱਕ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਦੂਜੀ ਟਾਈਪ ਵਿੱਚ ਬਦਲਦਾ ਹੈ, ਇਸ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਨੂੰ ਇੰਪਲੀਸਿੱਟ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਕਨਵਰਜ਼ਨ ਪ੍ਰੋਸੈਸ ਨੂੰ ਪਾਈਥਨ ਦੁਆਰਾ ਆਪਣੇ ਆਪ ਹੈਂਡਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਇਸ ਲਈ ਇਸ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਨੂੰ ਆਟੋਮੈਟਿਕ ਕਨਵਰਜ਼ਨ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਸਾਨੂੰ ਇਸ ਕਿਸਮ ਦੇ ਕਨਵਰਜ਼ਨ ਵਿੱਚ ਸਪਸ਼ਟ ਤੌਰ 'ਤੇ (explicity) ਕਿਸੇ ਮੁੱਲ ਨੂੰ ਕਿਸੇ ਹੋਰ ਡਾਟਾ ਟਾਈਪ

ਵਿੱਚ ਤਬਦੀਲ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਪੈਂਦੀ। ਇਸ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਕੰਪਾਈਲੇਸ਼ਨ ਦੌਰਾਨ ਜਾਂ ਰਨ ਟਾਈਮ ਦੌਰਾਨ ਹੁੰਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਨ ਵਿਚ ਡਾਟਾ ਦੇ ਨੁਕਸਾਨ (loss of data) ਤੋਂ ਬਚਣ ਲਈ ਹੇਠਲੀ (lower) ਡਾਟਾ ਟਾਈਪ (integer) ਦੇ ਮੁੱਲ ਨੂੰ ਉਪਰਲੀ (higher) ਡਾਟਾ ਟਾਈਪ (float) ਦੇ ਮੁੱਲ ਵਿਚ ਆਟੋਮੈਟੀਕਲੀ ਤਬਦੀਲ ਕੀਤਾ ਜਾ ਰਿਹਾ ਹੈ:



ਚਿੱਤਰ 3.10: ਇੰਪਲੀਸਿਟ ਕਨਵਰਜ਼ਨ ਲਈ ਪ੍ਰੋਗਰਾਮ

ਇਹ ਪ੍ਰੋਗਰਾਮ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ ਆਊਟਪੁੱਟ ਦਰਸਾਏਗਾ:



ਚਿੱਤਰ 3.11: ਪ੍ਰੋਗਰਾਮ (implicit.py) ਦੀ ਆਊਟਪੁੱਟ

ਉਪਰੋਕਤ ਉਦਾਹਰਨ ਵਿੱਚ ਅਸੀਂ ਅੰਕਾਂ ਦਾ ਜੋੜ ਕਰਨ ਲਈ ਦੋ ਵੇਰੀਏਬਲ ਲਏ ਹਨ: ਇਕ ਵੇਰੀਏਬਲ num1 ਇੰਟੀਜ਼ਰ ਕਿਸਮ ਦਾ ਹੈ ਅਤੇ ਦੂਸਰਾ ਵੇਰੀਏਬਲ num2 ਫਲੋਟ ਕਿਸਮ ਦਾ ਹੈ। ਇਸ ਤੋਂ ਬਾਅਦ ਅਸੀਂ sum ਨਾਮਕ ਇੱਕ ਹੋਰ ਵੇਰੀਏਬਲ ਡਿਕਲੇਅਰ ਕੀਤਾ ਹੈ ਅਤੇ ਇਸ ਵਿੱਚ ਦੋਵੇਂ ਵੇਰੀਏਬਲਾਂ ਦੇ ਜੋੜ ਨੂੰ ਸਟੋਰ ਕੀਤਾ ਹੈ। ਇਸ ਤੋਂ ਬਾਅਦ ਜਦੋਂ ਅਸੀਂ sum ਵੇਰੀਏਬਲ ਦੀ ਡਾਟਾ ਕਿਸਮ ਦੀ ਡਾਂਚ ਕੀਤੀ, ਅਸੀਂ ਵੇਖ ਸਕਦੇ ਹਾਂ ਕਿ ਪਾਈਥਨ ਕੰਪਾਈਲਰ ਨੇ sum ਵੇਰੀਏਬਲ ਦੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਆਪਣੇ ਆਪ ਫਲੋਟ ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਬਦਲ ਦਿੱਤਾ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਦੀ ਕਨਵਰਜ਼ਨ ਨੂੰ ਇੰਪਲੀਸਿੱਟ (ਅਪ੍ਰਤੱਖ) ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਡਾਟਾ ਦੇ ਨੁਕਸਾਨ (loss of data) ਤੋਂ ਬਚਣ ਲਈ sum ਵੇਰੀਏਬਲ ਨੂੰ ਇੰਟੀਜ਼ਰ ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਤਬਦੀਲ ਕਰਨ ਦੀ ਬਜਾਏ ਫਲੋਟ ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਤਬਦੀਲ ਕੀਤਾ ਗਿਆ ਸੀ। ਜੇਕਰ ਕੰਪਾਈਲਰ ਨੇ sum ਵੇਰੀਏਬਲ ਨੂੰ ਇੰਟੀਜ਼ਰ ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਤਬਦੀਲ ਕਰ ਦਿੱਤਾ ਹੁੰਦਾ, ਤਾਂ ਕੰਪਾਈਲਰ ਨੂੰ ਫਰੈਕਸ਼ਨਲ ਹਿੱਸੇ ਨੂੰ ਖਤਮ ਕਰਨਾ ਪੈਂਦਾ ਅਤੇ ਜਿਸਦੇ ਨਤੀਜੇ ਵਜੋਂ ਡਾਟਾ ਦਾ ਨੁਕਸਾਨ ਹੁੰਦਾ। ਇਸ ਲਈ ਪਾਈਥਨ ਹਮੇਸ਼ਾ ਡਾਟਾ ਦੇ ਨੁਕਸਾਨ ਨੂੰ ਰੋਕਣ ਲਈ ਛੋਟੇ (smaller) ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਵੱਡੇ ਆਕਾਰ ਵਾਲੇ (wider) ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਬਦਲਦਾ ਹੈ।

3.4.2 ਐਕਸਪਲੀਸਿੱਟ ਕਨਵਰਜ਼ਨ (Explicit Conversion):

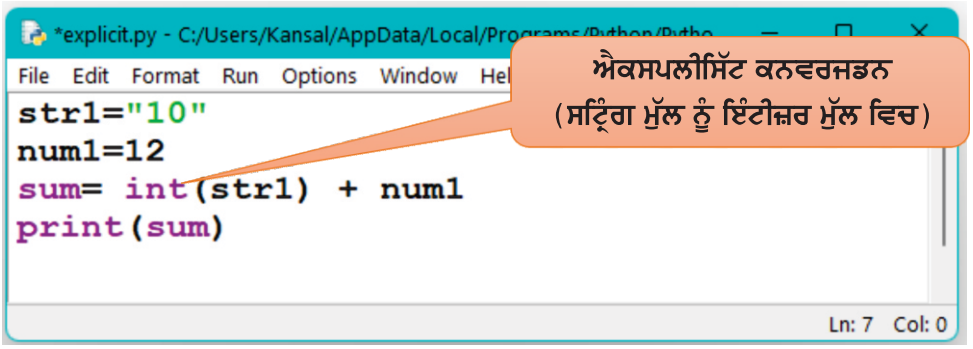
ਕੁੱਝ ਮਾਮਲਿਆਂ ਵਿੱਚ ਪਾਈਥਨ ਕਨਵਰਜ਼ਨ ਨੂੰ ਇੰਪਲੀਸਿੱਟ ਰੂਪ ਵਿੱਚ ਨਹੀਂ ਕਰ ਸਕਦਾ ਹੈ, ਅਜਿਹੀ ਸਥਿਤੀ ਵਿਚ ਐਕਸਪਲੀਸਿੱਟ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਮਹੱਤਵਪੂਰਣ ਰੋਲ ਅਦਾ ਕਰਦੀ ਹੈ। ਐਕਸਪਲੀਸਿੱਟ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਨੂੰ **ਟਾਈਪ ਕਾਸਟਿੰਗ** (Type Casting) ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਵਿਚ ਯੂਜ਼ਰ ਨੂੰ ਇੱਕ

ਆਬਜੈਕਟ/ਮੁੱਲ ਦੇ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਲੋੜੀਂਦੇ ਡਾਟਾ ਟਾਈਪ ਵਿੱਚ ਬਦਲਣ ਲਈ ਪਹਿਲਾਂ ਤੋਂ ਪਰਿਭਾਸ਼ਿਤ (predefiend) ਫੰਕਸ਼ਨਾਂ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਪੈਂਦੀ ਹੈ।

ਇਸ ਕਿਸਮ ਦੀ ਕਨਵਰਜ਼ਨ ਵਿੱਚ ਪ੍ਰੋਗਰਾਮਰ ਪਾਇਥਨ ਦੇ ਪਹਿਲਾਂ ਤੋਂ ਪਰਿਭਾਸ਼ਿਤ ਫੰਕਸ਼ਨਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਕਨਵਰਜ਼ਨ ਲਈ ਲੋੜੀਂਦੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਸਪਸ਼ਟ (explicit) ਰੂਪ ਵਿੱਚ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦਾ ਹੈ। ਇਸ ਕਿਸਮ ਦੇ ਕਨਵਰਜ਼ਨ ਵਿੱਚ ਡਾਟਾ ਦਾ ਨੁਕਸਾਨ (loss of data) ਹੋ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਅਸੀਂ ਆਬਜੈਕਟ/ਮੁੱਲ ਨੂੰ ਬਲਪੂਰਵਕ (forcefully) ਇਕ ਖਾਸ ਡਾਟਾ ਟਾਈਪ ਵਿਚ ਤਬਦੀਲ ਕਰਦੇ ਹਾਂ। ਹੇਠਾ ਕੁੱਝ ਇਨ-ਬਿਲਟ ਫੰਕਸ਼ਨਾਂ ਦਾ ਵਰਨਣ ਕੀਤਾ ਗਿਆ ਹੈ ਜੋ ਪਾਈਥਨ ਵਿੱਚ ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ:

- **int() :** ਇਹ ਫੰਕਸ਼ਨ ਕਿਸੇ ਵੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਇੰਟੀਜ਼ਰ ਰੂਪ ਵਿੱਚ ਬਦਲਦਾ ਹੈ।
- **float() :** ਇਹ ਫੰਕਸ਼ਨ ਕਿਸੇ ਵੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਫਲੋਟਿੰਗ-ਪੁਆਇੰਟ ਨੰਬਰ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
- **tuple() :** ਇਹ ਫੰਕਸ਼ਨ ਕਿਸੇ ਵੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਟਪਲ ਟਾਈਪ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
- **set() :** ਇਹ ਫੰਕਸ਼ਨ ਕਿਸੇ ਵੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਸੈੱਟ ਟਾਈਪ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
- **list() :** ਇਹ ਫੰਕਸ਼ਨ ਕਿਸੇ ਵੀ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਲਿਸਟ ਟਾਈਪ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
- **str() :** ਇਹ ਫੰਕਸ਼ਨ ਇੰਟੀਜ਼ਰ ਨੂੰ ਇੱਕ ਸਟ੍ਰਿੰਗ ਵਿੱਚ ਬਦਲਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
- **complex(real, image):** ਇਹ ਫੰਕਸ਼ਨ ਰੀਅਲ ਨੰਬਰਾਂ ਨੂੰ ਕੰਪਲੈਕਸ ਨੰਬਰ ਵਿੱਚ ਬਦਲਦਾ ਹੈ।

ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਨ ਵਿਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ ਕਿ ਕਿਸ ਤਰ੍ਹਾਂ ਐਕਸਪਲੀਸਿੱਟ ਡਾਟਾ ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਲਈ ਇਹਨਾਂ ਫੰਕਸ਼ਨਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ:



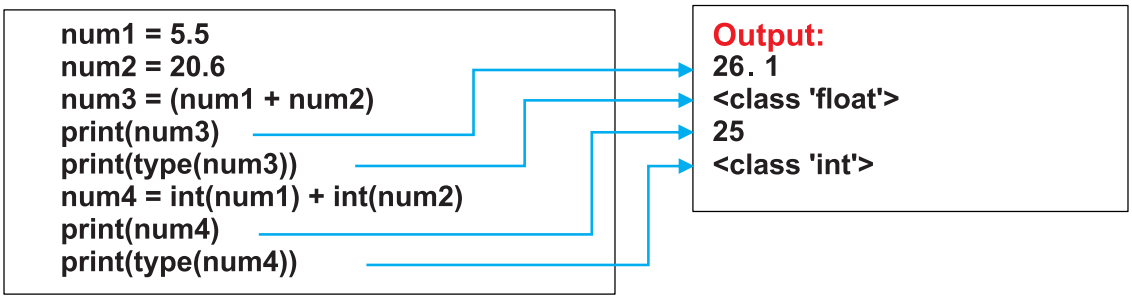
ਚਿੱਤਰ 3.12: ਐਕਸਪਲੀਸਿੱਟ ਕਨਵਰਜ਼ਨ ਲਈ ਪ੍ਰੋਗਰਾਮ (explicit.py)

ਇਹ ਪ੍ਰੋਗਰਾਮ ਹੇਠਾਂ ਦਿੱਤੀ ਆਊਟਪੁੱਟ ਦਿਖਾਵੇਗਾ:

```
22
```

ਦਿੱਤੀ ਗਈ ਉਦਾਹਰਨ ਵਿੱਚ ਵੇਰੀਏਬਲ str1 ਸਟਰਿੰਗ ਟਾਈਪ ਦਾ ਹੈ ਅਤੇ ਵੇਰੀਏਬਲ num1 ਇੰਟੀਜ਼ਰ ਟਾਈਪ ਦਾ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਬਿਨ੍ਹਾਂ ਕਿਸੇ ਕਨਵਰਜ਼ਨ ਦੇ ਇਹਨਾਂ ਦੋ ਮੁੱਲਾਂ (str1 + num1) ਨੂੰ ਜੋੜਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰਾਂਗੇ, ਤਾਂ ਪਾਈਥਨ TypeError ਮੈਸੇਜ਼ ਦਿਖਾਏਗਾ। ਇਸ ਲਈ ਇਸ ਕੰਮ ਨੂੰ ਕਰਨ ਲਈ ਸਾਨੂੰ ਐਕਸਪਲੀਸਿੱਟ ਟਾਈਪ ਕਾਸਟਿੰਗ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਪਵੇਗੀ। ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਉਪਰੋਕਤ ਉਦਾਹਰਣ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਵੇਖ ਸਕਦੇ ਹਾਂ, ਅਸੀਂ **int()** ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵੇਰੀਏਬਲ str1 ਨੂੰ ਇੰਟੀਜ਼ਰ ਕਿਸਮ ਵਿੱਚ ਬਦਲ ਦਿੱਤਾ ਹੈ ਅਤੇ ਫਿਰ num1 ਦੇ ਨਾਲ str1 ਦੇ ਕਨਵਰਟਡ ਮੁੱਲ (ਇੰਟੀਜ਼ਰ ਰੂਪ) ਨੂੰ ਜੋੜਿਆ ਗਿਆ ਹੈ। ਇਸ ਤੋਂ ਬਾਅਦ ਨਤੀਜਾ ਮੁੱਲ sum ਨਾਮਕ ਵੇਰੀਏਬਲ ਵਿੱਚ ਸਟੋਰ ਕੀਤਾ ਹੈ। ਜਦੋਂ ਇਸ ਨਤੀਜੇ ਨੂੰ ਪ੍ਰਿੰਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ 22 ਨੂੰ ਆਉਟਪੁੱਟ ਦੇ ਰੂਪ ਵਿੱਚ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ।

ਐਕਸਪਲੀਸਿਟ ਕਨਵਰਜ਼ਨ ਲਈ ਇਕ ਹੋਰ ਉਦਾਹਰਣ:



ਅਭਿਆਸ

ੳ. ਬਹੁਪਸੰਦੀ ਪ੍ਰਸ਼ਨ (Multiple Choice Questions:)

1. ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜਾ ਪ੍ਰੋਗਰਾਮ ਐਲੀਮੈਂਟ ਮੁੱਲ ਨੂੰ ਮੈਮਰੀ ਵਿਚ ਸਟੋਰ ਕਰਕੇ ਰੱਖਦਾ ਹੈ ?
   ੳ) ਵੇਰੀਏਬਲ — ਅ) ਕਮੈਂਟਸ
   ੲ) ਆਪਰੇਟਰਜ਼ — ਸ) ਉਪਰੋਕਤ ਸਾਰੇ
2. ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜੀ ਪਾਈਥਨ ਦੀ ਸਟੈਂਡਰਡ ਨੁਮੇਰਿੱਕ ਡਾਟਾ ਟਾਈਪ ਨਹੀਂ ਹੈ ?
   ੳ) ਇੰਟੀਜ਼ਰ — ਅ) ਫਲੋਟਿੰਗ
   ੲ) ਬੂਲੀਅਨ — ਸ) ਕੰਪਲੈਕਸ
3. ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜੀ ਟਾਈਪ ਪਾਈਥਨ ਦੀ ਮੈਪਿੰਗ ਟਾਈਪ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ?
   ੳ) ਲਿਸਟ — ਅ) ਡਿਕਸ਼ਨਰੀ
   ੲ) ਟਪਲ — ਸ) ਸੈੱਟ
4. ਪਾਈਥਨ ਵਿਚ _____ ਕੀਅਵਰਡ ਦੀ ਵਰਤੋਂ **null** ਮੁੱਲ ਜਾਂ **no value** ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ?
   ੳ) ਨਥਿੰਗ (Nothing) — ਅ) ਨੱਲ (Null)
   ੲ) ਜ਼ੀਰੋ (Zero) — ਸ) ਨਨ (None)
5. ਪਾਈਥਨ ਵਿਚ **True/False** ਮੁੱਲ ਨੂੰ ਦਰਸਾਉਣ ਲਈ _____ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
   ੳ) bool — ਅ) boolean
   ੲ) Boolean — ਸ) None
6. ਪਾਈਥਨ ਵਿਚ ਲਿਸਟ ਬਨਾਉਣ ਲਈ ਅਸੀਂ ਡਾਟਾ ਆਈਟਮਾਂ ਨੂੰ _____ ਵਿਚ ਲਿਖਦੇ ਹਾਂ।
   ੳ ਗੋਲ ਬਰੈਕਟਸ (Parenthesis) — ਅ) ਚਕੌਰ ਬਰੈਕੇਟਸ (Square Brackets)
   ੲ) ਘੁੰਡੀਦਾਰ ਬਰੈਕਟਸ (Curly Brackets) — ਸ) ਐਂਗੁਲਰ ਬਰੈਕਟਸ (Angular Brackets)
7. ਪਾਈਥਨ ਵਿਚ ਯੂਨੀਕੋਡ ਮੁੱਲਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ _____ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
   ੳ) ਸਟ੍ਰਿੰਗ — ਅ) ਟਪਲ
   ੲ) ਲਿਸਟ — ਸ) ਸੈੱਟ

8. _____ ਟਾਈਪਸ ਉਹ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਜਿਹਨਾਂ ਦੇ ਮੁੱਲ ਉਹਨਾਂ ਨੂੰ ਬਨਾਉਣ ਤੋਂ ਬਾਅਦ ਵੀ ਬਦਲੇ ਜਾ ਸਕਦੇ ਹਨ।

ੳ) ਮਿਊਟੇਬਲ (Mutable)　　ਅ) ਇਮਮਿਊਟੇਬਲ (Immutable)

ੲ) ਨਨ (None)　　ਸ) ਮੈਪਿੰਗ (Mapping)

9. ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜਾ ਪਾਈਥਨ ਦਾ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰ ਨਹੀਂ ਹੈ ?

ੳ) /　　ਅ) //

ੲ) **　　ਸ) ++

10. ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜਾ ਫੰਕਸ਼ਨ ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ?

ੳ) int()　　ਅ) float()

ੲ) str()　　ਸ) ਉਪਰੋਕਤ ਸਾਰੇ

ਅ. ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਲਈ **True** ਜਾਂ **False** ਲਿਖੋ।

1. ਚਿੰਨ੍ਹ % ਮਾਡਿਊਲਸ ਆਪਰੇਟਰ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਜੋ ਦੋ ਸੰਖਿਆਵਾਂ ਦੀ ਵੰਡ ਤੋਂ ਬਾਅਦ ਬਾਕੀ ਬਚੇ ਮੁੱਲ ਨੂੰ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
2. ਰਿਲੇਸ਼ਨਲ ਓਪਰੇਟਰ ਤੁਲਨਾ ਕਰਨ ਤੋਂ ਬਾਅਦ True ਜਾਂ False ਮੁੱਲ ਵਾਪਸ ਕਰਦੇ ਹਨ।
3. ਓਪਰੇਟਰ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਹੁੰਦੇ ਹਨ ਜੋ ਓਪਰੇਡਾਂ 'ਤੇ ਸਧਾਰਣ ਕੰਮ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।
4. ਚਕੋਰ ਬਰੈਕਟਾਂ ਵਿੱਚ ਕੌਮਿਆਂ ਨਾਲ ਵੱਖ ਕੀਤੇ ਮੁੱਲਾਂ ਦੇ ਇੱਕ ਕ੍ਰਮਬੱਧ ਸੰਗ੍ਰਹਿ ਨੂੰ ਸੈੱਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
5. ਸਿਕੁਐਂਸ (Sequence) ਇੱਕ ਸਮਾਨ ਜਾਂ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੀਆਂ ਡਾਟਾ-ਆਈਟਮਾਂ ਦਾ ਕ੍ਰਮਬੱਧ ਸੰਗ੍ਰਹਿ ਹੁੰਦਾ ਹੈ।

ੲ. ਛੋਟੇ ਉਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

1. ਫਲੋਟਿੰਗ ਪੁਆਇੰਟ ਨੰਬਰ ਕੀ ਹੁੰਦੇ ਹਨ। ਉਦਾਹਰਣਾਂ ਦਿਓ।
2. ਤੁਸੀਂ ਪਾਈਥਨ ਦੀ ਬੁਲੀਅਨ (Boolean) ਡਾਟਾ ਟਾਈਪ ਬਾਰੇ ਕੀ ਜਾਣਦੇ ਹੋ ?
3. ਪਾਈਥਨ ਵਿੱਚ ਸਿਕੁਐਂਸ (Sequence) ਡਾਟਾ ਟਾਈਪਸ ਕੀ ਹਨ ? ਉਹਨਾਂ ਦੇ ਨਾਮ ਲਿਖੋ।
4. ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਆਪਰੇਂਡਸ (Operands) ਦੀ ਵਿਆਖਿਆ ਕਰੋ।
5. ਅਰਿੱਥਮੈਟਿਕ (Arithmetic) ਆਪਰੇਟਰ ਕੀ ਹਨ ? ਉਦਾਹਰਣਾਂ ਦਿਓ।
6. ਐਕਸਪ੍ਰੈਸ਼ਨ (Expression) ਕੀ ਹੁੰਦੀ ਹੈ ?

ਸ. ਵੱਡੇ ਉਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

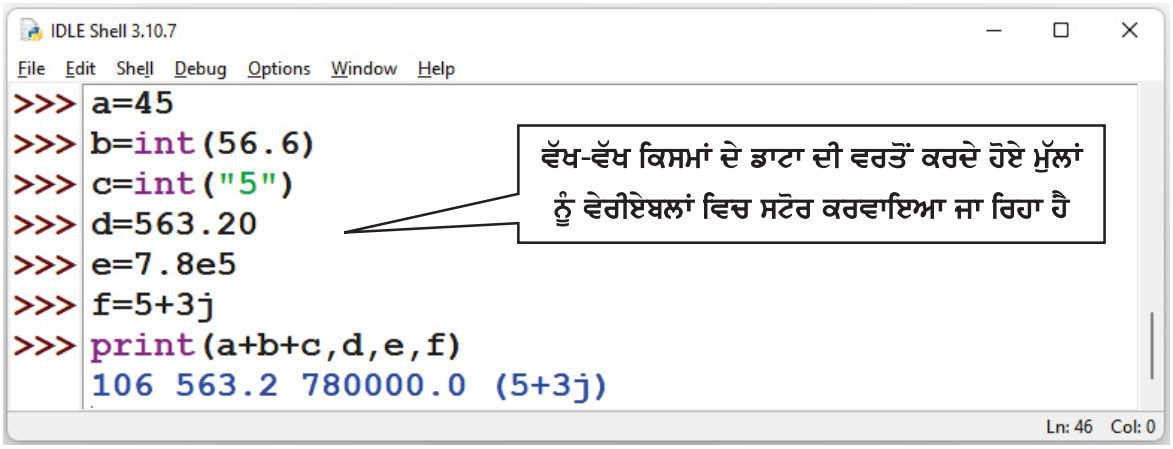
1. ਡਾਟਾ ਟਾਈਪਸ (Data Types) ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਭਾਵ ਹੈ ? ਪਾਈਥਨ ਵਿੱਚ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਆਮ ਸਟੈਂਡਰਡ ਡਾਟਾ-ਟਾਈਪਸ ਨੂੰ ਦਰਸਾਉਂਦਾ ਚਾਰਟ ਬਣਾਓ।
2. ਮਿਊਟੇਬਲ (Mutable) ਅਤੇ ਇਮਮਿਊਟੇਬਲ (Immutable) ਟਾਈਪਸ ਵਿੱਚ ਕੀ ਅੰਤਰ ਹੈ ?
3. ਆਪਰੇਟਰ ਕੀ ਹੁੰਦੇ ਹਨ ? ਪਾਈਥਨ ਵਿੱਚ ਵਰਤੇ ਜਾਂਦੇ ਕਿਸੇ ਦੋ ਤਰ੍ਹਾਂ ਦੇ ਆਪਰੇਟਰਾਂ ਦਾ ਵਰਨਣ ਕਰੋ।

ਲੈਬ ਐਕਟੀਵਿਟੀ

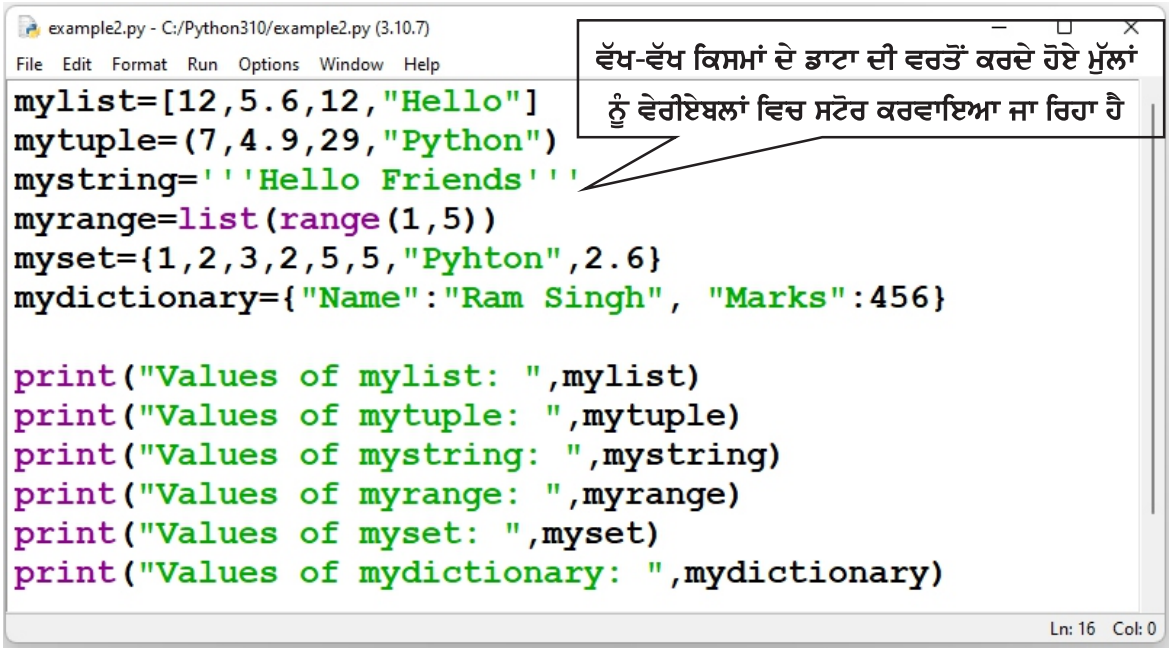
3.1 IDLE ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੇਠਾਂ ਦਰਸਾਏ ਗਏ ਨਿਰਦੇਸ਼ਾਂ ਨੂੰ ਇੰਟਰਐਕਟਿਵ/ਸਕ੍ਰਿਪਟ ਮੋਡ ਵਿੱਚ ਚਲਾਉਂਦੇ ਹੋਏ ਅਭਿਆਸ ਕਰੋ:

ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਉਦਾਹਰਣਾਂ ਪਾਈਥਨ ਵਿੱਚ ਵੱਖ ਵੱਖ ਡਾਟਾ ਟਾਈਪਸ ਨਾਲ ਕੰਮ ਕਰਨ ਸਬੰਧੀ ਪ੍ਰੈਕਟਿਸ ਲਈ ਦਿਤੀਆਂ ਗਈਆਂ ਹਨ:

1. ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਮੁੱਲਾਂ ਵਾਲੇ ਵੇਰੀਏਬਲਾਂ ਨਾਲ ਕੰਮ ਕਰਨ ਲਈ ਇੰਟਰਐਕਟਿਵ ਮੋਡ ਵਿੱਚ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ ਕੋਡ ਲਿਖੋ:



2. ਇੱਕ ਸਕ੍ਰਿਪਟ ਫਾਈਲ ਬਣਾਓ ਜਿਸ ਵਿਚ ਵੱਖ-ਵੱਖ ਤਰ੍ਹਾਂ ਦੇ ਡਾਟਾ ਟਾਈਪਸ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਕੋਡ ਲਿਖੋ:



ਹੁਣ **F5** ਕੀਅ ਪ੍ਰੈਸ ਕਰੋ, ਇਹ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ ਆਊਟਪੁਟ ਦਿਖਾਵੇਗਾ:

```
Values of mylist:  [12, 5.6, 12, 'Hello']
Values of mytuple:  (7, 4.9, 29, 'Python')
Values of mystring:  Hello Friends
Values of myrange:  [1, 2, 3, 4]
Values of myset:  {1, 2, 3, 2.6, 5, 'Pyhton'}
Values of mydictionary:  {'Name': 'Ram Singh', 'Marks': 456}
```

ਹੇਠਾਂ ਪਾਈਥਨ ਦੇ ਵੱਖ-ਵੱਖ ਓਪਰੇਟਰਾਂ ਦੀ ਵਰਤੋਂ ਸਬੰਧੀ ਪ੍ਰੈਕਟਿਸ ਲਈ ਉਦਾਹਰਣ ਪ੍ਰੋਗਰਾਮ ਦਿਤੇ ਗਏ ਹਨ:

3. ਪਾਈਥਨ ਦੇ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰਟਰਾਂ ਦੀ ਵਰਤੋਂ ਸਬੰਧੀ ਪ੍ਰੋਗਰਾਮ

example3.py - C:/Python310/example3.py (3.10.7)

File Edit Format Run Options Window Help

```
#Program showing use of arithmetic operators
num1=int(input("Enter value of num1: "))
num2=int(input("Enter value of num2: "))
print("Addition Operator (+): ",num1+num2)
print("Subtraction Operator (-): ",num1-num2)
print("Multiplication Operator (*): ",num1*num2)
print("Real Division Operator (/): ",num1/num2)
print("Integer Division Operator (//): ",num1//num2)
print("Modulus Operator (%): ",num1%num2)
print("Exponent Operator (**): ",num1**num2)
```

Ln: 12 Col: 0

ਹੁਣ ਪ੍ਰੋਗਰਾਮ ਚਲਾਓ, ਇਹ ਹੇਠਾਂ ਦਿਤੇ ਅਨੁਸਾਰ ਆਉਟਪੁੱਟ ਦਿਖਾਵੇਗਾ:

```
Enter value of num1: 5
Enter value of num2: 3
Addition Operator (+):  8
Subtraction Operator (-):  2
Multiplication Operator (*):  15
Real Division Operator (/):  1.6666666666666667
Integer Division Operator (//):  1
Modulus Operator (%):  2
Exponent Operator (**):  125
```

4. ਪਾਈਥਨ ਵਿਚ ਅਸਾਈਨਮੈਂਟ ਸਟੈਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾਉਂਦਾ ਪ੍ਰੋਗਰਾਮ:

example4.py - C:/Python310/example4.py (3.10.7)

File Edit Format Run Options Window Help

```
#Different types of assignement statements
num1=int(input("Enter value of num1: "))
num2=int(input("Enter value of num2: "))
x, y = num2, num1+3      #Parallel Assigment
print("Value of x and y: ",x,y)
a = b = num1 // num2     #Multiple Assignment
print("Value of a and b: ",a,b)
num1 *= num2             #Augmented Assignment
num2 **= 2               #Augmented Assignment
print("Value of num1 and num2: ",num1,num2)
x = x + 10               #Self Assignment
print("Value of x: ",x)
```

Ln: 14 Col: 0

ਹੁਣ ਪ੍ਰੋਗਰਾਮ ਚਲਾਓ, ਇਹ ਹੇਠਾਂ ਦਿਤੇ ਅਨੁਸਾਰ ਆਉਟਪੁੱਟ ਦਿਖਾਵੇਗਾ:

```
Enter value of num1: 5
Enter value of num2: 3
Value of x and y:  3 8
Value of a and b:  1 1
Value of num1 and num2:  15 9
Value of x:  13
```

3.2 ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦੇ ਉੱਤਰ ਪਤਾ ਕਰੋ:

1. ਹੇਠਾਂ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜੀ ਸਟੇਟਮੈਂਟ ਸਹੀ ਨਹੀਂ ਹੈ ?

   ੳ) print("7" + "9")
   ਅ) print(int("seven"))
   ੲ) print(7+9)
   ਸ) print(7)

2. ਹੇਠਾਂ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜਾ ਅਰਿੱਥਮੈਟਿਕ ਐਕਸਪ੍ਰੈਸ਼ਨ ਸਹੀ ਨਹੀਂ ਹੈ ?

   ੳ) A = 3-1*2
   ਅ) Y += 1
   ੲ) 15 + 6 = Z
   ਸ) Ten= 5 * 4

3. ਹੇਠਾਂ ਦਰਸਾਏ ਵਿਚੋਂ ਕਿਹੜਾ ਟਾਈਪ ਕਨਵਰਜ਼ਨ ਦੀ ਸਹੀ ਉਦਾਹਰਣ ਹੈ ?

   ੳ) int (5.0+"0.1")
   ਅ) int("four")
   ੲ) int("77"+"five")
   ਸ) int(5.0+2.3)

4. ਹੇਠ ਲਿਖਿਆਂ ਵਿੱਚੋਂ ਕਿਹੜੀ ਐਕਸਪ੍ਰੈਸ਼ਨ 4 ਨਤੀਜਾ ਨਹੀਂ ਦਿੰਦਾ ?

   ੳ) 17 // 4
   ਅ) 13 % 9
   ੲ) 2 * * 2
   ਸ) 17/4

5. ਹੇਠ ਲਿਖਿਆਂ ਵਿੱਚੋਂ ਕਿਹੜਾ ਪਾਈਥਨ ਵਿੱਚ ਕਮੈਂਟਸ ਦਾ ਗਲਤ ਤਰੀਕਾ ਹੈ ?

   ੳ) """"This is a comment""""
   ਅ) #This is a comment
   ੲ) //this is a comment
   ਸ) ######This is a comment

6. ਹੇਠ ਲਿਖਿਆਂ ਵਿੱਚੋਂ ਕਿਹੜੇ ਮੁੱਲ ਪਾਈਥਨ ਦੀ ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ ਦੇ ਸਹੀ ਮੁੱਲਾਂ ਨੂੰ ਦਰਸਾਉਂਦੇ ਹਨ:

   ੳ) Yes/No
   ਅ) Agree/Disagree
   ੲ) True/False
   ਸ) Right/Wrong

7. ਜੇਕਰ a = 20 ਅਤੇ b = 20 ਹੋਵੇ, ਤਾਂ a + =b ਵੇਰੀਏਬਲ a ਨੂੰ _____ ਮੁੱਲ ਅਸਾਈਨ ਹੋਵੇਗਾ

   ੳ) 40
   ਅ) 30
   ੲ) 20
   ਸ) 10

8. _____ ਆਪਰੇਟਰ ਦੀ ਵਰਤੋਂ ਇਹ ਪਤਾ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ ਕੀ ਦੋ ਸੰਖਿਆਵਾਂ ਦੀ ਵੰਡ ਤੋਂ ਬਾਅਦ ਕਿੰਨ੍ਹਾਂ ਮੁੱਲ ਬਾਕੀ ਬਚਦਾ ਹੈ ?

   ੳ) /
   ਅ) +
   ੲ) %
   ਸ) //

3.3 ਪਾਈਥਨ ਵਿੱਚ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਚਲਾਓ ਅਤੇ ਆਉਟਪੁੱਟ ਲਿਖੋ:

ਸਟੇਟਮੈਂਟਸ — ਆਉਟਪੁੱਟ

ੳ)
```
print("Hello" + " Friends")
```
_____

ਅ)
```
print(15+7-3*2%4)
```
_____

ੲ)
```
a = 5.2                  #float Value
msg ="Good Morning"      #String Value
c = 4+3j                 #complex Value
d = 5%2.0                #modulus Operator
print(msg + "Friends")
print(msg, "Friends")
print(a, c, 5/2.0, 5//2.0, d)
```
_____

_____

_____

ਸ)
```
x, y = 12, 6
x, y = y, x + 2
print (x, y)
```
_____

ਹ)
```
p, q, r = 5, 2, 1
p **= q + r
r = '5' + '5'
print(p, q, r)
```
_____

ਕ)
```
result = float(10) + int(5.2)
print (result)
```
_____

ਖ)
```
a=7
b=4
exp1 = (a > b)
print("Result of Expression:" , exp1)
```
_____

3.4 ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਵਿੱਚ ਗਲਤੀਆਂ ਲੱਭੋ ਅਤੇ ਸਹੀ ਸਟੇਟਮੈਂਟ ਲਿਖੋ:

ਗਲਤੀਆਂ ਸਹਿਤ ਸਟੇਟਮੈਂਟਸ	ਸਹੀ ਸਟੇਟਮੈਂਟ
a) `print 7`	_____
b) `print(Hello)`	_____
c) `print("Hello" + Friends)`	_____
d) `a==10` `print("Value of a is " a)`	_____

3.5 ਸਕ੍ਰਿਪਟ ਮੋਡ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੇਟਮੈਂਟਾਂ ਲਈ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ :

ੳ) ਦੋ ਪੂਰਨ ਅੰਕਾਂ (Integers) ਨੂੰ ਦਾਖਲ ਕਰਨ ਲਈ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਅਤੇ ਉਹਨਾਂ ਉੱਪਰ ਸਾਰੇ ਅੰਕਗਣਿਤਕ ਆਪਰੇਸ਼ਨ ਲਾਗੂ ਕਰੋ।

ਅ) ਦੋ ਨੰਬਰਾਂ ਨੂੰ ਸਵੈਪ (Swap) ਕਰਨ ਲਈ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ। (Hint: ਦੋ ਵੇਰੀਏਬਲ ਦੇ ਮੁੱਲਾਂ ਦੀ ਅਦਲਾ-ਬਦਲੀ ਕਰੋ)

ੲ) ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜੋ ਤਿੰਨ ਸੰਖਿਆਵਾਂ ਦੀ ਔਸਤ (Average) ਲੱਭਣ ਲਈ ਤਿੰਨ ਮੁੱਲਾਂ ਦੀ ਮੰਗ ਕਰਦਾ ਹੋਵੇ। (Hint: (a+b+c)/3)

ਸ) ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜੋ ਰੇਡੀਅਸ (r) ਦਾ ਮੁੱਲ ਇਨਪੁੱਟ ਕਰਵਾਉਣ ਉਪਰੰਤ ਚੱਕਰ ਦਾ ਖੇਤਰਫਲ ਪਤਾ ਕਰੇ। (Hint: area = pr^2)

ਹ) ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜੋ ਤੁਹਾਡੇ 10ਵੀਂ ਜਮਾਤ ਦੇ ਨਤੀਜੇ ਦੇ ਵੱਖ-ਵੱਖ ਵਿਸ਼ਿਆਂ ਦੇ ਵਿਅਕਤੀਗਤ ਅੰਕਾਂ ਨੂੰ ਇਨਪੁੱਟ ਕਰਨ ਉਪਰੰਤ ਪ੍ਰਤੀਸ਼ਤ ਅੰਕਾਂ ਦੀ ਗਣਨਾ ਕਰੇ।
(Hint: Percent_Marks = Obt_Marks/Total_Marks*100)

ਕ) ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜੋ ਡਿਗਰੀ ਸੈਲਸੀਅਸ (Celsius) ਵਿੱਚ ਤਾਪਮਾਨ ਦੇ ਮੁੱਲ ਨੂੰ ਫਾਰਮੂਲੇ: F = C*9/5+32 ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਉਸਨੂੰ ਡਿਗਰੀ ਫਾਰਨਹੀਟ (Fahrenheit) ਵਿੱਚ ਬਦਲੇ (ਇੱਥੇ, C ਸੈਲਸੀਅਸ ਵਿੱਚ ਤਾਪਮਾਨ ਹੈ ਅਤੇ F ਫਾਰਨਹੀਟ ਵਿੱਚ ਤਾਪਮਾਨ ਹੈ)।

ਖ) ਵੇਚੀਆਂ ਗਈਆਂ ਵਸਤੂਆਂ (Sold Items) ਦੀ ਕੁੱਲ ਕੀਮਤ 'ਤੇ 10% ਛੋਟ (Discount) ਦੀ ਗਣਨਾ ਕਰਨ ਲਈ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ। (Hint: discount = total_price *10/100)

ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ

ਅੱਜ ਦੇ ਸਮੇਂ ਪਾਈਥਨ ਦੀ ਮੰਗ ਬਹੁਤ ਜ਼ਿਆਦਾ ਹੈ। ਸਾਰੀਆਂ ਵੱਡੀਆਂ ਕੰਪਨੀਆਂ ਵਧੀਆ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਰਾਂ ਦੀ ਭਾਲ ਕਰ ਰਹੀਆਂ ਹਨ। ਡਾਟਾ ਸਾਇੰਸ, **AI** (ਆਰਟੀਫੀਸ਼ੀਅਲ ਇੰਟੈਲੀਜੈਂਸ), ਅਤੇ **ML** (ਮਸ਼ੀਨ ਲਰਨਿੰਗ) ਤਕਨਾਲੋਜੀਆਂ ਨਾਲ ਕੰਮ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮਰ ਪਾਈਥਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੈਬਸਾਈਟਾਂ, ਸਾਫਟਵੇਅਰ ਅਤੇ ਅਪਲੀਕੇਸ਼ਨਾਂ ਨੂੰ ਵਿਕਸਤ ਕਰ ਸਕਦੇ ਹਨ। ਪਾਈਥਨ ਨੂੰ ਲਗਾਤਾਰ ਵਿਸ਼ਵ ਦੀਆਂ ਸਭ ਤੋਂ ਪ੍ਰਸਿੱਧ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚੋਂ ਇੱਕ ਵਿਸ਼ੇਸ਼ ਦਰਜਾ ਮਿਲਦਾ ਆ ਰਿਹਾ ਹੈ।

ਪਾਠ ਦਾ ਉਦੇਸ਼

- ✓ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ
- ✓ ਇੰਡੈਂਟੇਸ਼ਨ ਦੀ ਮਹੱਤਤਾ
- ✓ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਦੀਆਂ ਕਿਸਮਾਂ
- ✓ ਸਿਕੁਐਂਸ਼ਿਅਲ ਫਲੋਅ ਕੰਟਰੋਲ
- ✓ ਕੰਡੀਸ਼ਨਲ ਫਲੋਅ ਕੰਟਰੋਲ: **if** ਸਟੇਟਮੈਂਟ
- ✓ ਲੂਪਿੰਗ ਫਲੋਅ ਕੰਟਰੋਲ : **while** ਅਤੇ **for** ਲੂਪ
- ✓ ਲੂਪ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ: **continue, break** ਅਤੇ **pass**
- ✓ ਇਹਨਾਂ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਲਿਖਣਾ

ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ✓ ਪਾਈਥਨ ਵਿੱਚ ਪ੍ਰੋਗਰਾਮਜ਼ ਲਿਖਣਾ ਅਤੇ ਡੀਬੱਗ ਕਰਨਾ
- ✓ ਸਿਲੈਕਸ਼ਨ ਤੇ ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟਸ ਦੀ ਧਾਰਨਾ ਅਤੇ ਵਰਤੋਂ ਦੀ ਸਮਝ
- ✓ ਪਾਈਥਲ ਵਿੱਚ ਉਪਲਬਧ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਲੂਪਸ ਦੀ ਸਮਝ
- ✓ ਸਮੱਸਿਆ ਦਾ ਵਿਸ਼ਲੇਸ਼ਣ, ਫੈਸਲਾ ਕਰਨਾ ਅਤੇ ਕੰਡੀਸ਼ਨਾਂ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨਾ
- ✓ ਵੱਖ-ਵੱਖ ਸਟਰਕਚਰਜ਼ ਦੇ ਢੁਕਵੇਂ ਸੁਮੇਲ ਲਈ ਵਿਸ਼ਲੇਸ਼ਣ ਅਤੇ ਫੈਸਲਾ ਕਰਨਾ ਦੇ ਯੋਗ ਹੋਣਾ
- ✓ ਅਜਿਹੇ ਕੋਡ ਲਿਖਣਯੋਗ ਹੋਣਾ ਜਿਹਨਾਂ ਵਿੱਚ ਫੈਸਲੇ ਲੈਣ ਵਾਲੇ ਸਟਰਕਚਰਜ਼ ਦੀ ਲੜੀ ਅਤੇ ਨੇਸਟਿਡ ਫੈਸਲਿਆਂ ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੋਵੇ
- ✓ ਲੂਪਿੰਗ ਅਤੇ ਨੈਸਟਿਡ ਲੂਪਿੰਗ ਵਾਲੇ ਸਧਾਰਨ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨਾ

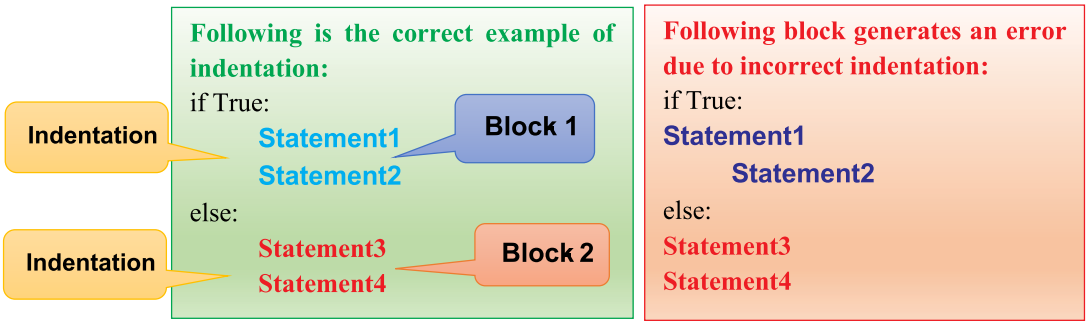
04 ਪਾਠ

ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ

ਪਿਛਲੇ ਪਾਠਾਂ ਵਿੱਚ ਅਸੀਂ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਬੁਨਿਆਦੀ ਧਾਰਨਾਵਾਂ ਜਿਵੇਂ ਕਿ: ਪਾਇਥਨ ਦੇ ਬਿਲਡਿੰਗ ਬਲਾਕਸ, ਵੇਰੀਏਬਲ, ਡਾਟਾ ਟਾਈਪਸ, ਆਪਰੇਟਰ ਆਦਿ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕੀਤੀ ਹੈ। ਹੁਣ ਤੱਕ ਬਣਾਏ ਗਏ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿਚ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਲਾਈਨ-ਦਰ-ਲਾਈਨ ਹੁੰਦਾ ਆਇਆ ਹੈ; ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਨੂੰ ਕ੍ਰਮਵਾਰ (Sequential) ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਅਸੀਂ ਕੰਟਰੋਲ ਫਲੋ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ (Execution flow) ਨੂੰ ਕੰਟਰੋਲ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਸ ਪਾਠ ਵਿੱਚ ਅਸੀਂ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਅਜਿਹੇ ਕੰਟਰੋਲ ਫਲੋਅ (Control flow) ਸਟੇਟਮੈਂਟਸ ਦੀ ਚਰਚਾ ਕਰਾਂਗੇ ਜੋ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਉਪਰ ਵਧੇਰੇ ਕੰਟਰੋਲ ਵਾਲੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਤਿਆਰ ਕਰਨ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੇ ਹਨ। ਪਰ ਇਹਨਾਂ ਕੰਟਰੋਲ ਫਲੋਅ ਸਟੇਟਮੈਂਟਾਂ 'ਤੇ ਚਰਚਾ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ, ਸਾਨੂੰ ਕੰਟਰੋਲ ਫਲੋਅ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਬਲਾਕਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਇੰਡੈਂਟੇਸ਼ਨ ਦੇ ਮਹੱਤਵ ਨੂੰ ਸਮਝਣਾ ਪਵੇਗਾ।

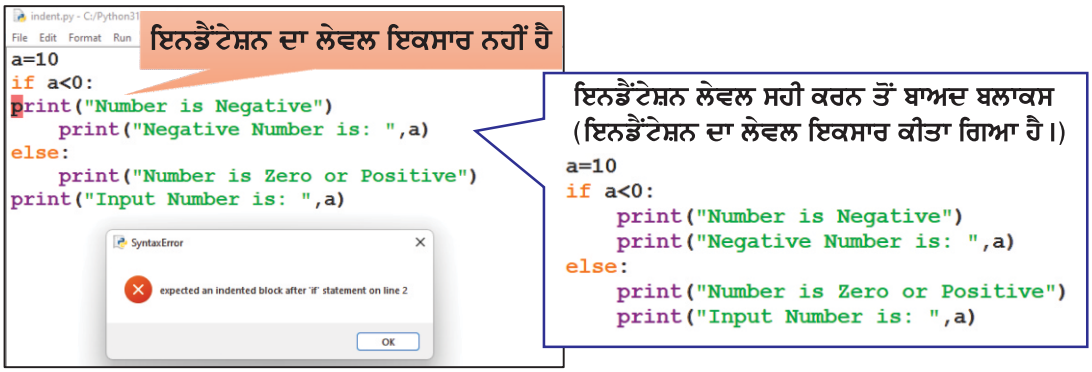
4.1 ਇੰਡੈਂਟੇਸ਼ਨ ਦਾ ਮਹੱਤਵ (IMPORTANCE OF INDENTATION)

ਪਾਈਥਨ ਵਿੱਚ ਇੰਡੈਂਟੇਸ਼ਨ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਹੈ। ਕੋਡ ਲਾਈਨ (Code Line) ਦੇ ਸ਼ੁਰੂ ਵਿੱਚ ਛੱਡੀਆਂ ਗਈਆਂ ਖਾਲੀ ਥਾਂਵਾਂ (Spaces) ਨੂੰ ਇੰਡੈਂਟੇਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਦੇ ਵੱਖ-ਵੱਖ ਤੱਤਾਂ (Elements), ਜਿਵੇਂ ਕਿ ਲੂਪਸ, ਫੰਕਸ਼ਨਾਂ ਅਤੇ ਕਲਾਸਾਂ ਦੇ ਬਲਾਕਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਇੰਡੈਂਟੇਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਪਾਈਥਨ ਕੋਡ ਵਿੱਚ ਇੰਡੈਂਟੇਸ਼ਨ ਦਾ ਇੱਕੋ ਲੇਵਲ (Same Level of Indentation) ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਇੱਕ ਸਿੰਗਲ ਬਲਾਕ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਹੋਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਅਕਸਰ ਇਸ ਉਦੇਸ਼ ਲਈ ਕਰਲੀ-ਬਰੈਕਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੀਆਂ ਹਨ। ਇੰਡੈਂਟੇਸ਼ਨ ਵਿੱਚ ਖਾਲੀ ਥਾਵਾਂ ਦੀ ਗਿਣਤੀ (Number of Spaces) ਵੇਰੀਏਬਲ ਹੁੰਦੀ ਹੈ, ਪਰ ਇੱਕ ਬਲਾਕ ਦੇ ਅੰਦਰ ਸਾਰੀਆਂ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਇਕੋ ਜਿੰਨ੍ਹੀ ਮਾਤਰਾ ਵਿੱਚ ਇੰਡੈਂਟ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਇੰਡੈਂਟੇਸ਼ਨ ਲਈ ਖਾਲੀ ਥਾਵਾਂ ਦੀ ਗਿਣਤੀ ਘੱਟੋ-ਘੱਟ ਇੱਕ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। ਉਦਾਹਰਣ ਲਈ:



ਚਿੱਤਰ 4.1: ਇੰਡੈਂਟੇਸ਼ਨ ਦੀ ਸਹੀ ਅਤੇ ਗਲਤ ਉਦਾਹਰਣ

ਸਾਨੂੰ ਕੋਡ ਦੇ ਇੱਕੋ ਬਲਾਕ ਵਿੱਚ ਇੱਕੋ ਜਿਨ੍ਹੀਆਂ ਖਾਲੀ ਥਾਵਾਂ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਪਵੇਗੀ, ਨਹੀਂ ਤਾਂ ਪਾਈਥਨ ਸਾਨੂੰ ਇੱਕ ਐਰਰ (Error) ਮੈਸੇਜ਼ ਦਿਖਾਵੇਗਾ। ਉਦਾਹਰਣ ਲਈ:



ਚਿੱਤਰ 4.2: ਇਨਡੈਂਟੇਸ਼ਨ ਲੇਵਲ ਇਕ ਸਾਰ ਨਾ ਹੋਣ ਕਾਰਨ ਸਿੰਟੈਕਸ ਐਰਰ

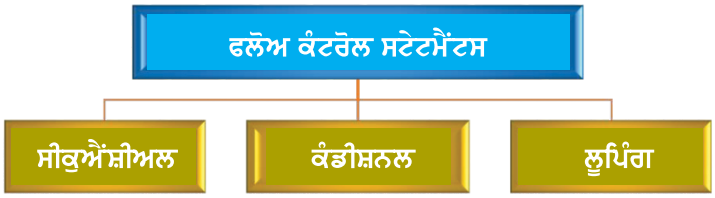
ਇਸ ਤਰ੍ਹਾਂ ਪਾਈਥਨ ਵਿੱਚ ਇੱਕੋ ਜਿਹੀ ਸਪੇਸ ਵਾਲੀਆਂ ਸਾਰੀਆਂ ਨਿਰੰਤਰ (Continuous) ਲਾਈਨਾਂ ਇੱਕ ਬਲਾਕ ਬਣਾਉਂਦੀਆਂ ਹਨ। ਇਸ ਸਮੇਂ ਤਰਕ ਨੂੰ ਸਮਝਣ ਦੀ ਕੋਸ਼ਿਸ਼ ਨਾ ਕਰੋ। ਬਸ ਇਹ ਯਕੀਨੀ ਬਣਾਓ ਕਿ ਤੁਸੀਂ ਕੋਡ ਦੇ ਬਲਾਕ ਨੂੰ ਬਨਾਉਣਾ ਸਮਝ ਲਿਆ ਹੈ, ਭਾਵੇਂ ਇਹ ਬਲਾਕ ਕਰਲੀ ਬਰੈਕਟਾਂ ਤੋਂ ਬਿਨਾਂ ਹੀ ਬਣਾਏ ਗਏ ਹਨ।

ਕੋਡ ਲਾਈਨ (Code Line) ਦੇ ਸ਼ੁਰੂ ਵਿੱਚ ਛੱਡੀਆਂ ਗਈਆਂ ਖਾਲੀ ਥਾਂਵਾਂ (Spaces) ਨੂੰ ਇੰਡੈਂਟੇਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਦੇ ਵੱਖ-ਵੱਖ ਤੱਤਾਂ (Elements), ਜਿਵੇਂ ਕਿ ਲੂਪਸ, ਫੰਕਸ਼ਨਾਂ ਅਤੇ ਕਲਾਸਾਂ ਦੇ ਬਲਾਕਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਇੰਡੈਂਟੇਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਹੋਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਅਕਸਰ ਇਸ ਉਦੇਸ਼ ਲਈ ਕਰਲੀ-ਬਰੈਕਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੀਆਂ ਹਨ।

4.2 ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ (FLOW CONTROL STATEMENTS)

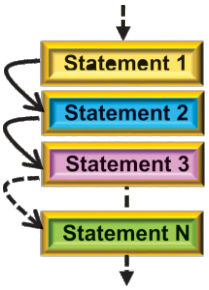
ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚ ਫਲੋਅ ਕੰਟਰੋਲ ਦਾ ਅਰਥ ਹੈ ਉਹ ਕ੍ਰਮ (Sequence) ਜਿਸ ਵਿੱਚ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਲਿਖੇ ਸਟੇਟਮੈਂਟਸ ਜਾਂ ਹਦਾਇਤਾਂ (Instructions) ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਸਰਲ ਸ਼ਬਦਾਂ ਵਿੱਚ ਅਸੀਂ ਕਹਿ ਸਕਦੇ ਹਾਂ ਕਿ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਲਾਗੂ ਕਰਨ ਦੇ ਕ੍ਰਮ ਨੂੰ ਕੰਟਰੋਲ ਦੇ ਪ੍ਰਵਾਹ (Flow of Control) ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।

ਪ੍ਰੋਗਰਾਮ ਦੇ ਫਲੋਅ (ਪ੍ਰਵਾਹ) ਨੂੰ ਸਮਝਣਾ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਹੈ। ਕਿਸੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਬਿਹਤਰ ਸਮਝ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਸਾਨੂੰ ਇਸ ਗੱਲ ਬਾਰੇ ਪਤਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਕਿ ਕਿਹੜੇ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਕਦੋਂ ਲਾਗੂ ਕੀਤਾ ਜਾਣਾ ਹੈ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਕਿਸ ਕ੍ਰਮ ਵਿੱਚ ਲਾਗੂ ਕੀਤਾ ਜਾਣਾ ਹੈ। ਕੰਟਰੋਲ ਸਟਰਕਚਰਜ਼ ਦੀ ਵਰਤੋਂ ਨਾਲ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਦੇ ਪ੍ਰਵਾਹ ਨੂੰ ਕੰਟਰੋਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਆਮ ਤੌਰ 'ਤੇ ਤਿੰਨ ਤਰੀਕਿਆਂ ਨਾਲ ਕਿਸੇ ਪ੍ਰੋਗਰਾਮ ਦੀਆਂ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ:

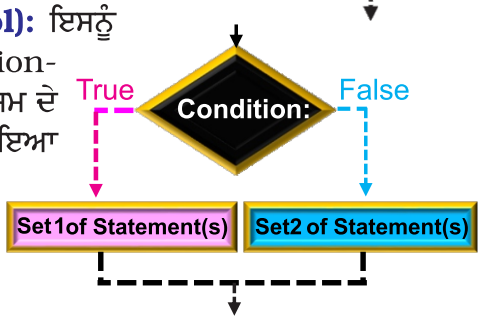


ਚਿੱਤਰ 4.3: ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਦੀਆਂ ਕਿਸਮਾਂ

1. **ਸੀਕੁਐਂਸ਼ੀਲ ਫਲੋਅ ਕੰਟਰੋਲ (Sequential Flow Control):** ਇਹ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਡਿਫਾਲਟ ਫਲੋਅ ਕੰਟਰੋਲ (Default Flow Control) ਹੈ। ਸੀਕੁਐਂਸ਼ੀਅਲ/ਕ੍ਰਮਵਾਰ ਸਟੇਟਮੈਂਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਅਸੀਂ ਸਧਾਰਨ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਤਿਆਰ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਸ ਕਿਸਮ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਵਿੱਚ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਇੱਕ ਤੋਂ ਬਾਅਦ ਇੱਕ ਕ੍ਰਮਵਾਰ (one after the other sequentially) ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਦਿੱਤੇ ਗਏ ਫਲੋਅ ਕੰਟਰੋਲ ਗ੍ਰਾਫ ਵਿੱਚ ਇਸ ਫਲੋਅ ਕੰਟਰੋਲ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਨੂੰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।

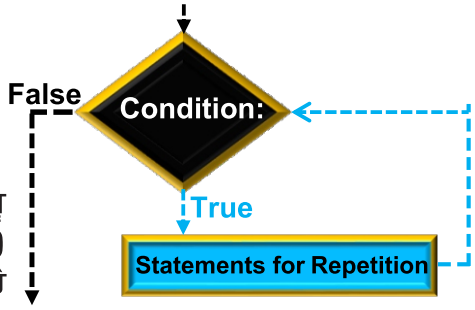


2. **ਕੰਡੀਸ਼ਨਲ ਫਲੋਅ ਕੰਟਰੋਲ (Conditional Flow Control):** ਇਸਨੂੰ ਬ੍ਰਾਂਚਿੰਗ (Branching) ਜਾਂ ਫੈਸਲਾ ਲੈਣ ਵਾਲੇ (Decision-Making) ਫਲੋਅ ਕੰਟਰੋਲ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਕਿਸਮ ਦੇ ਫਲੋਅ ਕੰਟਰੋਲ ਵਿੱਚ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਕੰਡੀਸ਼ਨ ਦੇ ਅਧਾਰ ਤੇ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਦਿੱਤੇ ਫਲੋਅ ਕੰਟਰੋਲ ਗ੍ਰਾਫ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ, ਜੇਕਰ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਦਾ ਸਹੀ ਮੁੱਲਾਂਕਣ (True evaluation) ਹੁੰਦਾ ਹੈ, ਤਾਂ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਇੱਕ ਸੈੱਟ ਨੂੰ ਚਲਾਇਆ ਜਾਵੇਗਾ ਅਤੇ ਜੇਕਰ ਇਹ ਗਲਤ (False) ਹੁੰਦਾ ਹੈ ਤਾਂ ਸਟੇਟਮੈਂਟਾਂ ਦਾ ਦੂਜਾ ਸੈੱਟ ਚਲਾਇਆ ਜਾਵੇਗਾ। ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟਾਂ ਦੀ ਵਰਤੋਂ ਗੁੰਜਲਦਾਰ (Complex) ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਬਹੁਤ ਜ਼ਿਆਦਾ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਦਿੱਤੇ ਗਏ ਫਲੋਅ ਕੰਟਰੋਲ ਗ੍ਰਾਫ ਵਿੱਚ ਇਸ ਫਲੋਅ ਕੰਟਰੋਲ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਨੂੰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।



3. ਲੂਪਿੰਗ ਫਲੋਅ ਕੰਟਰੋਲ (Looping Flow Control):

ਇਸਨੂੰ ਆਈਟ੍ਰੇਟਿਵ (Iterative) ਜਾਂ ਦੁਹਰਾਉਣ ਵਾਲੇ (Repetitive) ਫਲੋਅ ਕੰਟਰੋਲ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਸਦੀ ਵਰਤੋਂ ਸਟੇਟਮੈਂਟਾਂ ਦੀ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਨੂੰ ਦੁਹਰਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਉਸ ਸਮੇਂ ਤੱਕ ਵਾਰ-ਵਾਰ ਲਾਗੂ (Executes) ਕਰਦਾ ਹੈ ਜਦੋਂ ਤੱਕ ਕਿ ਕੰਡੀਸ਼ਨ ਗਲਤ (False) ਨਾਂ ਹੋ ਜਾਵੇ। ਇੱਕ ਵਾਰ ਜਦੋਂ ਕੰਡੀਸ਼ਨ ਗਲਤ (False) ਹੋ ਜਾਂਦੀ ਹੈ ਤਾਂ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਉਸ ਲੂਪ ਵਿੱਚੋਂ ਬਾਹਰ ਆ ਜਾਂਦੀ ਹੈ। ਦਿੱਤੇ ਗਏ ਫਲੋਅ ਕੰਟਰੋਲ ਗ੍ਰਾਫ ਵਿੱਚ ਇਸ ਫਲੋਅ ਕੰਟਰੋਲ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਨੂੰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।

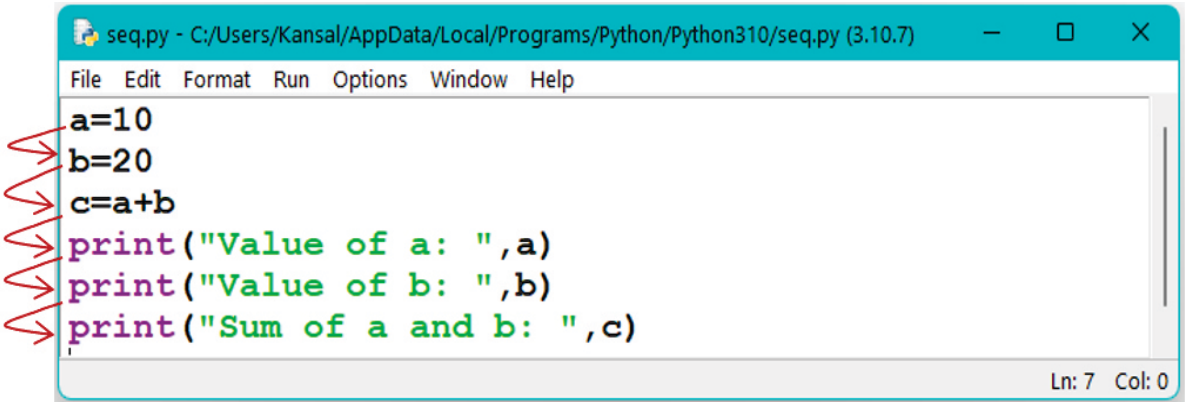


ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚ ਫਲੋਅ ਕੰਟਰੋਲ ਦਾ ਅਰਥ ਹੈ ਉਹ ਕ੍ਰਮ (Sequence) ਜਿਸ ਵਿੱਚ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਲਿਖੇ ਸਟੇਟਮੈਂਟਸ ਜਾਂ ਹਦਾਇਤਾਂ (Instructions) ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮਾਂ ਵਿਚ ਤਿੰਨ ਕਿਸਮਾਂ ਦਾ ਫਲੋਅ ਕੰਟਰੋਲ ਦੇਖਿਆ ਜਾ ਸਕਦਾ ਹੈ: ਸੀਕੁਐਂਸ਼ੀਅਲ, ਕੰਡੀਸ਼ਨਲ ਅਤੇ ਲੂਪਿੰਗ

ਆਉ ਹੁਣ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਇਹਨਾਂ ਫਲੋਅ ਕੰਟਰੋਲਸ ਬਾਰੇ ਵਧੇਰੇ ਵਿਸਥਾਰ ਨਾਲ ਚਰਚਾ ਕਰੀਏ:

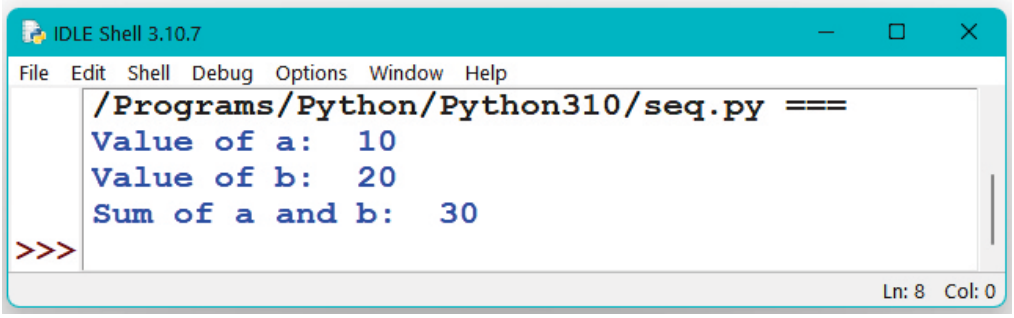
4.3 ਸੀਕੁਐਂਸ਼ਅਲ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ (Sequential Flow Control Statements)

ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਪਹਿਲਾਂ ਚਰਚਾ ਕੀਤੀ ਹੈ, ਕ੍ਰਮਵਾਰ (Sequential) ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਉੱਪਰ ਤੋਂ ਹੇਠਾਂ ਤੱਕ ਲਾਈਨ-ਦਰ-ਲਾਈਨ ਉਸੇ ਕ੍ਰਮ ਵਿੱਚ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ, ਜਿਸ ਕ੍ਰਮ ਵਿੱਚ ਉਹ ਦਿਖਾਈ ਦਿੰਦੀਆਂ ਹਨ। ਇਸਦੀ ਉਦਾਹਰਣ ਦਰਸਾਉਂਦਾ ਪ੍ਰੋਗਰਾਮ ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਹੈ:



ਪ੍ਰੋਗਰਾਮ (seq.py): ਸੀਕੁਐਂਸ਼ਅਲ ਐਗਜ਼ੀਕਿਊਸ਼ਨ (Sequential Execution) ਨੂੰ ਦਰਸਾਉਂਦਾ ਪ੍ਰੋਗਰਾਮ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਅਸੀਂ ਤਿੰਨ ਵੇਰੀਏਬਲ a, b ਅਤੇ c ਬਣਾਏ (create) ਅਤੇ ਇਨਸ਼ੀਅਲਾਈਜ਼ (Initialize) ਕੀਤੇ ਹਨ। ਵੇਰੀਏਬਲ a ਅਤੇ b ਨੂੰ ਕ੍ਰਮਵਾਰ ਕਾਂਸਟੈਂਟ ਮੁੱਲਾਂ 10 ਅਤੇ 20 ਨਾਲ ਇਨਸ਼ੀਅਲਾਈਜ਼ ਕੀਤਾ ਹੈ। ਉਸ ਤੋਂ ਬਾਅਦ ਵੇਰੀਏਬਲ a ਅਤੇ b ਦਾ ਜੋੜ ਵੇਰੀਏਬਲ c ਨੂੰ ਅਸਾਈਨ ਕੀਤਾ ਹੈ। ਇਹਨਾਂ ਤਿੰਨੇ ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਲਾਈਨ-ਦਰ-ਲਾਈਨ ਲਾਗੂ ਕੀਤਾ ਜਾਵੇਗਾ। ਜੋੜ ਕਰਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ print() ਫੰਕਸ਼ਨਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ a, b ਅਤੇ c ਵੇਰੀਏਬਲ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਢੁਕਵੇਂ ਸੰਦੇਸ਼ਾਂ ਨਾਲ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਹੈ। ਇਹਨਾਂ print() ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਵੀ ਕ੍ਰਮਵਾਰ (Sequentially) ਲਾਈਨ-ਦਰ-ਲਾਈਨ ਚਲਾਇਆ ਜਾਵੇਗਾ। ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਚਲਾਉਣ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਅੱਗੇ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਪ੍ਰਾਪਤ ਕਰਾਂਗੇ:



ਚਿੱਤਰ 4.4: ਪ੍ਰੋਗਰਾਮ (seq.py) ਦੀ ਆਊਟਪੁੱਟ

4.4 ਕੰਡੀਸ਼ਨਲ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ (Conditional Flow Control Statements)

ਇਹਨਾਂ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਬ੍ਰਾਂਚਿੰਗ (Branching) ਜਾਂ ਫੈਸਲੇ ਲੈਣ (Decision-Making) ਵਾਲੇ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਫਲੋਅ ਕੰਟਰੋਲਸ ਵਿੱਚ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਦੇ ਅਧਾਰ ਤੇ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ (Execution Flow) ਇਸ ਗੱਲ 'ਤੇ ਨਿਰਭਰ ਕਰਦਾ ਹੈ ਕਿ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਸਹੀ (TRUE) ਹੈ ਜਾਂ ਗਲਤ (FALSE) ਜੇਕਰ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ (TRUE) ਹੁੰਦਾ ਹੈ ਤਾਂ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਇੱਕ ਸੈੱਟ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕੀਤਾ ਜਾਵੇਗਾ, ਨਹੀਂ ਤਾਂ (ਜੇ ਦਿੱਤੀ ਕੰਡੀਸ਼ਨ ਗਲਤ (FALSE) ਹੁੰਦੀ ਹੈ) ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਦੂਜੇ ਸੈੱਟ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕੀਤਾ ਜਾਵੇਗਾ।

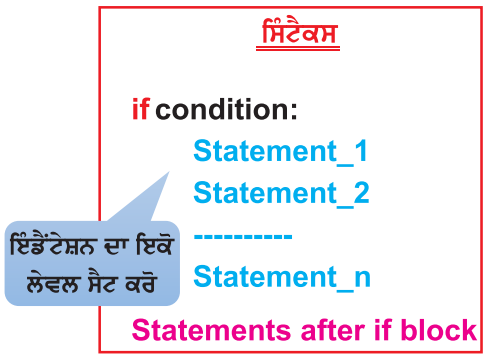
ਆਉ ਕੁੱਝ ਢੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਨਾਲ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਿਸਥਾਰ ਵਿੱਚ ਚਰਚਾ ਕਰੀਏ। ਪਾਈਥਨ ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਵਰਤੇ ਜਾਂਦੇ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟਸ ਦੀਆਂ ਕਿਸਮਾਂ ਹੇਠ ਲਿਖੀਆਂ ਹਨ:

ਚਿੱਤਰ 4.5: ਕੰਡੀਸ਼ਨਲ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਦੀਆਂ ਕਿਸਮਾਂ

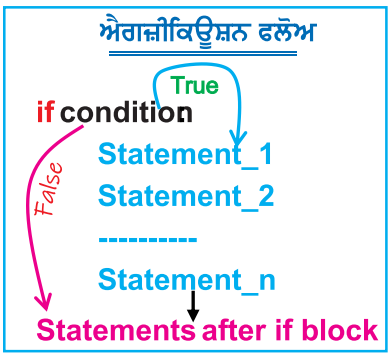
4.4.1 if ਸਟੇਟਮੈਂਟ (if statement):

ਇਹ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਇੱਕ ਵਿਸ਼ੇਸ਼ ਕੋਡ ਨੂੰ ਚਲਾਉਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦਾ ਹੈ, ਪਰ ਕੇਵਲ ਉਸ ਸਮੇਂ ਜਦੋਂ ਇੱਕ ਖਾਸ ਕੰਡੀਸ਼ਨ ਸਹੀ (True) ਹੁੰਦੀ ਹੈ। ਕੰਡੀਸ਼ਨ ਇੱਕ BOOLEAN ਕਿਸਮ ਦੇ ਮੁੱਲ ਵਿੱਚ ਨਤੀਜਾ ਦਿੰਦੀ ਹੈ – ਸਹੀ (True) ਜਾਂ ਗਲਤ (False)। ਜੇਕਰ ਕੰਡੀਸ਼ਨ ਦਾ ਨਤੀਜਾ ਸਹੀ (True) ਆਉਂਦਾ ਹੈ, ਤਾਂ "if ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਬਲਾਕ" ਨੂੰ ਚਲਾਇਆ ਜਾਵੇਗਾ। ਜੇਕਰ ਕੰਡੀਸ਼ਨ ਦਾ ਨਤੀਜਾ ਗਲਤ (False) ਹੁੰਦਾ ਹੈ, ਤਾਂ "if ਸਟੇਟਮੈਂਟਾਂ ਦਾ ਬਲਾਕ" ਲਾਗੂ ਨਹੀਂ ਕੀਤਾ ਜਾਵੇਗਾ।

ਪਾਈਥਨ ਵਿੱਚ if ਸਟੇਟਮੈਂਟ ਦਾ ਸਿੰਟੈਕਸ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਅੱਗੇ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

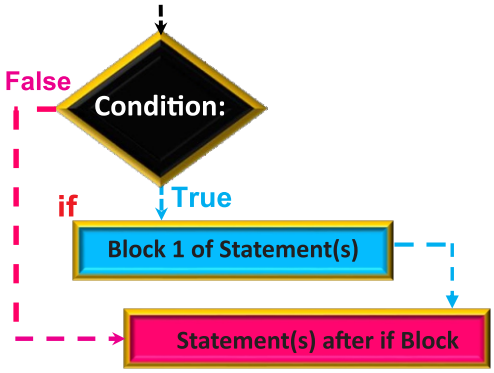


ਚਿੱਤਰ 4.6: if ਸਟੇਟਮੈਂਟ ਦਾ ਸਿੰਟੈਕਸ



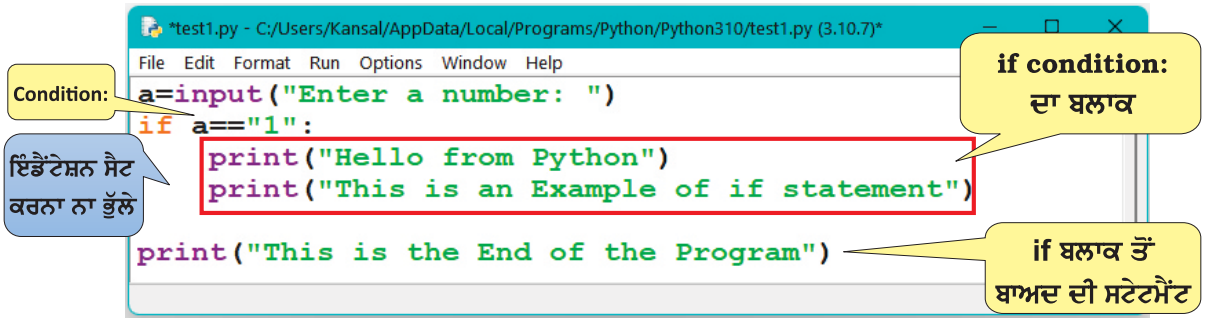
ਚਿੱਤਰ 4.7: if ਸਟੇਟਮੈਂਟ ਦਾ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ

if-ਸਟੇਟਮੈਂਟ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਨੂੰ ਇੱਕ ਫਲੋਅ-ਗ੍ਰਾਫ ਦੇ ਰੂਪ ਵਿੱਚ ਵੀ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਦੇਖਿਆ ਜਾ ਸਕਦਾ ਹੈ:



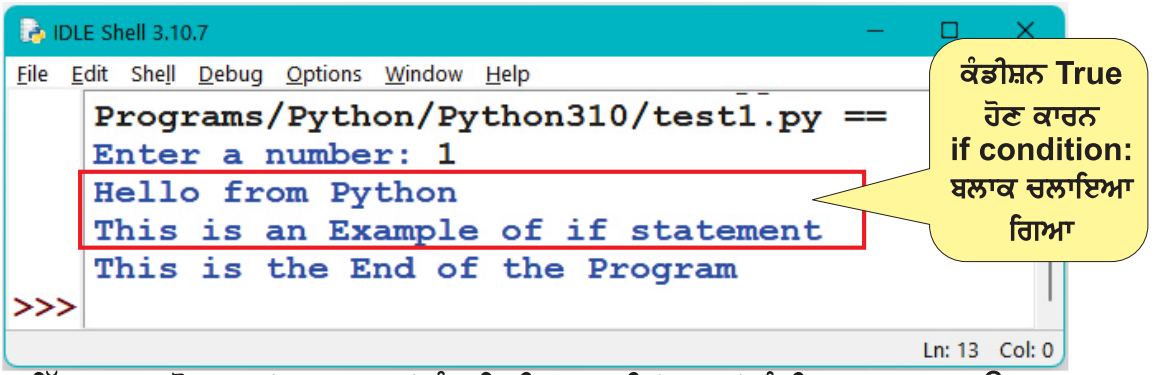
ਚਿੱਤਰ 4.8: **if** ਸਟੇਟਮੈਂਟ ਲਈ ਫਲੋਅ-ਗ੍ਰਾਫ

ਸਿੰਟੈਕਸ ਅਨੁਸਾਰ **if condition** ਤੋਂ ਬਾਅਦ ਕੋਲਨ (:) ਲਗਾਉਣਾ ਲਾਜ਼ਮੀ ਹੈ, ਨਹੀਂ ਤਾਂ ਪਾਈਥਨ ਸਿੰਟੈਕਸ ਐਰਰ (Syntax Error) ਦਿਖਾਏਗਾ। ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਪਹਿਲਾਂ ਚਰਚਾ ਕੀਤੀ ਹੈ, ਪਾਈਥਨ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਬਲਾਕ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਇੰਡੈਂਟੇਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਇਸ ਲਈ "**if condition:**" ਨਾਲ ਜੁੜੇ ਸਾਰੇ ਬਲਾਕ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਇਕੋ ਜਿੰਨ੍ਹੀ ਖਾਲੀ ਥਾਂ ਨਾਲ ਇੰਡੈਂਟ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਚਿੱਤਰ 4.6 ਵਿੱਚ if ਸਟੇਟਮੈਂਟ ਦੇ ਸਿੰਟੈਕਸ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਇਕ ਉਦਾਹਰਨ ਪ੍ਰੋਗਰਾਮ ਰਾਹੀਂ ਪਾਈਥਨ ਵਿੱਚ if ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਹੇਠਾਂ ਦਰਸਾਇਆ ਗਿਆ ਹੈ:



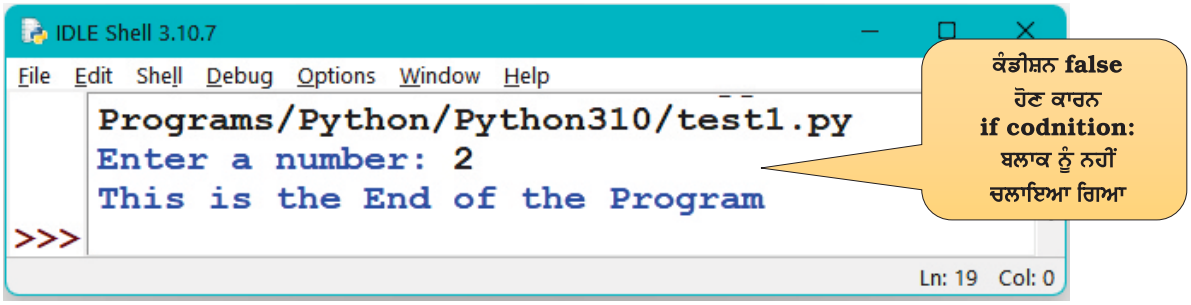
ਪ੍ਰੋਗਰਾਮ (**test1.py**): **if** ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾਉਂਦਾ ਪ੍ਰੋਗਰਾਮ

ਨੋਟ: ਸਿੰਟੈਕਸ ਐਰਰ ਪ੍ਰੋਗਰਾਮ ਕੋਡ ਵਿੱਚ ਇੱਕ ਅਜਿਹੀ ਗਲਤੀ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ ਜੋ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੇ ਨਿਯਮਾਂ ਦੀ ਉਲੰਘਣਾ ਕਰਦੀ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਦੀਆਂ ਗਲਤੀਆਂ ਕੋਡ ਨੂੰ ਸਹੀ ਢੰਗ ਨਾਲ ਚੱਲਣ ਤੋਂ ਰੋਕਦੀਆਂ ਹਨ। ਜੇਕਰ ਪਾਈਥਨ ਕੋਡ ਵਿੱਚ ਕੋਈ ਸਿੰਟੈਕਸ ਗਲਤੀ ਮੌਜੂਦ ਹੋਵੇ, ਤਾਂ ਇਸਦਾ ਇੰਟਰਪ੍ਰੈਟਰ ਐਰਰ ਮੈਸੇਜ਼ ਦਿਖਾਵੇਗਾ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਨੂੰ ਉੱਥੇ ਹੀ ਰੋਕ ਦੇਵੇਗਾ।



ਚਿੱਤਰ 4.9: ਪ੍ਰੋਗਰਾਮ (**test1.py**) ਨੂੰ ਪਹਿਲੀ ਵਾਰ ਸਹੀ (**True**) ਕੰਡੀਸ਼ਨਲ ਨਾਲ ਚਲਾਉਣਾ

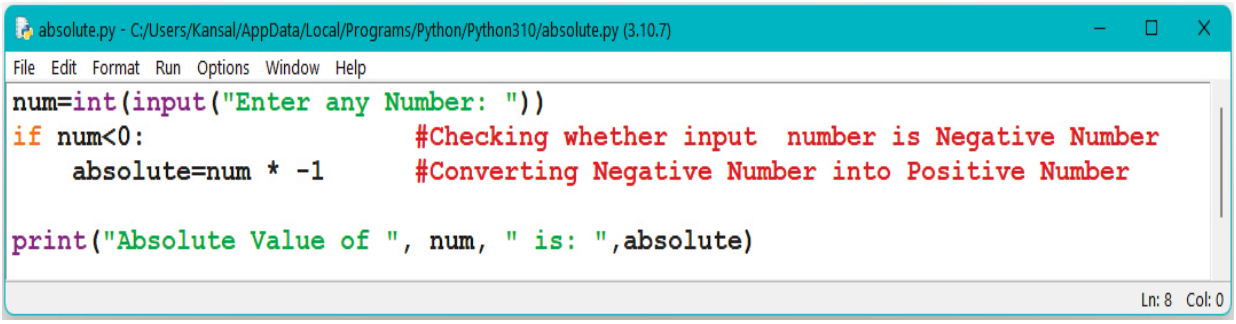
ਪ੍ਰੋਗਰਾਮ **test1.py** ਨੂੰ ਪਹਿਲੀ ਵਾਰ ਚਲਾਉਣ ਸਮੇਂ 1 ਮੁੱਲ (ਸਟਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ, ਕਿਉਂਕਿ **input** ਫੰਕਸ਼ਨ ਸਟਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ ਮੁੱਲ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ) ਇਨਪੁੱਟ ਕਰਕੇ ਵੇਰੀਏਬਲ **'a'** ਨੂੰ ਅਸਾਈਨ ਕੀਤਾ ਗਿਆ। ਇਸ ਲਈ, **if** ਸਟੇਟਮੈਂਟ ਦੀ ਕੰਡੀਸ਼ਨ **a = = "1"** (ਜੋ ਕਿ **"1" = = "1"** ਹੈ) ਦਾ ਮੁਲਾਂਕਣ **True** ਵਿੱਚ ਕੀਤਾ ਗਿਆ, ਜਿਸ ਕਾਰਨ **"if condition:"** ਦੇ ਬਲਾਕ ਵਾਲੇ **print** ਫੰਕਸ਼ਨਾਂ ਨੂੰ ਚਲਾਇਆ ਗਿਆ। **if** ਸਟੇਟਮੈਂਟ ਦੇ ਬਲਾਕ ਦੀ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਹੋਣ ਦੇ ਬਾਅਦ, ਬਲਾਕ ਤੋਂ ਬਾਅਦ ਵਾਲੀ ਸਟੇਟਮੈਂਟ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕੀਤਾ ਗਿਆ।



ਚਿੱਤਰ 4.10: ਪ੍ਰੋਗਰਾਮ (test1.py) ਨੂੰ ਗਲਤ (False) ਕੰਡੀਸ਼ਨ ਨਾਲ ਦੂਜੀ ਵਾਰ ਚਲਾਉਣਾ (test1.py)

ਪ੍ਰੋਗਰਾਮ **test1.py** ਨੂੰ ਦੂਜੀ ਵਾਰ ਚਲਾਉਣ ਸਮੇਂ ਮੁੱਲ 2 (ਸਟਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ, ਕਿਉਂਕਿ **input** ਫੰਕਸ਼ਨ ਸਟਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ ਮੁੱਲ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ) ਇਨਪੁੱਟ ਕਰਕੇ ਵੇਰੀਏਬਲ **'a'** ਨੂੰ ਅਸਾਈਨ ਕੀਤਾ ਗਿਆ। ਇਸ ਲਈ, **if** ਸਟੇਟਮੈਂਟ ਦੀ ਕੰਡੀਸ਼ਨ **a = = "1"** (ਜੋ ਕਿ **"2" = = "1"** ਹੈ) ਦਾ ਮੁਲਾਂਕਣ **False** ਵਿੱਚ ਕੀਤਾ ਗਿਆ, ਜਿਸ ਕਾਰਨ **"if condition:"** ਦੇ ਬਲਾਕ ਵਾਲੇ **print** ਫੰਕਸ਼ਨਾਂ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਨਹੀਂ ਕੀਤਾ ਗਿਆ ਅਤੇ **if** ਬਲਾਕ ਤੋਂ ਬਾਅਦ ਵਾਲੀ ਸਟੇਟਮੈਂਟ ਨੂੰ ਚਲਾਇਆ ਗਿਆ।

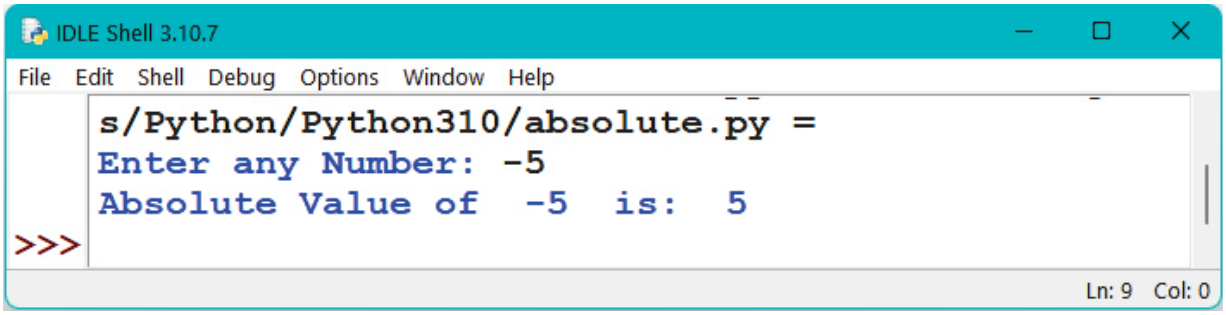
ਆਉ ਇਸ ਸਧਾਰਨ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਦੀ ਹੋਰ ਵਧੀਆ ਸਮਝ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਇੱਕ ਹੋ ਉਦਾਹਰਣ ਪ੍ਰੋਗਰਾਮ ਤੇ ਵਿਚਾਰ ਕਰੀਏ। ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਅਸੀਂ ਦਿੱਤੀ ਗਈ ਸੰਖਿਆ ਦਾ ਐਬਸੋਲਿਊਟ ਮੁੱਲ (Absolute value) ਪਤਾ ਕਰਨਾ ਹੈ। ਇੱਕ ਵਾਸਤਵਿਕ ਸੰਖਿਆ (Real number) **x** ਦਾ ਐਬਸੋਲਿਊਟ ਮੁੱਲ $|x|$, ਇਸਦੇ ਚਿੰਨ੍ਹ ਦੀ ਪਰਵਾਹ ਕੀਤੇ ਬਿਨਾਂ, ਇਕ ਗੈਰ-ਨਕਾਰਾਤਮਕ (non-negative) ਮੁੱਲ **x** ਹੋਵੇਗਾ। ਉਦਾਹਰਨ ਲਈ: 10 ਦਾ ਐਬਸੋਉਿਟ ਮੁੱਲ 10 ਹੋਵੇਗਾ ਅਤੇ -10 ਦਾ ਐਬਲੋਲਿਊਟ ਮੁੱਲ ਵੀ 10 ਹੋਵੇਗਾ। ਸਧਾਰਨ ਸ਼ਬਦਾਂ ਵਿੱਚ, ਐਬਸੋਲਿਊਟ ਮੁੱਲ ਦਾ ਮਤਲਬ ਹੈ ਕਿਸੇ ਸੰਖਿਆ ਦੇ ਅੱਗੇ ਲੱਗੇ ਹੋਏ ਨਕਾਰਾਤਮਕ ਚਿੰਨ੍ਹ (Negative sign) ਨੂੰ ਹਟਾਉਣਾ, ਅਤੇ ਸਾਰੀਆਂ ਸੰਖਿਆਵਾਂ ਨੂੰ ਸਕਾਰਾਤਮਕ (ਜਾਂ ਜ਼ੀਰੋ) ਸਮਝਣਾ। ਇਸ ਲਈ, ਸਾਨੂੰ ਨੈਗੇਟਿਵ ਸੰਖਿਆਵਾਂ ਦਾ ਐਬਸੋਲਿਊਟ ਮੁੱਲ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਸਿਰਫ਼ ਨੈਗੇਟਿਵ ਨੰਬਰਾਂ ਨੂੰ -1 ਨਾਲ ਗੁਣਾ ਕਰਕੇ ਸਕਾਰਾਤਮਕ ਸੰਖਿਆ ਵਿੱਚ ਬਦਲਣਾ ਪਵੇਗਾ। ਐਬਸੋਲਿਊਟ ਮੁੱਲ ਪਤਾ ਕਰਨ ਦਾ ਪ੍ਰੋਗਰਾਮ ਇਸ ਤਰ੍ਹਾਂ ਬਣਾਇਆ ਜਾ ਸਕਦਾ ਹੈ:



ਪ੍ਰੋਗਰਾਮ (absolute.py): ਕਿਸੇ ਨੰਬਰ ਦਾ ਐਬਸੋਲਿਊਟ (absolute) ਮੁੱਲ ਪਤਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਅਸੀਂ ਯੂਜ਼ਰ ਤੋਂ ਸਟ੍ਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ ਇੱਕ ਨੰਬਰ ਪ੍ਰਾਪਤ ਕਰ ਰਹੇ ਹਾਂ (ਕਿਉਂਕਿ **input()** ਫੰਕਸ਼ਨ ਸਿਰਫ ਸਟ੍ਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ ਸਾਰੇ ਮੁੱਲ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ), ਅਤੇ ਇਨਪੁਟ ਮੁੱਲ (ਸਟ੍ਰਿੰਗ ਫਾਰਮੈਟ ਵਿੱਚ) ਨੂੰ ਇੰਟੀਜ਼ਰ ਵਿੱਚ

ਤਬਦੀਲ ਕਰਨ ਤੋਂ ਬਾਅਦ **(int()** ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਨਾਲ) ਇਸਨੂੰ ਇੱਕ ਵੇਰੀਏਬਲ **num** ਨੂੰ ਅਸਾਈਨ ਕਰ ਰਹੇ ਹਾਂ। ਇਸ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਟੈਸਟ ਕਰ ਰਹੇ ਹਾਂ ਕਿ ਕੀ **num** ਇੱਕ ਰਿਣਾਤਮਕ ਮੁੱਲ ਹੈ (ਜੇ **num<0**) ਜਾਂ ਨਹੀਂ। ਜੇਕਰ ਇਹ ਨਕਾਰਾਤਮਕ **(negative)** ਮੁੱਲ ਹੈ (ਜੇਕਰ **if condition:** ਦਾ ਨਤੀਜਾ **true** ਹੈ), ਤਾਂ ਇਸਨੂੰ -1 (ਜੋ ਕਿ **num** * -1 ਹੋਵੇਗੀ) ਨਾਲ ਗੁਣਾ ਕਰਕੇ ਸਕਾਰਾਤਮਕ **(positive)** ਮੁੱਲ ਵਿੱਚ ਬਦਲ ਦਿਤਾ ਜਾਵੇਗਾ। ਇਸ ਤੋਂ ਬਾਅਦ ਅਸੀਂ **if** ਬਲਾਕ ਤੋਂ ਬਾਹਰ ਆ ਜਾਵਾਂਗੇ ਜਿੱਥੇ ਸਾਡੇ ਕੋਲ ਇਨਪੁੱਟ ਕੀਤੇ ਗਏ ਨਕਾਰਾਤਮਕ ਮੁੱਲ ਦਾ ਐਬਸੋਲਿਉਟ ਮੁੱਲ ਹੋਵੇਗਾ। ਪਰੰਤੂ ਜੇਕਰ ਯੂਜ਼ਰ ਕੋਈ ਸਕਾਰਾਤਮਕ ਜਾਂ ਜ਼ੀਰੋ ਮੁੱਲ ਇਨਪੁੱਟ ਕਰਦਾ ਹੈ, ਤਾਂ **if condition:** ਗਲਤ **(false)** ਹੋ ਜਾਵੇਗੀ ਅਤੇ **if** ਬਲਾਕ ਨੂੰ ਲਾਗੂ ਨਹੀਂ ਕੀਤਾ ਜਾਵੇਗਾ। ਇਸ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਚਲਾਉਣ **(F5)** ਤੋਂ ਬਾਅਦ ਸਾਨੂੰ ਹੇਠਾਂ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਮਿਲੇਗੀ।



ਚਿੱਤਰ 4.11: ਪ੍ਰੋਗਰਾਮ **(absolute.py)** ਦੀ ਆਊਟਪੁੱਟ

ਕੰਡੀਸ਼ਨ ਕੋਈ ਵੀ ਪਾਈਥਨ ਐਕਸਪ੍ਰੈਸ਼ਨ ਹੋ ਸਕਦੀ ਹੈ (ਜਿਵੇਂ ਕਿ: ਜੋ ਕੋਈ ਮੁੱਲ ਵਾਪਸ ਕਰਦੀ ਹੈ)। ਨਿਮਨਲਿਖਤ ਮੁੱਲ ਜਦੋਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਦੁਆਰਾ ਵਾਪਸ ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਤਾਂ ਉਹਨਾਂ ਨੂੰ **False** ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ:

None — A String of Length Zero
Number Zero — An Empty Collection

Test Yourself ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਵਿੱਚ ਗਲਤੀਆਂ ਲੱਭੋ ਅਤੇ ਸਹੀ ਕੋਡ ਲਿਖੋ:

4.1

```
age=18
if age>=18
    print("Eligible to Vote")
```

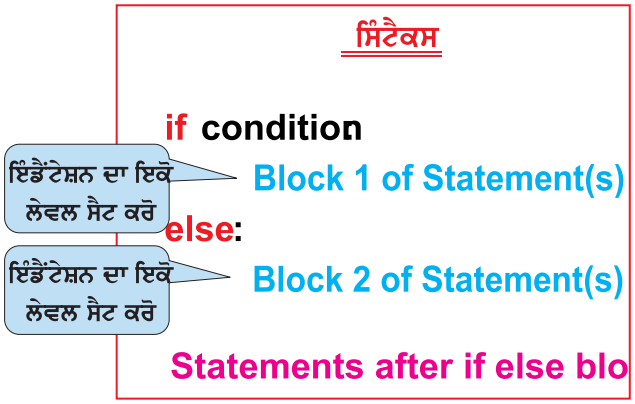
```
age=18
if age>=18:
print("Eligible to Vote")
```

```
age=18
if age>=18:
    print("Eligible to Vote")
        print("Elections in India")
```

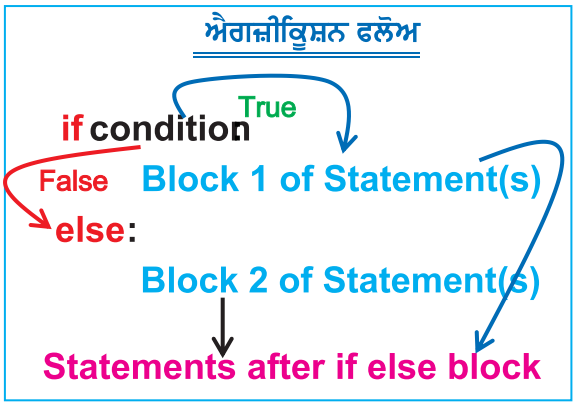
4.4.2 if-else ਸਟੇਟਮੈਂਟ (if-else statement):

ਇਹ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਦਾ ਇੱਕ ਹੋਰ ਰੂਪ ਹੈ। if-else ਸਟੇਟਮੈਂਟ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਕਰਦੀ ਹੈ ਅਤੇ ਮੁਲਾਂਕਣ ਤੋਂ ਬਾਅਦ ਜੇਕਰ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਸਹੀ (**True**) ਨਤੀਜਾ ਦਿੰਦੀ ਹੈ ਤਾਂ **if** ਬਲਾਕ ਦੇ ਕੋਡ ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਪਰ ਜੇਕਰ ਕੰਡੀਸ਼ਨ ਦੇ ਮੁਲਾਂਕਣ ਦਾ ਨਤੀਜਾ ਗਲਤ (**False**) ਆਉਂਦਾ ਹੈ, ਤਾਂ ਇਹ **else** ਬਲਾਕ

ਦੇ ਕੋਡ ਨੂੰ ਚਲਾਉਂਦੀ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ **if-else** ਸਟੇਟਮੈਂਟ ਦਾ ਸਿੰਟੈਕਸ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਅੱਗੇ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:

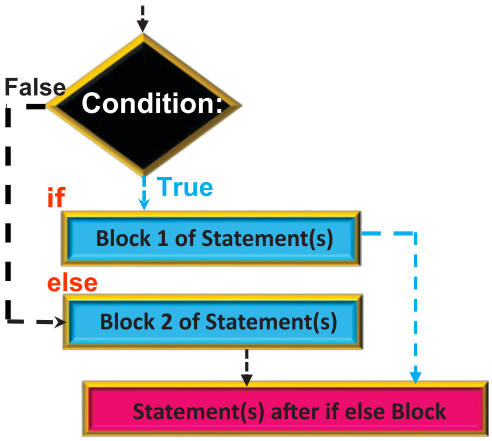


ਚਿੱਤਰ 4.12: **if-else** ਦਾ ਸਿੰਟੈਕਸ



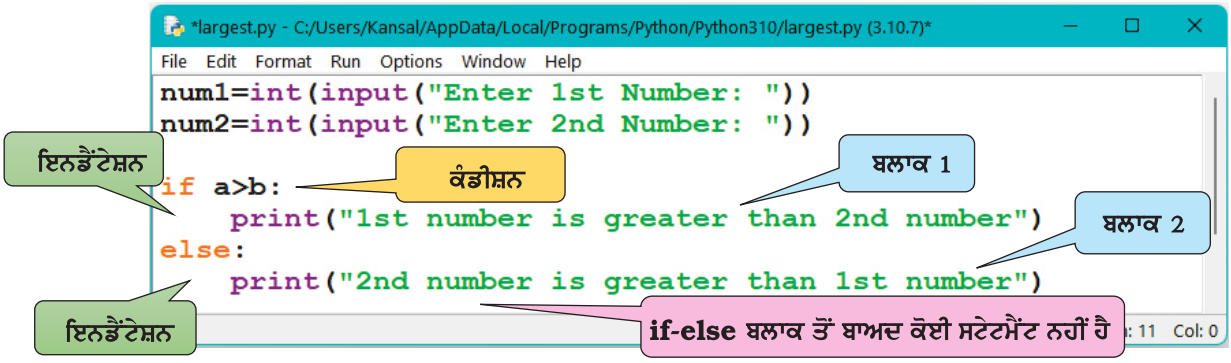
ਚਿੱਤਰ 4.13: **if-else** ਦਾ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ

if-else ਸਟੇਟਮੈਂਟ ਦੇ ਇਸ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਨੂੰ ਫਲੋਅ-ਗ੍ਰਾਫ ਦੇ ਰੂਪ ਵਿਚ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਦੇਖਿਆ ਜਾ ਸਕਦਾ ਹੈ:



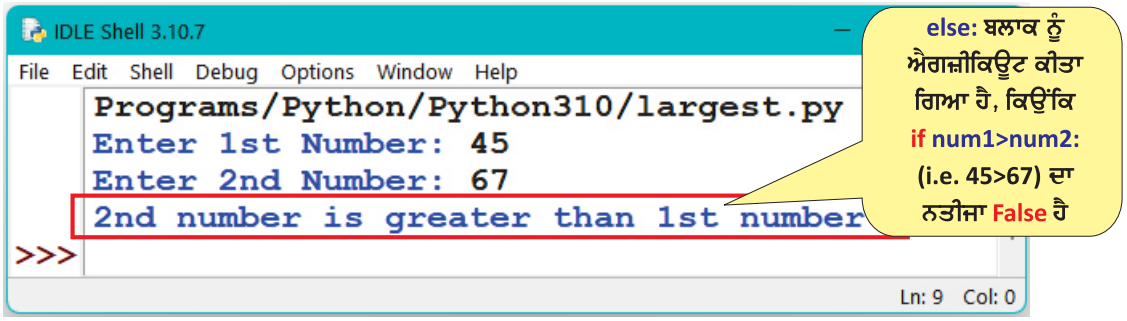
ਚਿੱਤਰ 4.14: **if-else** ਦਾ ਫਲੋਅ ਗ੍ਰਾਫ

ਸਿੰਟੈਕਸ ਅਨੁਸਾਰ **if condition** ਅਤੇ **else** ਤੋਂ ਬਾਅਦ ਕੋਲਨ (:) ਲਗਾਉਣਾ ਲਾਜ਼ਮੀ ਹੈ, ਨਹੀਂ ਤਾਂ ਪਾਈਥਨ ਸਿੰਟੈਕਸ ਐਰਰ (Syntax error) ਦਿਖਾਵੇਗਾ। ਹੇਠਾਂ ਦਿੱਤੇ ਉਦਾਹਰਨ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ if-else ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ। ਇਹ ਪ੍ਰੋਗਰਾਮ ਦੋ ਦਿੱਤੇ ਗਏ ਮੁੱਲਾਂ ਵਿੱਚੋਂ ਸਭ ਤੋਂ ਵੱਡਾ (largest) ਮੁੱਲ ਪਤਾ ਕਰਨ ਲਈ ਬਣਾਇਆ ਗਿਆ ਹੈ:



ਪ੍ਰੋਗਰਾਮ (**largest.py**): ਦੋ ਅੰਕਾਂ ਵਿਚੋਂ ਵੱਡਾ (**largest**) ਅੰਕ ਪਤਾ ਕਰਨ ਦਾ ਪ੍ਰੋਗਰਾਮ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਅਸੀਂ ਯੂਜ਼ਰ ਤੋਂ ਦੋ ਨੰਬਰ (**num1** ਅਤੇ **num2**) ਇਨਪੁੱਟ ਕਰਵਾ ਰਹੇ ਹਾਂ ਅਤੇ **if condition: (if num1>num2:)** ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਉਹਨਾਂ ਨੂੰ ਟੈਸਟ ਕਰ ਰਹੇ ਹਾਂ। ਜੇਕਰ ਇਸ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ (**True**) ਆਉਂਦਾ ਹੈ, ਤਾਂ **if** ਸਟੇਟਮੈਂਟ ਦਾ ਬਲਾਕ ਐਗਜ਼ੀਕਿਊਟ ਹੋ ਜਾਵੇਗਾ, ਨਹੀਂ ਤਾਂ **else** ਭਾਗ ਵਾਲਾ ਬਲਾਕ ਐਗਜ਼ੀਕਿਊਟ ਕੀਤਾ ਜਾਵੇਗਾ।



ਚਿੱਤਰ 4.15: ਪ੍ਰੋਗਰਾਮ **(largest.py)** ਦੀ

ਉਪਰੋਕਤ ਆਉਟਪੁੱਟ ਵਿਚ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਦੌਰਾਨ ਯੂਜ਼ਰ, ਵੇਰੀਏਬਲ **num1** ਅਤੇ **num2** ਲਈ ਕ੍ਰਮਵਾਰ ਦੋ ਮੁੱਲ 45 ਅਤੇ 67 ਇਨਪੁੱਟ ਕਰ ਰਿਹਾ ਹੈ। ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ **num1 > num2 (45 > 67)** ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ ਇਸ ਦਾ ਨਤੀਜਾ ਗਲਤ **(false)** ਆ ਰਿਹਾ ਹੈ, ਇਸਲਈ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਕੰਟਰੋਲ, ਪ੍ਰੋਗਰਾਮ ਦੇ **else** ਭਾਗ ਨੂੰ ਚਲਾ ਰਿਹਾ ਹੈ। ਇਹੀ ਕਾਰਨ ਹੈ ਕਿ ਯੂਜ਼ਰ ਨੂੰ ਆਉਟਪੁੱਟ **"2nd number is greater than 1st number"** ਪ੍ਰਾਪਤ ਹੋ ਰਹੀ ਹੈ।

4.2 TEST YOURSELF — ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਵਿੱਚ ਗਲਤੀਆਂ ਲੱਭੋ ਅਤੇ ਸਹੀ ਕੋਡ ਲਿਖੋ:

```
age=18
if age>=18:
print("Eligible to Vote")
else:
print("Not Eligible to Vote")
```

ਉਦਾਹਰਣ:

```
#Program to find the divisibilty of two numbers
dividend=int(input("Enter value of dividend: "))
divisor=int(input("Enter value of divisor: "))
if(dividend%divisor==0):
    print(dividend, "is divisible by ", divisor)
else:
    print(dividend, "is not divisible by ", divisor)
```

ਆਊਟਪੁੱਟ:

```
Enter value of dividend: 10
Enter value of divisor: 2
10 is divisible by  2
```

Test Yourself

ਹੇਠਾਂ ਦਿੱਤੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਆਊਟਪੁਟ ਲਿਖੋ:

4.3

```
num1=11
if num1%2 == 0:
    print(num1, " is Even Number")
else:
    print(num1, " is Odd Number")
```

ਆਊਟਪੁਟ:

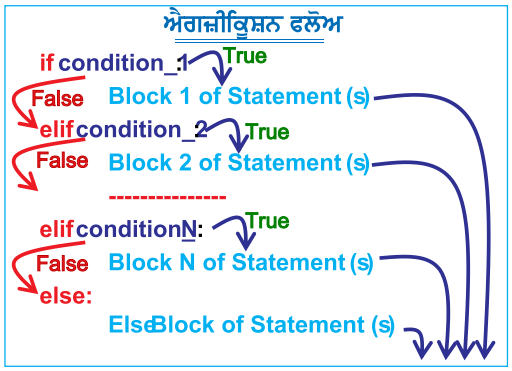
4.4.3 if-elif-else ਸਟੇਟਮੈਂਟ (if-elif-else Statement):

ਇਹ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਦਾ ਇੱਕ ਹੋਰ ਰੂਪ ਹੈ ਜਿਸ ਵਿੱਚ ਅਸੀਂ ਕਈ ਕੰਡੀਸ਼ਨਾਂ ਦੀ ਟੈਸਟ/ਜਾਂਚ (test multiple conditions) ਕਰ ਸਕਦੇ ਹਾਂ। ਇਸ ਸਟੇਟਮੈਂਟ ਵਿਚ ਵਰਤਿਆ ਜਾਂਦਾ **elif** "**else if**" ਦਾ ਛੋਟਾ ਰੂਪ ਹੈ। ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਦੇ ਇਸ ਰੂਪ ਵਿੱਚ **if condition:** ਸਟੇਟਮੈਂਟ ਨੂੰ **elif condition:** ਅਤੇ **else:** ਸਟੇਟਮੈਂਟਸ ਨਾਲ ਟੈਸਟਾਂ ਦੀ ਇੱਕ ਲੜੀ (series of test) ਨੂੰ ਜੋੜਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਸਟੇਟਮੈਂਟ ਵਿਚ ਅਸੀਂ **elif** ਦਾ ਅਰਥ ਇਸ ਤਰ੍ਹਾਂ ਸਮਝ ਸਕਦੇ ਹਾਂ, "ਜੇ ਪਿਛਲੀ ਕੰਡੀਸ਼ਨ ਸਹੀ ਨਹੀਂ (not true) ਸੀ, ਤਾਂ ਇਸ if condition ਦੀ ਜਾਂਚ ਕਰੋ।

ਸਿੰਟੈਕਸ

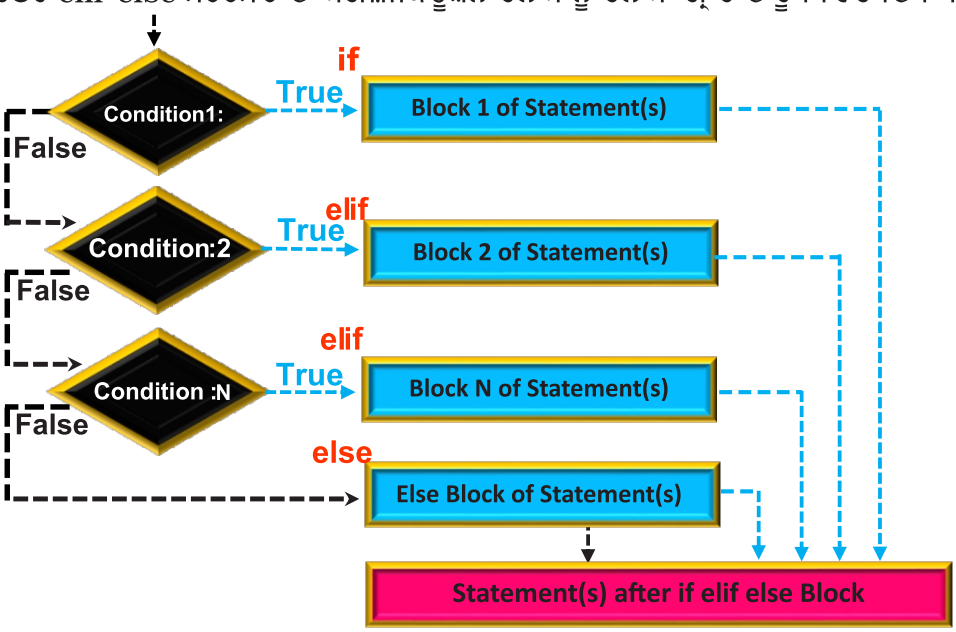
```
if condition_1
        Block 1 of Statement(s)
elif condition_2
        Block 2 of Statement(s)
        ----------
elif conditionN:
        Block N of Statement(s)
else:
        Else Block of Statement(s)
```

ਚਿੱਤਰ 4.16: **if-elif-else** ਦਾ ਸਿੰਟੈਕਸ



ਚਿੱਤਰ 4.17: **if-elif-else** ਦਾ ਐਗਜ਼ੀਕਿਊਸ਼ਨ

ਹੇਠਾਂ ਦਿੱਤਾ ਚਿੱਤਰ elif-else ਸਟੇਟਮੈਂਟ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਨੂੰ ਫਲੋਅ-ਗ੍ਰਾਫ ਦੇ ਰੂਪ ਵਿਚ ਦਿਖਾ ਰਿਹਾ ਹੈ:

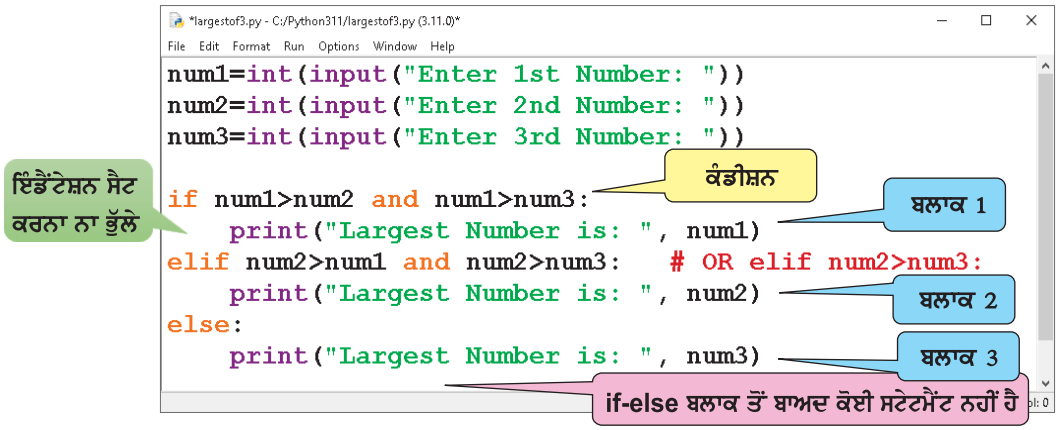


ਚਿੱਤਰ 4.18: **if-elif-else** ਸਟੇਟਮੈਂਟ ਦਾ ਫਲੋਅ ਗ੍ਰਾਫ

ਚਿੱਤਰ ਵਿੱਚ ਦਰਸਾਏ ਗਏ ਸਿੰਟੈਕਸ ਅਨੁਸਾਰ ਹਰੇਕ condition ਅਤੇ else ਸਟੇਟਮੈਂਟ ਤੋਂ ਬਾਅਦ ਕੋਲਨ (:) ਲਗਾਉਣਾ ਜ਼ਰੂਰੀ ਹੁੰਦਾ ਹੈ, ਨਹੀਂ ਤਾਂ ਪਾਈਥਨ ਸਿੰਟੈਕਸ ਐਰਰ ਦਿਖਾਵੇਗਾ। ਦਿੱਤੀਆਂ ਕੰਡੀਸ਼ਨਾਂ ਦਾ ਮੁਲਾਂਕਣ ਠੀਕ ਉਸੇ ਕ੍ਰਮ ਵਿੱਚ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਸ ਕ੍ਰਮ ਵਿੱਚ ਉਹਨਾਂ ਨੂੰ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਵਿਚੋਂ ਜਿਹੜੀ ਸਭ ਤੋਂ ਪਹਿਲੀ ਕੰਡੀਸ਼ਨ ਸਹੀ (True) ਹੋ ਜਾਂਦੀ ਹੈ, ਤਾਂ ਫਿਰ ਉਸ ਸਹੀ (True) ਕੰਡੀਸ਼ਨ ਨਾਲ ਸੰਬੰਧਿਤ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਬਲਾਕ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰ ਦਿਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਹੋਰ ਕੰਡੀਸ਼ਨਾਂ ਦੀ ਜਾਂਚ ਕੀਤੇ ਬਿਨਾਂ ਇਸ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਵਿਚੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ।

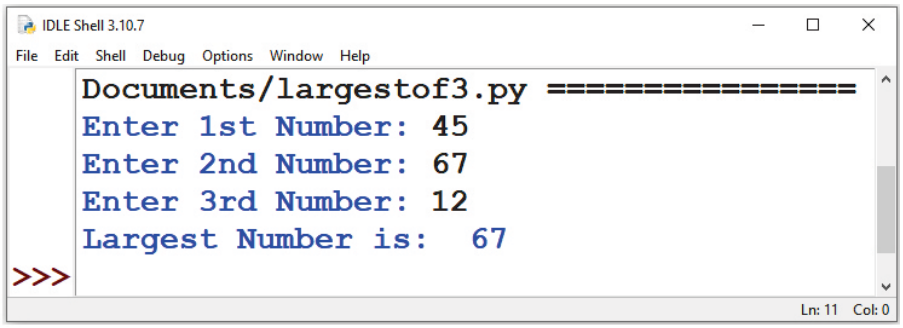
ਅਸੀਂ ਆਪਣੀ ਜ਼ਰੂਰਤ ਅਨੁਸਾਰ ਜਿੰਨੇ ਮਰਜੀ ਵਾਰ ਇਸ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਦੇ if ਅਤੇ else ਭਾਗਾਂ ਵਿਚਕਾਰ elif ਸਟੇਟਮੈਂਟਾਂ ਰੱਖ ਸਕਦੇ ਹਾਂ। ਇੱਕ if ਤੋਂ ਬਾਅਦ elif ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਕਈ ਵਾਰ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ, ਪਰ else ਭਾਗ ਵੱਧ ਤੋਂ ਵੱਧ ਇੱਕ ਵਾਰ ਹੀ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ। C, C++ ਅਤੇ Java ਵਰਗੀਆਂ ਹੋਰ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ **switch-case** ਦੀ ਤਰ੍ਹਾਂ ਪਾਈਥਨ **switch-case** ਸਟੇਟਮੈਂਟ ਪ੍ਰਦਾਨ ਨਹੀਂ ਕਰਦਾ ਹੈ ਪਰ ਅਸੀਂ switch-case ਸਟੇਟਮੈਂਟ ਦੀ ਜਗ੍ਹਾ if-elif-else ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਉਦਾਹਰਨ ਪ੍ਰੋਗਰਾਮ ਪਾਈਥਨ ਵਿੱਚ **if-elif-else** ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾ ਰਿਹਾ ਹੈ। ਇਹ ਪ੍ਰੋਗਰਾਮ ਤਿੰਨ ਦਿੱਤੇ ਗਏ ਮੁੱਲਾਂ ਵਿੱਚੋਂ ਸਭ ਤੋਂ ਵੱਡਾ ਮੁੱਲ ਪਤਾ ਕਰ ਰਿਹਾ ਹੈ:



ਪ੍ਰੋਗਰਾਮ (largestof3.py): ਤਿੰਨ੍ ਅੰਕਾਂ ਵਿਚੋਂ ਵੱਡਾ ਅੰਕ ਪਤਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਅਸੀਂ ਕੀਬੋਰਡ ਰਾਹੀਂ ਤਿੰਨ ਨੰਬਰ **(num1, num2, num3)** ਪ੍ਰਾਪਤ ਕਰ ਰਹੇ ਹਾਂ ਅਤੇ **if-elif-else** ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਉਹਨਾਂ ਨੂੰ ਟੈਸਟ ਕਰ ਰਹੇ ਹਾਂ। ਸਭ ਤੋਂ ਪਹਿਲਾਂ **if condition:** ਨੂੰ ਟੈਸਟ ਕੀਤਾ ਗਿਆ ਹੈ ਜਿੱਥੇ **"and"** ਲਾਜ਼ੀਕਲ ਆਪਰੇਟਰ **(num1 > num2 and num1 > num3)** ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਦੋ ਕੰਡੀਸ਼ਨਾਂ ਦੀ ਜਾਂਚ ਕੀਤੀ ਗਈ ਹੈ ਜੋ ਸਿਰਫ ਉਦੋਂ ਹੀ ਸਹੀ **(true)** ਨਤੀਜਾ ਪ੍ਰਦਾਨ ਕਰੇਗੀ ਜਦੋਂ ਦੋਵੇਂ ਕੰਡੀਸ਼ਨਾਂ ਸਹੀ **(True)** ਹੋਣਗੀਆਂ। ਜੇਕਰ ਇਹਨਾਂ ਕੰਡੀਸ਼ਨਾਂ ਦਾ ਮੁਲਾਂਕਣ ਗਲਤ **(False)** ਹੁੰਦਾ ਹੈ, ਤਾਂ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ **elif condition:** ਵੱਲ ਟ੍ਰਾਂਸਫਰ ਕਰ ਦਿਤਾ ਜਾਵੇਗਾ, ਜਿੱਥੇ ਦੇ ਹੋਰ ਕੰਡੀਸ਼ਨਾਂ **(num2>num1 and num2>num3)** ਦਾ ਮੁਲਾਂਕਣ **"and"** ਲਾਜ਼ੀਕਲ ਆਪਰੇਟਰ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੀਤਾ ਜਾ ਰਿਹਾ ਹੈ। ਜੇਕਰ ਇਸ **elif** ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ **(true)** ਹੋ ਜਾਂਦਾ ਹੈ, ਤਾਂ **elif** ਸਟੇਟਮੈਂਟਾਂ ਦਾ ਬਲਾਕ ਐਗਜ਼ੀਕਿਊਟ ਹੋ ਜਾਵੇਗਾ, ਨਹੀਂ ਤਾਂ **else** ਹਿੱਸੇ ਦਾ ਬਲਾਕ ਐਗਜ਼ੀਕਿਊਟ ਹੋਵੇਗਾ। ਆਓ ਹੁਣ ਇਸ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਚਲਾ ਕੇ ਇਸਦੀ ਆਊਟਪੁੱਟ ਦੇਖਦੇ ਹਾਂ:



ਚਿੱਤਰ 4.19: ਪ੍ਰੋਗਰਾਮ (**largestof3.py**) ਦੀ ਆਊਟਪੁੱਟ

ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਦੌਰਾਨ ਅਸੀਂ ਵੇਰੀਏਬਲ **num1, num2** ਅਤੇ **num3** ਲਈ ਕ੍ਰਮਵਾਰ ਤਿੰਨ ਮੁੱਲ 45, 67 ਅਤੇ 12 ਇਨਪੁਟ ਕੀਤਾ ਹੈ। ਕੰਡੀਸ਼ਨ **num1 > num2 and num1 > num3 (i.e. 45 > 67 and 45 > 12)** ਦਾ ਮੁਲਾਂਕਣ ਕਰਨ ਤੋਂ ਬਾਅਦ ਇਸਦਾ ਨਤੀਜਾ ਗਲਤ (**False**) ਪ੍ਰਾਪਤ ਹੋ ਰਿਹਾ ਹੈ, ਇਸਲਈ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੌਅ ਕੰਟਰੋਲ ਪ੍ਰੋਗਰਾਮ ਦੇ **elif** ਭਾਗ ਵੱਲ ਟ੍ਰਾਂਸਫਰ ਕੀਤਾ ਜਾ ਰਿਹਾ ਹੈ ਜਿੱਥੇ ਇੱਕ ਹੋਰ ਕੰਡੀਸ਼ਨ **num2 > num1 and num2 > num3 (i.e. 67 > 45 and 67 > 12)** ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜੋ ਸਹੀ **(True)** ਮੁੱਲ ਦੇ ਰਿਹਾ ਹੈ ਅਤੇ ਜਿਸ ਕਾਰਨ ਸਾਨੂੰ ਆਉਟਪੁੱਟ **"Largest Number is: 67"** ਪ੍ਰਾਪਤ ਹੋ ਰਹੀ ਹੈ। ਇਸ ਤੋਂ ਬਾਅਦ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ **if-elif-else** ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਤੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ।

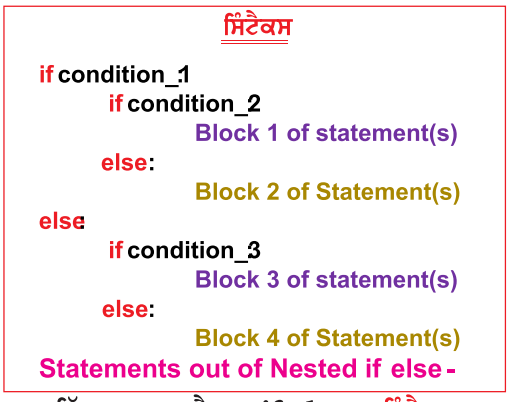
4.4 TEST YOURSELF — ਹੇਠਾਂ ਦਿਤੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਆਊਟਪੁੱਟ ਲਿਖੋ:

```
#Program to calculate percentage marks and show the Division
phy=65
che=70
math=58
obt_marks=phy+che+math
print("Total Marks Obtained: ", obt_marks)
percent=obt_marks/300.0 * 100.0
print("Percentage Marks: ",percent)
if percent>=80:
    print("First Division with Distinction")
elif percent>=60:
    print("First Division")
elif percent>=50:
    print("Second Division")
elif percent>=35:
    print("Third Division")
else:
    print("Fail")
```

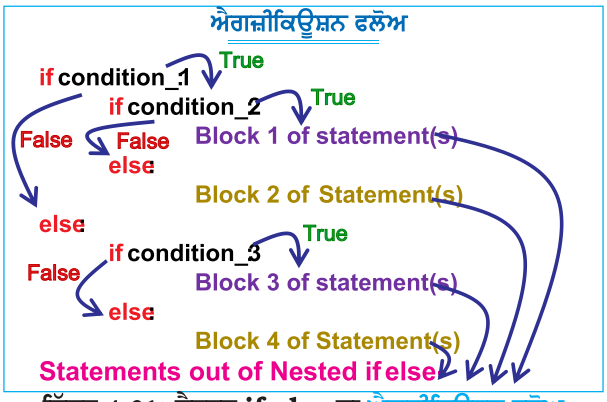
ਆਊਟਪੁੱਟ:

4.4.4 ਨੈਸਟਡ if-else ਸਟੇਟਮੈਂਟ (Nested if-else Statement):

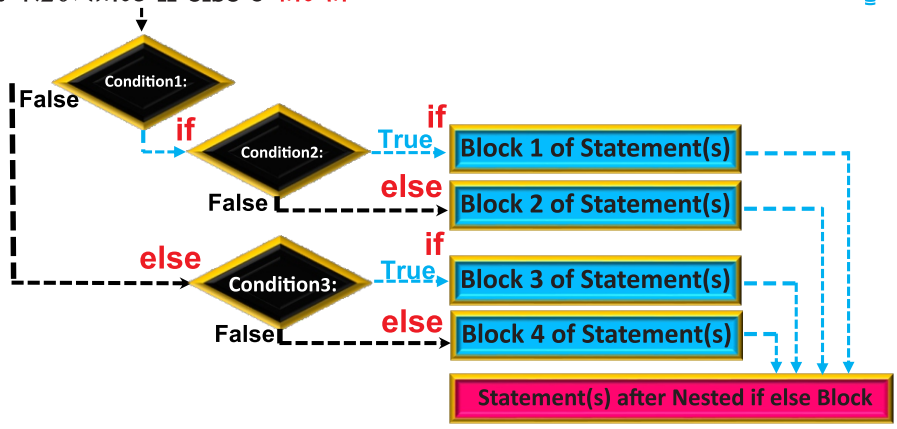
ਇਹ if ਸਟੇਟਮੈਂਟ ਦਾ ਇੱਕ ਹੋਰ ਰੂਪ ਹੈ ਜੋ ਉਸ ਸਮੇਂ ਉਪਯੋਗੀ ਹੁੰਦਾ ਹੈ ਜਦੋਂ ਅਸੀਂ ਫੈਸਲਿਆਂ ਦੀ ਇੱਕ ਲੜੀ (series of decisions) ਬਨਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ। ਨੈਸਟਡ if-else ਸਟੇਟਮੈਂਟ ਵਿਚ ਇੱਕ if-else ਸਟੇਟਮੈਂਟ ਦੇ ਅੰਦਰ ਇੱਕ ਹੋਰ if-else ਸਟੇਟਮੈਂਟ ਲਿਖੀ ਜਾਂਦੀ ਹੈ। ਪਾਈਥਨ ਵਿੱਚ ਅਸੀਂ ਇੱਕ if-else ਸਟੇਟਮੈਂਟ ਅੰਦਰ ਕਿੰਨ੍ਹੀਆਂ ਮਰਜੀ if-else ਸਟੇਟਮੈਂਟਸ ਲਗਾ ਕੇ ਨੈਸਟਿੰਗ ਕਰ ਸਕਦੇ ਹਾਂ। ਨੈਸਟਿੰਗ ਦੇ ਲੇਵਲ ਨੂੰ ਵੱਖ ਕਰਨ ਦਾ ਇੱਕੋ-ਇੱਕ ਤਰੀਕਾ ਹੈ ਇੰਡੈਂਟੇਸ਼ਨ। ਅੱਗੇ ਦਿੱਤੇ ਚਿਤਰਾਂ ਵਿਚ ਨੈਸਟਡ if-else ਸਟੇਟਮੈਂਟ ਦੇ ਸਿੰਟੈਕਸ, ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਅਤੇ ਫਲੋਅ ਗ੍ਰਾਫ ਨੂੰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ:



ਚਿੱਤਰ 4.20: ਨੈਸਟਡ **if-else** ਦਾ ਸਿੰਟੈਕਸ

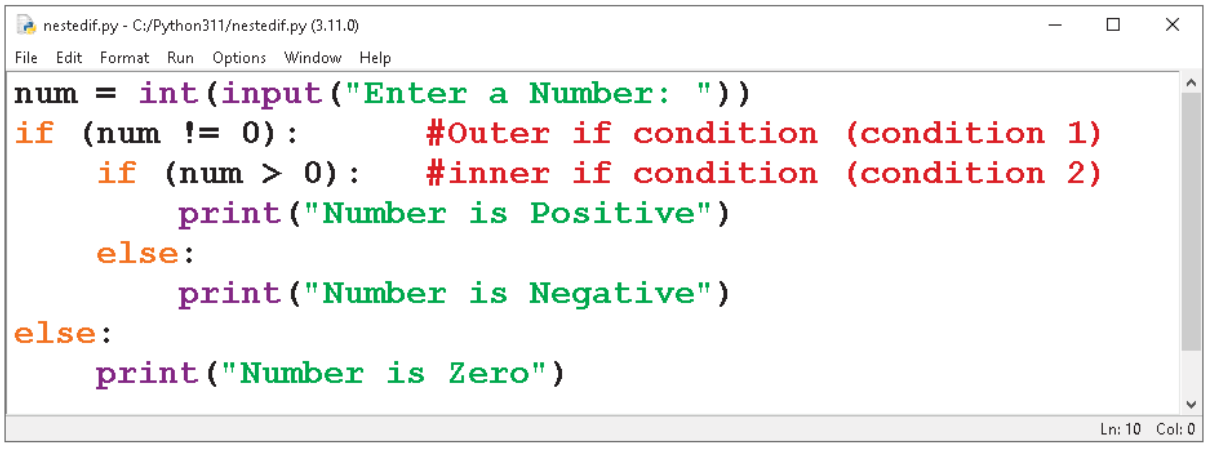


ਚਿੱਤਰ 4.21: ਨੈਸਟਡ **if-else** ਦਾ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ



ਚਿੱਤਰ 4.22: ਨੈਸਟਡ **if-else** ਦਾ ਫਲੋਅ ਗ੍ਰਾਫ

ਨੈਸਟਡ if ਸਟੇਟਮੈਂਟ ਵਿੱਚ ਅਸੀਂ ਇੱਕ if-elif-else ਸਟੇਟਮੈਂਟ ਵਿਚ ਹੋਰ if-elif-else ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੀ ਨੈਸਟਿੰਗ ਕਰ ਸਕਦੇ ਹਾਂ। ਅੱਗੇ ਦਿੱਤਾ ਗਿਆ ਉਦਾਹਰਣ ਪ੍ਰੋਗਰਾਮ ਪਾਈਥਨ ਵਿੱਚ ਨੈਸਟਡ if-else ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾ ਰਿਹਾ ਹੈ:

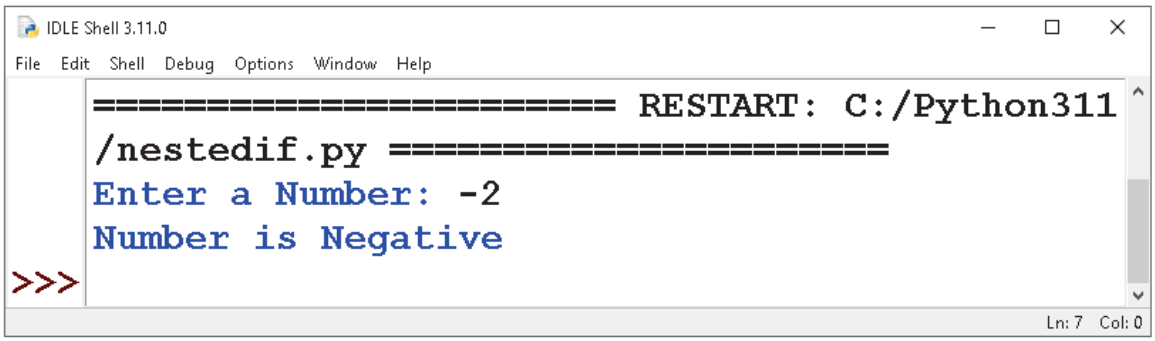


ਪ੍ਰੋਗਰਾਮ **(nestedif.py):** ਦਿੱਤਾ ਗਿਆ ਨੰਬਰ ਜ਼ੀਰੋ, ਸਕਾਰਾਤਮਕ ਜਾਂ ਨਕਾਰਾਤਮਕ ਹੈ, ਪਤਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਅਸੀਂ ਸਿਰਫ if ਬਲਾਕ ਦੇ ਅੰਦਰ "if-else" ਬਲਾਕ ਸ਼ਾਮਲ ਕੀਤਾ ਹੈ। ਅਸੀਂ "else" ਬਲਾਕ ਦੇ ਅੰਦਰ ਵੀ ਜ਼ਰੂਰਤ ਅਨੁਸਾਰ ਹੋਰ "if-else" ਬਲਾਕ ਸ਼ਾਮਲ ਕਰ ਸਕਦੇ ਹਾਂ।

ਉਪਰੋਕਤ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਪਹਿਲੀ if condition: (if (num!=0):) ਵਿੱਚ ਅਸੀਂ ਇਹ ਟੈਸਟ ਕਰ ਰਹੇ ਹਾਂ ਕਿ ਕੀ num ਜ਼ੀਰੋ ਦੇ ਬਰਾਬਰ ਹੈ ਜਾਂ ਨਹੀਂ, ਜੇਕਰ ਇਸ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ (true) ਆਉਂਦਾ ਹੈ ਤਾਂ ਪ੍ਰੋਗਰਾਮ

ਅੰਦਰੂਨੀ if condition: (if (num>0):) ਦੀ ਜਾਂਚ ਕਰੇਗਾ ਜੋ ਇਹ ਟੈਸਟ ਕਰ ਰਿਹਾ ਹੈ ਕਿ ਕੀ num ਜ਼ੀਰੋ ਤੋਂ ਵੱਡੀ ਹੈ ਜਾਂ ਨਹੀਂ, ਜੇਕਰ ਇਸ ਅੰਦਰੂਨੀ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ (true) ਹੁੰਦਾ ਹੈ ਤਾਂ ਇਹ print("Number is Positive") ਸਟੇਟਮੈਂਟ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰੇਗਾ ਅਤੇ ਨੇਸਟਡ if ਸਟੇਟਮੈਂਟ ਨੂੰ ਛੱਡ ਕੇ ਉਸ ਵਿੱਚੋਂ ਬਾਹਰ ਆ ਜਾਵੇਗਾ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਸਮਾਪਤ ਹੋ ਜਾਵੇਗਾ। ਪਰ ਜੇਕਰ ਅੰਦਰਲੀ ਕੰਡੀਸ਼ਨ (if (num>0:) ਗਲਤ (false) ਹੋ ਜਾਂਦੀ ਹੈ ਤਾਂ else ਭਾਗ ਵਾਲੇ print("Number is Negative") ਸਟੇਟਮੈਂਟ ਨੂੰ ਚਲਾਇਆ ਜਾਵੇਗਾ। ਅੰਦਰੂਨੀ if else ਸਟੇਟਮੈਂਟ ਦੀ ਜਾਂਚ ਤਾਂ ਹੀ ਕੀਤੀ ਜਾਵੇਗੀ ਜੇਕਰ ਬਾਹਰੀ if condition: (if (num!=0):) ਸਹੀ (true) ਹੋਵੇਗੀ, ਨਹੀਂ ਤਾਂ ਪ੍ਰੋਗਰਾਮ ਬਾਹਰੀ if condition: ਦੇ else ਭਾਗ ਵਾਲੇ print("Number is Zero") ਸਟੇਟਮੈਂਟ ਨੂੰ ਲਾਗੂ ਕਰੇਗਾ। ਆਓ ਹੁਣ ਇਸ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰਨ ਤੋਂ ਬਾਅਦ ਇਸਦੀ ਆਊਟਪੁੱਟ ਦੇਖਦੇ ਹਾਂ:

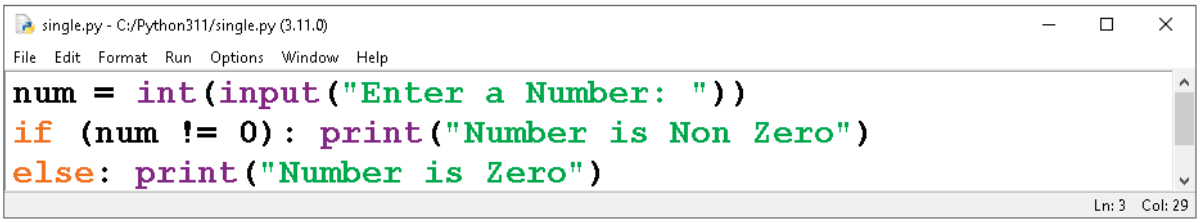


ਚਿੱਤਰ 4.23: ਪ੍ਰੋਗਰਾਮ (nestedif.py) ਦੀ ਆਊਟਪੁੱਟ

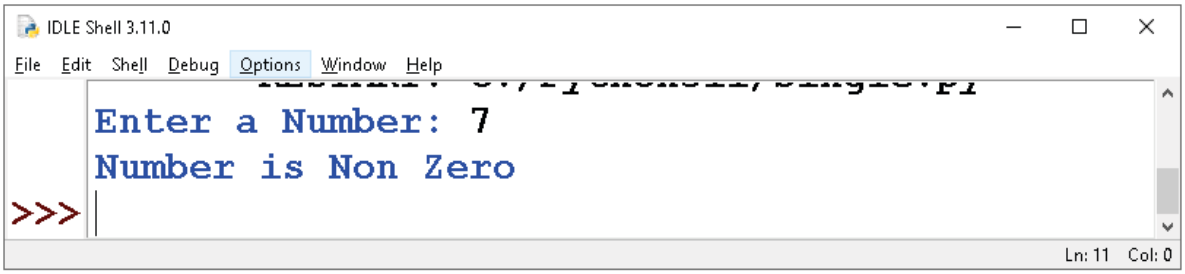
ਪ੍ਰੋਗਰਾਮ ਦੀ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਦੌਰਾਨ **num** ਵੇਰੀਏਬਲ ਦਾ ਮੁੱਲ **–2** ਇਨਪੁੱਟ ਕੀਤਾ ਗਿਆ ਹੈ। ਇਸ ਲਈ ਬਾਹਰੀ **if condition: (if -2 ! = 0)** ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ **(true)** ਆਵੇਗਾ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਅੰਦਰੂਨੀ **if condition:** ਕੋਲ ਭੇਜ ਦਿਤਾ ਜਾਵੇਗਾ। ਅੰਦਰੂਨੀ **if condition:** ਵਿੱਚ ਕੰਡੀਸ਼ਨ **(–2 > 0)** ਗਲਤ **(false)** ਹੋ ਜਾਵੇਗੀ ਜਿਸ ਕਾਰਨ ਅੰਦਰੂਨੀ **else** ਭਾਗ ਦੇ **print("Number is Negative")** ਸਟੇਟਮੈਂਟ ਨੂੰ ਚਲਾਇਆ ਜਾਵੇਗਾ।

ਨੋਟ: ਜਦੋਂ ਵੀ ਅਸੀਂ ਮਲਟੀਪਲ if ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਨਾਲ ਕੋਡ ਦਾ ਇੱਕ ਬਲਾਕ ਲਿਖਦੇ ਹਾਂ, ਤਾਂ ਇੰਡੈਂਟੇਸ਼ਨ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਭੂਮਿਕਾ ਨਿਭਾਉਂਦੀ ਹੈ।

ਮਹੱਤਵਪੂਰਨ ਨੋਟ: ਜਦੋਂ ਵੀ ਅਸੀਂ ਕੋਡ ਦੇ ਇੱਕ ਬਲਾਕ ਵਿਚ ਮਲਟੀਪਲ ਸਟੇਟਮੈਂਟਸ ਲਿਖਦੇ ਹਾਂ, ਤਾਂ ਇੰਡੈਂਟੇਸ਼ਨ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਭੂਮਿਕਾ ਨਿਭਾਂਦੀ ਹੈ। ਪਰ ਕਈ ਵਾਰ, ਅਜਿਹੀ ਸਥਿਤੀ ਹੁੰਦੀ ਹੈ ਜਿੱਥੇ ਬਲਾਕ ਵਿੱਚ ਸਿਰਫ਼ ਸਿੰਗਲ ਲਾਈਨ ਸਟੇਟਮੈਂਟ ਹੁੰਦੀ ਹੈ। ਕੋਲਨ ਦੇ ਬਾਅਦ ਇੱਕ ਬਲਾਕ ਲਿਖਣ ਦੀ ਬਜਾਏ ਅਸੀਂ ਕੋਲਨ ਦੇ ਤੁਰੰਤ ਬਾਅਦ ਵੀ ਇਹ ਸਿੰਗਲ ਸਟੇਟਮੈਂਟ ਲਿਖ ਸਕਦੇ ਹਾਂ। ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਉਦਾਹਰਨ ਦਿੱਤੀ ਗਈ ਹੈ:



ਪ੍ਰੋਗਰਾਮ (single.py): ਸਿੰਗਲ ਲਾਈਨ ਬਲਾਕ ਨਾਲ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ



ਚਿੱਤਰ 4.24: ਪ੍ਰੋਗਰਾਮ **(single.py)** ਦੀ ਆਊਟਪੁੱਟ

4.5 ਲੂਪਿੰਗ ਫਲੋਅ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ (Looping Flow Control Statements)

ਅਸੀਂ ਜਾਣਦੇ ਹਾਂ ਕਿ ਕੰਪਿਊਟਰਾਂ ਦੀ ਵਰਤੋਂ ਅਕਸਰ ਦੁਹਰਾਉਣ ਵਾਲੇ (repetitive) ਕੰਮਾਂ ਨੂੰ ਸਵੈਚਲਿਤ (automate) ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਅਜਿਹੇ ਦੁਹਰਾਉਣ ਵਾਲੇ ਕੰਮ ਲੂਪਿੰਗ ਬਣਤਰਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਪੂਰੇ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਲੂਪਿੰਗ ਬਣਤਰਾਂ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਸੈੱਟ ਨੂੰ ਦੁਹਰਾਉਣ ਦੀ ਸਹੂਲਤ ਪ੍ਰਦਾਨ ਕਰਦੀਆਂ ਹਨ। ਇਹ ਦੁਹਰਾਉਣਾ ਇੱਕ ਕੰਡੀਸ਼ਨ 'ਤੇ ਨਿਰਭਰ ਕਰਦਾ ਹੈ। ਲੂਪ ਵਿੱਚ ਸਟੇਟਮੈਂਟਾਂ ਨੂੰ ਬਾਰ ਬਾਰ ਉਸ ਸਮੇਂ ਤੱਕ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਤੱਕ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ true ਰਹਿੰਦੀ ਹੈ। ਇਸ ਕੰਡੀਸ਼ਨ ਦੀ ਜਾਂਚ ਇੱਕ ਵੇਰੀਏਬਲ ਦੇ ਮੁੱਲ ਦੇ ਅਧਾਰ ਤੇ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜਿਸਨੂੰ ਲੂਪ ਦਾ ਕੰਟਰੋਲ ਵੇਰੀਏਬਲ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਕੰਡੀਸ਼ਨ false ਹੋ ਜਾਂਦੀ ਹੈ ਤਾਂ ਲੂਪ ਬੰਦ ਹੋ ਜਾਂਦਾ ਹੈ। ਆਓ ਹੁਣ ਪਾਈਥਨ ਲੂਪਿੰਗ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਬਣਤਰਾਂ ਬਾਰੇ ਵਿਸਥਾਰ ਵਿਚ ਚਰਚਾ ਕਰੀਏ :

ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਆਈਟ੍ਰੇਟਿਵ (Ilterative) ਸਟੇਟਮੈਂਟਸ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਟੈਟਮੈਂਟਸ ਸਾਨੂੰ ਕੋਡ ਦੇ ਇੱਕ ਬਲਾਕ ਨੂੰ ਵਾਰ-ਵਾਰ (repeatedly) ਚਲਾਉਣ ਦੀ ਸਹੂਲਤ ਦਿੰਦੀਆਂ ਹਨ। ਲੂਪਿੰਗ ਸੰਕਲਪ ਨੂੰ ਬਿਹਤਰ ਤਰੀਕੇ ਨਾਲ ਸਮਝਣ ਲਈ, ਆਓ ਹੇਠਾਂ ਦਿੱਤੇ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਦੇਖਦੇ ਹਾਂ:

#Write a program to print the word "Python" five times on your screen

```
print("Python")
print("Python")
print("Python")
print("Python")
print("Python")
```

Output:

```
Python
Python
Python
Python
Python
```

ਉਪਰੋਕਤ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਕੋ ਸਟੇਟਮੈਂਟ ਨੂੰ ਪੰਜ ਵਾਰ ਲਿਖਿਆ ਗਿਆ ਹੈ (ਭਾਵ ਇਕ ਹੀ ਕੰਮ ਨੂੰ ਦੁਹਰਾਇਆ ਜਾ ਰਿਹਾ ਹੈ)। ਜੇਕਰ ਸਾਨੂੰ 1000 ਵਾਰ "Python" ਸ਼ਬਦ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਕਿਹਾ ਜਾਵੇ ਤਾਂ ਅਸੀਂ ਕੀ ਕਰਾਂਗੇ? ਪ੍ਰਿੰਟ ਸਟੇਟਮੈਂਟ ਨੂੰ 1000 ਵਾਰ ਲਿਖਣਾ ਕੋਈ ਵਧੀਆ ਹੱਲ ਨਹੀਂ ਹੋਵੇਗਾ। ਇਹ ਕੰਮ ਥਕਾਵਟ ਵਾਲਾ ਹੋਵੇਗਾ ਅਤੇ ਇਹ ਕੰਮ ਕਰਨ ਦਾ ਵਧੀਆ ਤਰੀਕਾ ਵੀ ਨਹੀਂ ਹੋਵੇਗਾ। ਅਜਿਹੇ ਕੰਮਾਂ ਲਈ ਲੂਪ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਲਿਖਣਾ ਇੱਕ ਬਿਹਤਰ ਹੱਲ ਹੈ। ਅਜਿਹੇ ਦੁਹਰਾਉਣ ਵਾਲੇ ਕੰਮ ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ਦਾ ਲਾਜ਼ਿਕ (ਤਰਕ) ਅੱਗੇ ਦਿੱਤੇ ਅਨੁਸਾਰ ਸਮਝਿਆ ਜਾ ਸਕਦਾ ਹੈ:

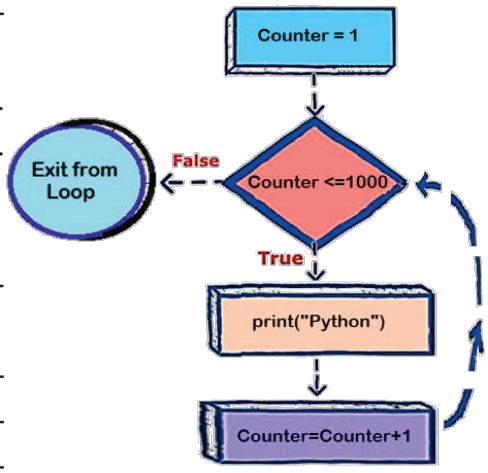
I. ਇਕ ਵੇਰੀਏਬਲ ਲਵੋ ਜਿਸਦਾਂ ਨਾਂ **counter** ਹੋਵੇ ਅਤੇ ਇਸਦਾ ਮੁੱਲ 1 ਸੈਟ ਕਰੋ (ਭਾਵ: counter=1)

ii. ਸਟੈਪ iii ਅਤੇ iv ਨੂੰ ਉਸ ਸਮੇਂ ਤੱਕ **ਦੁਹਰਾਓ** ਜਦੋਂ ਤੱਕ counter ਦਾ ਮੁੱਲ 1000 ਜਾਂ ਇਸਤੋਂ ਘੱਟ ਹੋਵੇ (ਭਾਵ: counter <=1000)

iii. "**Python**" ਮੁੱਲ ਪ੍ਰਿੰਟ ਕਰੋ (ਭਾਵ: print("Python"))

iv. Counter ਵੇਰੀਏਬਲ ਦੇ ਮੁੱਲ ਵਿਚ 1 ਅੰਕ ਦਾ **ਵਾਧਾ** ਕਰੋ (counter+=1)

ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਲੂਪਿੰਗ ਬਣਤਰਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਉਪਰੋਕਤ ਦਿੱਤੇ ਲਾਜ਼ਿਕ (ਤਰਕ) ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ, ਜਿਸ ਨਾਲ ਲੂਪ ਵਾਲੇ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਦੁਹਰਾਉਣ ਵਾਲੇ ਕੰਮਾਂ ਨੂੰ ਬਹੁਤ ਆਸਾਨੀ ਨਾਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।



ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟਸ ਨੂੰ **ਆਈਟ੍ਰੇਟਿਵ (Iterative) ਸਟੇਟਮੈਂਟਸ** ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਟੇਟਮੈਂਟਸ ਸਾਨੂੰ ਕੋਡ ਦੇ ਇੱਕ ਬਲਾਕ ਨੂੰ ਵਾਰ-ਵਾਰ (repeatedly) ਚਲਾਉਣ ਦੀ ਸਹੂਲਤ ਦਿੰਦੀਆਂ ਹਨ।

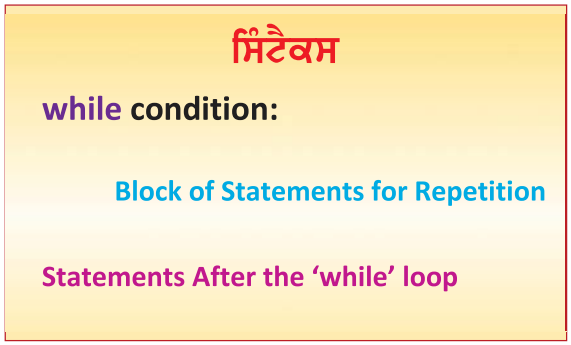
ਪਾਈਥਨ ਸਾਨੂੰ ਕਿਸੇ ਕੰਮ ਨੂੰ ਵਾਰ-ਵਾਰ ਕਰਵਾਉਣ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਦੋ ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ:

1. while ਲੂਪ (ਕੰਡੀਸ਼ਨਲ ਲੂਪ (condition loop))
2. for ਲੂਪ (ਕਾਊਂਟਿੰਗ ਲੂਪ (counting loop))

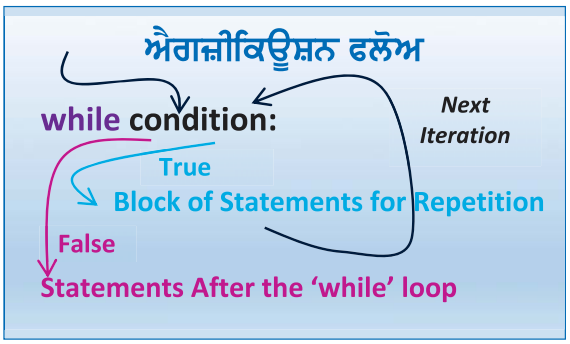
4.5.1 while ਲੂਪ (while loop):

while ਲੂਪ ਨੂੰ ਕੰਡੀਸ਼ਨਲ (Conditional) ਲੂਪ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ, ਕਿਉਂਕਿ ਇਸ ਲੂਪ ਵਿੱਚ ਹਰ ਵਾਰ ਲੂਪ ਦੇ ਸ਼ੁਰੂ ਵਿੱਚ ਇੱਕ ਕੰਡੀਸ਼ਨ ਦੀ ਜਾਂਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਅਤੇ ਜੇਕਰ ਇਹ ਕੰਡੀਸ਼ਨ ਸਹੀ (true) ਹੋਵੇ, ਤਾਂ ਲੂਪ ਦੀ ਬਾਡੀ ਨੂੰ ਚਲਾਇਆ ਜਾਵੇਗਾ; ਜਦੋਂ ਕੰਡੀਸ਼ਨ ਗਲਤ (false) ਹੋ ਜਾਂਦੀ ਹੈ, ਤਾਂ ਲੂਪ ਦੀ ਬਾਡੀ ਨੂੰ ਚਲਾਏ ਬਿਨ੍ਹਾਂ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਲੂਪ ਤੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ। ਕਿਉਂਕਿ while ਲੂਪ ਵਿੱਚ ਦਾਖਲ ਹੋਣ ਤੋਂ ਪਹਿਲਾਂ ਕੰਡੀਸ਼ਨ ਦੀ ਜਾਂਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਇਸਲਈ ਜੇਕਰ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸ਼ੁਰੂ ਵਿਚ ਹੀ ਗਲਤ (false) ਹੋ ਜਾਂਦਾ ਹੈ ਤਾਂ while ਲੂਪ ਇੱਕ ਵਾਰ ਵੀ ਨਹੀਂ ਚੱਲੇਗਾ।

ਸਧਾਰਨ ਸ਼ਬਦਾਂ ਵਿੱਚ, ਅਸੀਂ ਕਹਿ ਸਕਦੇ ਹਾਂ ਕਿ ਜਦੋਂ ਤੱਕ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ ਸੰਤੁਸ਼ਟ (satisfied) ਰਹਿੰਦੀ ਹੈ ਉਦੋਂ ਤੱਕ ਲੂਪ ਦੀ ਵਰਤੋਂ ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਬਲਾਕ ਨੂੰ ਵਾਰ-ਵਾਰ ਚਲਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਅੱਗੇ ਦਿੱਤੇ ਚਿੱਤਰਾਂ ਵਿੱਚ while ਲੂਪ ਦੇ ਸਿੰਟੈਕਸ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਨੂੰ ਦਰਸਾ ਰਹੇ ਹਨ:

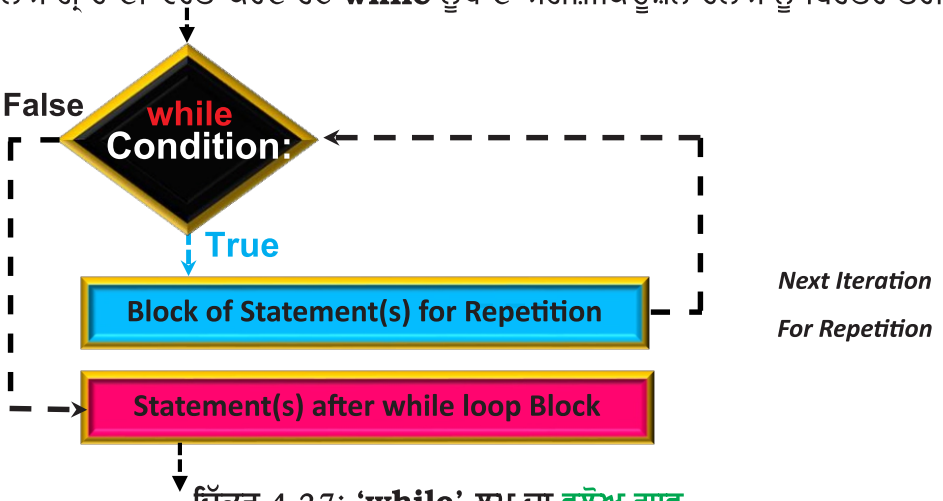


ਚਿੱਚਰ 4.25: **'while'** ਲੂਪ ਦਾ ਸਿੰਟੈਕਸ



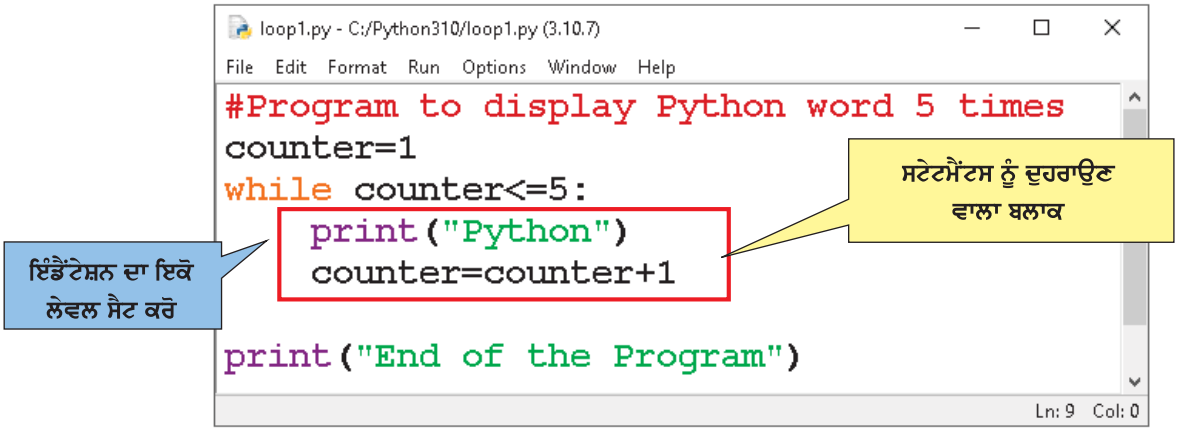
ਚਿੱਤਰ 4.26: **'while'** ਲੂਪ ਦਾ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ

ਅਸੀਂ ਅੱਗੇ ਦਿਤੇ ਫਲੋਅ ਗ੍ਰਾਫ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ **while** ਲੂਪ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਨੂੰ ਬਿਹਤਰ ਤਰੀਕੇ ਨਾਲ ਸਮਝ ਸਕਦੇ ਹਾਂ:



ਚਿੱਤਰ 4.27: **'while'** ਲੂਪ ਦਾ ਫਲੋਅ ਗ੍ਰਾਫ

ਆਓ ਹੁਣ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ ਜਿੱਥੇ ਅਸੀਂ **while** ਲੂਪ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ 5 ਵਾਰ "Python" ਸ਼ਬਦ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ:



ਪ੍ਰੋਗਰਾਮ (loop1.py): while ਲੂਪ ਦੀ ਵਰਤੋਂ ਨਾਲ "Python" ਸ਼ਬਦ ਨੂੰ 5 ਵਾਰ ਦਰਸਾਉਣਾ

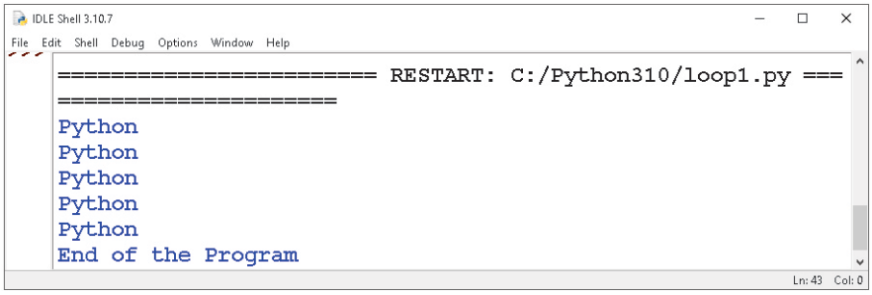
ਪ੍ਰੋਗਰਾਮ loop1.py ਵਿਚ while ਲੂਪ ਤੋਂ ਪਹਿਲਾਂ ਅਸੀਂ ਵੇਰੀਏਬਲ counter ਦਾ ਮੁੱਲ ਸੈਟ ਕੀਤਾ ਹੈ। ਇੱਥੇ counter ਨੂੰ ਕਾਊਂਟਰ ਵੇਰੀਏਬਲ ਦੇ ਤੌਰ ਤੇ ਵਰਤਿਆ ਗਿਆ ਹੈ ਜੋ while ਲੂਪ ਦੀਆਂ ਆਈਟ੍ਰੇਸ਼ਨਜ਼ ਨੂੰ ਕੰਟਰੋਲ ਕਰੇਗਾ। ਇਹ while ਲੂਪ ਫਲੋਅ ਗ੍ਰਾਫ ਵਿਚ ਦਿਖਾਏ ਅਨੁਸਾਰ ਕੰਮ ਕਰੇਗਾ।

ਲੂਪ ਦੀ ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਦੋ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਉਸ ਸਮੇਂ ਤੱਕ ਵਾਰ-ਵਾਰ ਦੁਹਰਾਇਆ ਜਾਵੇਗਾ ਜਦੋਂ ਤੱਕ ਦਿੱਤਾ ਗਿਆ ਕੰਡੀਸ਼ਨ (count <=5) ਸੰਤੁਸ਼ਤ ਰਹੇਗਾ, ਭਾਵ ਜਦੋਂ ਤੱਕ ਕੰਡੀਸ਼ਨ true ਹੋਵੇਗਾ:

I. print("Python") (ਲੂਪ ਦੀ ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ Python ਸ਼ਬਦ ਸਕ੍ਰੀਨ ਉਪਰ ਪ੍ਰਿੰਟ ਹੁੰਦਾ ਰਹੇਗਾ)

ii. counter = counter + 1 (ਲੂਪ ਦੀ ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ ਕਾਊਂਟਰ ਵੇਰੀਏਬਲ 'counter' ਦੇ ਮੁੱਲ ਵਿੱਚ 1 ਅੰਕ ਦਾ ਵਾਧਾ ਹੁੰਦਾ ਰਹੇਗਾ, ਭਾਵ: counter = 1+1, counter = 2+1, counter = 3+1, counter = 4+1, coutner = 5+1)

ਜਦੋਂ ਵੇਰੀਏਬਲ 'counter' ਦਾ ਮੁੱਲ 6 ਹੋ ਜਾਵੇਗਾ, ਲੂਪ ਦੀ ਕੰਡੀਸ਼ਨ **(6 <=5) false** ਹੋ ਜਾਵੇਗੀ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਲੂਪ ਤੋਂ ਬਾਹਰ ਆ ਜਾਵੇਗਾ। ਹੁਣ ਪ੍ਰੋਗਰਾਮ **print("End of the Program")** ਸਟੇਟਮੈਂਟ ਜੋ while ਲੂਪ ਦੇ ਬਲਾਕ ਤੋਂ ਬਾਅਦ ਲਿਖੀ ਗਈ ਹੈ, ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰੇਗਾ। ਆਓ ਹੁਣ ਕੀਬੋਰਡ ਤੋਂ **F5** ਕੀਅ ਪ੍ਰੈਸ ਕਰਕੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰਦੇ ਹਾਂ। ਇਹ ਅੱਗੇ ਦਰਸਾਏ ਅਨੁਸਾਰ ਆਊਟਪੁੱਟ ਦਿਖਾਵੇਗਾ:



ਚਿੱਤਰ 4.28: ਪ੍ਰੋਗਰਾਮ **(loop1.py)** ਦੀ ਆਊਟਪੁੱਟ

ਜੇਕਰ ਅਸੀਂ **Python** ਸ਼ਬਦ ਨੂੰ **1000** ਵਾਰ ਪ੍ਰਿੰਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਸਾਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਕ ਛੋਟਾ ਜਿਹਾ ਬਦਲਾਵ ਕਰਨਾ ਪਵੇਗਾ; ਸਾਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ **while** ਲੂਪ ਦੇ ਕੰਡੀਸ਼ਨ ਭਾਗ ਵਿਚ 5 ਮੁੱਲ ਦੀ ਜਗ੍ਹਾ 1000 (ਜਾਂ ਕੋਈ ਵੀ ਹੋਰ ਮੁੱਲ ਜੋ ਅਸੀਂ ਲਿਖਣਾ ਚਾਹੁੰਦ ਹਾਂ) ਲਿਖਣਾ ਪਵੇਗਾ।

TEST YOURSELF

ਪਾਈਥਲ ਦੇ ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਵਿੱਚ ਗਲਤੀਆਂ ਲੱਭੋ ਅਤੇ ਸਹੀ ਕੋਡ ਲਿਖੋ:

4.5

```
i=1
while(i<=10)
print("Python")
i+=1
```

```
i=10
while(i<=15):
        print("Hello Python")
```

```
i=
while(i<=15)
        print(Hello)
        print("Python")
        i=i+1
```

ਪਾਇਥਨ ਕੋਡਜ਼ ਦਾ ਬਲਾਕ ਬਨਾਉਣ ਲਈ ਇੰਡੈਂਟੇਸ਼ਨ (indentation) ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਸਟੇਟਮੈਂਟ ਦੇ ਸ਼ੁਰੂ ਵਿੱਚ ਛੱਡੀ ਜਾਣ ਵਾਲੀ ਖਾਲੀ ਥਾਂ (ਸਪਸੇ ਅਤੇ ਟੈਬਜ਼) ਨੂੰ ਇੰਡੈਂਟੇਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਪਾਈਥਨ ਕੋਡ ਵਿੱਚ ਇੰਡੈਂਟੇਸ਼ਨ ਦਾ ਇੱਕੋ ਲੇਵਲ (Same Level of Indentation) ਸਟੇਟਮੈਂਟਾਂ ਦੇ ਇੱਕ ਸਿੰਗਲ ਬਲਾਕ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇਸ ਲਈ while ਲੂਪ ਦਾ ਬਲਾਕ ਬਣਾਉਣ ਲਈ ਇੰਡੈਂਟੇਸ਼ਨ ਸੈਟ ਕਰਨਾ ਨਾ ਭੁੱਲੋ।

ਆਓ ਹੁਣ ਇਕ ਹੋਰ ਉਦਾਹਰਣ ਦੇਖਦੇ ਹਾਂ ਜਿਸ ਵਿਚ ਅਸੀਂ ਲੂਪ ਦੀ ਮਦਦ ਨਾਲ ਪਹਿਲੇ 10 ਕੁਦਰਤੀ ਨੰਬਰਾਂ (Natural Numbers) ਦਾ ਜੋੜ ਕੈਲਕੁਲੇਟ ਕਰਾਂਗੇ:

```
#Program to calculate sum of first 10 natural numbers
sum=0
i=1
while i<=10:      #block of while loop begins
    sum=sum+i
    i=i+1
                  #End of while loop
print("Sum is: ", sum)
```

ਦੁਹਰਾਈਆਂ ਜਾਣ ਵਾਲੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦਾ ਬਲਾਕ

ਪ੍ਰੋਗਰਾਮ **(loop2.py): while** ਲੂਪ ਦੀ ਵਰਤੋਂ ਨਾਲ ਪਹਿਲੇ 10 ਕੁਦਰਤੀ ਅੰਕਾਂ ਦਾ ਜੋੜ ਪਤਾ ਕਰਨ ਲਈ

ਉਪਰੋਕਤ ਪ੍ਰੋਗਰਾਮ ਵਿਚ while ਲੂਪ ਤੋਂ ਪਹਿਲਾਂ ਅਸੀਂ ਦੋ ਵੇਰੀਏਬਲਾਂ: 'sum' ਅਤੇ 'i' ਦਾ ਮੁੱਲ ਲੜੀਵਾਰ 0 ਅਤੇ 1 ਸੈੱਟ ਕੀਤਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਵਿਚ 'sum' ਵੇਰੀਏਬਲ ਦੀ ਵਰਤੋਂ while ਲੂਪ ਵਿਚ ਪਹਿਲੇ 10 ਕੁਦਰਤੀ ਨੰਬਰਾਂ ਦਾ ਜੋੜ ਕੈਲਕੁਲੇਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਵੇਗੀ ਅਤੇ ਵੇਰੀਏਬਲ 'i' ਨੂੰ ਕਾਊਂਟਰ ਵੇਰੀਏਬਲ ਵੱਜੋਂ ਵਰਤਿਆ ਜਾਵੇਗਾ ਜੋ while ਲੂਪ ਦੀਆਂ ਆਈਟਰੇਸ਼ਨਾਂ ਨੂੰ ਕੰਟਰੋਲ ਕਰੇਗਾ।

while ਲੂਪ ਦੀ ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ, ਲੂਪ ਵਿਚਲੀਆਂ ਦੋ ਸਟੇਟਮੈਂਟਸ ਦਾ ਉਸ ਸਮੇਂ ਤੱਕ ਦੁਹਰਾਓ ਹੁੰਦਾ ਰਹੇਗਾ ਜਦੋਂ ਤੱਕ ਦਿੱਤੀ ਗਈ ਕੰਡੀਸ਼ਨ (i <= 10) ਗਲਤ (False) ਨਹੀਂ ਹੋ ਜਾਂਦੀ।

i. sum = sum + i (ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ sum ਵੇਰੀਬਲ ਦੇ ਪਿਛਲੇ ਮੁੱਲ ਵਿਚ i ਵਰੀਏਬਲ ਦਾ ਮੌਜੂਦ ਮੁੱਲ ਜੋੜ ਕੇ sum ਦਾ ਮੁੱਲ ਅਪਡੇਟ ਕੀਤਾ ਜਾਂਦਾ ਰਹੇਗਾ, ਭਾਵ: sum=0+1, sum=1+2, sum=3+3, sum=6+4, ... sum=45+10)

ii. i = i + 1 (ਕਾਊਂਟਰ ਵੇਰੀਏਬਲ 'i' ਦੇ ਮੁੱਲ ਵਿੱਚ 1 ਅੰਕ ਦਾ ਵਾਧਾ ਹੁੰਦਾ ਰਹੇਗਾ, ਭਾਵ: i=1+1, i=2+1, i=3+1, i=4+1, ..., i=10+1)

ਜਦੋਂ ਵੇਰੀਏਬਲ i ਦਾ ਮੁੱਲ 11 ਹੋ ਜਾਵੇਗਾ, ਲੂਪ ਦੀ ਕੰਡੀਸ਼ਨ **(11 <= 10) false** ਹੋ ਜਾਵੇਗੀ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਲੂਪ ਤੋਂ ਬਾਹਰ ਆ ਜਾਵੇਗਾ। ਹੁਣ ਪ੍ਰੋਗਰਾਮ **print("Sum is:", sum)** ਸਟੇਟਮੈਂਟ ਜੋ while ਲੂਪ ਦੇ ਬਲਾਕ ਤੋਂ ਬਾਅਦ ਲਿਖੀ ਗਈ ਹੈ, ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰੇਗਾ। ਆਓ ਹੁਣ ਕੀਬੋਰਡ ਤੋਂ **F5** ਕੀਅ ਪ੍ਰੈਸ ਕਰਕੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰਦੇ ਹਾਂ। ਇਹ ਹੇਠਾਂ ਦਰਸਾਏ ਅਨੁਸਾਰ ਆਊਟਪੁੱਟ ਦਿਖਾਵੇਗਾ:

```
Sum is:  55
```

ਜੇਕਰ ਉਪਰੋਕਤ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਅਸੀਂ ਪਹਿਲੇ 100 ਕੁਦਰਤੀ ਨੰਬਰਾਂ ਦਾ ਜੋੜ ਪਤਾ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ **while** ਲੂਪ ਦੀ ਕੰਡੀਸ਼ਨ ਵਿਚ ਮੁੱਲ 10 ਨੂੰ 100 ਨਾਲ (ਜਾਂ ਜਿਹੜੀ ਵੀ ਸੰਖਿਆ ਤੱਕ ਤੁਸੀਂ ਜੋੜ ਚਾਹੁੰਦੇ ਹੋ) ਬਦਲ ਦਵੋ।

TEST YOURSELF

ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਦੀ ਆਊਟਪੁੱਟ ਲਿਖੋ:

4.6

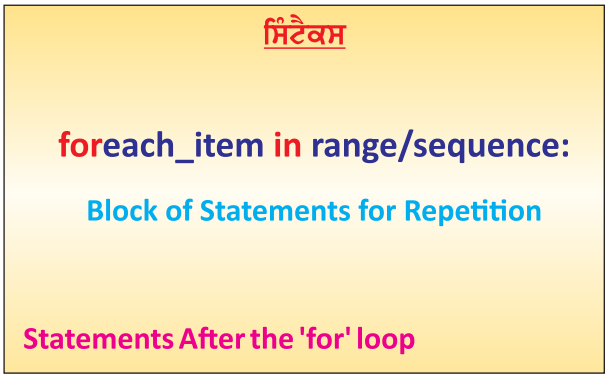
```
i=4
while(i>=1):
        print(i)
        i=i  - 1
```

```
i=1
while(i<=10):
        print(i*7     , end=" "    )
        i+=1
```

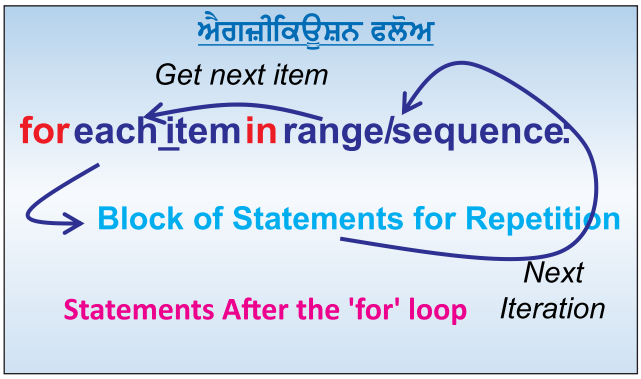
while ਲੂਪ ਦੇ ਬਾਡੀ ਅੰਦਰ ਅਜਿਹੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦਾ ਹੋਣਾ ਤੈਅ ਕਰੋ ਜਿਸ ਨਾਲ ਲੂਪ ਦੀ ਕੰਡੀਸ਼ਨ ਆਖਰਕਾਰ **false** ਹੋ ਜਾਵੇ: ਨਹੀਂ ਤਾਂ ਲੂਪ ਅਨੰਤ ਲੂਪ **(Infinite Loop)** ਬਣ ਜਾਵੇਗਾ, ਜਿਸ ਨਾਲ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਲਾਜ਼ੀਕਲ ਐਰਰ **Logical Error)** ਆ ਜਾਵੇਗੀ। ਅਨੰਤ ਲੂਪ ਵਿਚੋਂ ਬਾਹਰ ਆਉਣ ਲਈ ਕੀਬੋਰਡ ਤੋਂ **Ctrl+C** ਪ੍ਰੈਸ ਕਰਨਾ ਪਵੇਗਾ।

4.5.2 for ਲੂਪ (for loop):

ਪਾਈਥਨ ਵਿੱਚ for ਲੂਪ ਦੀ ਵਕਤੋਂ ਆਮ ਤੌਰ 'ਤੇ ਮੁੱਲਾਂ ਦੀ ਇੱਕ ਰੇਂਜ਼ ਜਾਂ ਸਿਕੁਐਂਸ (Range of Values of Sequence) (ਜਿਵੇਂ ਕਿ: ਸਟ੍ਰਿੰਗ, ਲਿਸਟ, ਟਪਲ, ਡਿਕਸ਼ਨਰੀ, ਜਾਂ ਸੈੱਟ) ਉੱਪਰ ਆਈਟ੍ਰੇਸ਼ਨਜ਼ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ਰੇਂਜ ਜਾਂ ਸਿਕੁਐਂਸ ਦੀ ਹਰੇਕ ਆਈਟਮ ਲਈ ਇੱਕ-ਇੱਕ ਵਾਰ ਲੂਪ ਦੀਆਂ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਚਲਾ (Execute Statements Once for Each Item) ਸਕਦੇ ਹਾਂ। for ਲੂਪ ਦਾ ਸਿੰਟੈਕਸ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ ਹੇਠਾਂ ਦਿੱਤੇ ਚਿੱਤਰਾਂ ਵਿੱਚ ਦੇਖਿਆ ਜਾ ਸਕਦਾ ਹੈ:

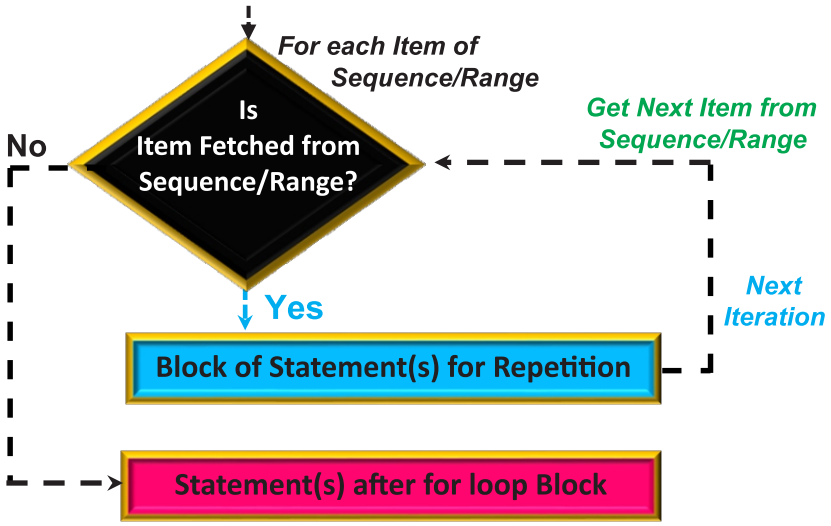


ਚਿੱਤਰ 4.29: **'for'** ਲੂਪ ਦਾ ਸਿੰਟੈਕਸ



ਚਿੱਤਰ 4.30: **'for'** ਲੂਪ ਦਾ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋਅ

ਹੇਠਾਂ ਦਿਤਾ ਚਿੱਤਰ ਲੂਪ ਦੇ ਐਗਜ਼ਿਕਿਊਸ਼ਨ-ਫਲੋਅ ਨੂੰ ਫਲੋਅ-ਗ੍ਰਾਫ ਦੇ ਰੂਪ ਵਿਚ ਦਰਸਾ ਰਿਹਾ ਹੈ:



ਚਿੱਤਰ 4.31: **'for'** ਲੂਪ ਦਾ ਫਲੋਅ ਗ੍ਰਾਫ

ਲੂਪ ਦੇ ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ, **each_item** ਵੇਰੀਏਬਲ ਜਾਂਚ ਕਰਦਾ ਹੈ ਕਿ ਰੇਂਜ/ਸਿਕੁਐਂਸ ਵਿੱਚ ਮੌਜੂਦ ਹਰੇਕ ਮੁੱਲ ਦੀ ਵਰਤੋਂ (traverse) ਕਰ ਲਈ ਹੈ ਜਾਂ ਨਹੀਂ। ਜਦੋਂ ਰੇਂਜ/ਸਿਕੁਐਂਸ ਦੀਆਂ ਸਾਰੀਆਂ ਆਈਟਮਾਂ ਨੂੰ ਵਰਤ (traverse) ਲਿਆ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਲੂਪ ਦੀਆਂ ਸਟੇਟਮੈਂਟਸ ਦੇ ਬਲਾਕ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕਰਨਾ ਬੰਦ ਕਰ ਦਿਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਨੂੰ for ਲੂਪ ਤੋਂ ਬਾਹਰ ਲੂਪ ਤੋਂ ਤੁਰੰਤ ਬਾਅਦ ਵਾਲੀ ਸਟੇਟਮੈਂਟ ਉਪਰ ਟ੍ਰਾਂਸਫਰ ਕਰ ਦਿਤਾ ਜਾਂਦਾ ਹੈ।

for ਲੂਪ ਨੂੰ **ਕਾਉਂਟਿੰਗ ਲੂਪ (Counting Loop)** ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ, ਕਿਉਂਕਿ ਇਹ ਲੂਪ ਆਮ ਤੌਰ 'ਤੇ ਇੱਕ ਨਿਸ਼ਚਿਤ ਸੰਖਿਆ ਤੱਕ (fixed number of times) ਕਿਸੇ ਕੰਮ ਨੂੰ ਦੁਹਰਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਲਈ ਜਦੋਂ ਵੀ ਅਸੀਂ for ਲੂਪ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਹੋਵੇ ਤਾਂ ਸਾਨੂੰ ਪਹਿਲਾਂ ਇਹ ਜਾਣਨ ਦੀ ਜ਼ਰੂਰਤ ਹੋਵੇਗੀ ਕਿ ਕੋਡ ਨੂੰ ਕਿੰਨੀ ਵਾਰ ਦੁਹਰਾਉਣਾ ਹੈ, ਉਦਾਹਰਨ ਲਈ: 5 ਵਾਰ, 10 ਵਾਰ, n ਵਾਰ ਆਦਿ।

ਜਦੋਂ ਅਸੀਂ for ਲੂਪ ਨੂੰ ਕਿਸੇ ਖਾਸ ਸੰਖਿਆ ਤੱਕ ਚਲਾਉਣਾ ਹੋਵੇ ਤਾਂ **range( )** ਫੰਕਸ਼ਨ **for** ਲੂਪ ਚਲਾਉਣ ਵਿਚ ਅਹਿਮ ਭੂਮਿਕਾ ਨਿਭਾਉਂਦਾ ਹੈ। ਅਸੀਂ ਚੁਕਵੀਆਂ ਉਦਾਹਰਣਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ **range( )** ਫੰਕਸ਼ਨ ਸਬੰਧੀ ਪਿਛਲੇ ਪਾਠਾਂ ਵਿਚ ਪੜ੍ਹ ਚੁੱਕੇ ਹਾਂ। **range( )** ਫੰਕਸ਼ਨ ਦਾ ਸਿੰਟੈਕਸ ਇਸ ਪ੍ਰਕਾਰ ਹੈ:

```
range(lower_limit, upper_limit, step)
```

ਇਹ ਫੰਕਸ਼ਨ **lower_limit** ਤੋਂ **upper_limit-1** ਤੱਕ ਮੁੱਲਾਂ ਦਾ ਇਕ ਸੈੱਟ ਜੈਨਰੇਟ ਕਰਦਾ ਹੈ।

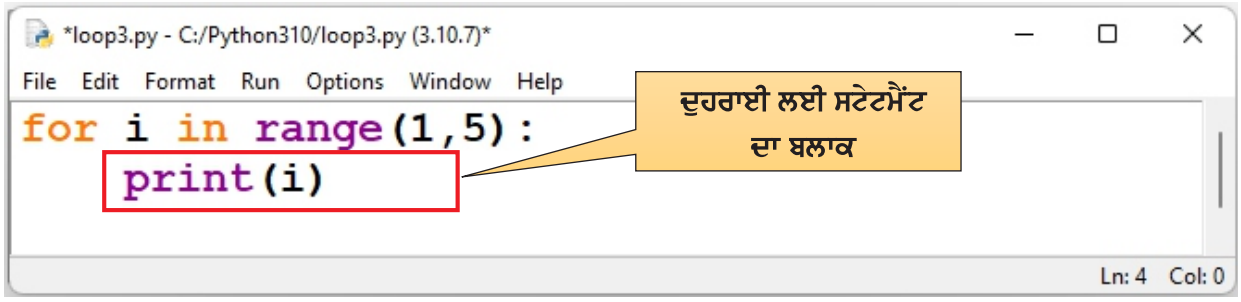
range( ) ਫੰਕਸ਼ਨ ਦੇ ਸਾਰੇ ਪੈਰਾਮੀਟਰਜ਼ (i.e. lower_limit, upper_limit, and step) ਇੰਟੀਜ਼ਰ (integers_ ਫਾਰਮੇਟ ਵਿਚ ਹੋਣੇ ਜ਼ਰੂਰੀ ਹਨ।

- **lower_limit** ਇੱਕ ਆਪਸ਼ਨਲ ਪੈਰਾਮੀਟਰ ਹੈ ਜੋ ਕ੍ਰਮ ਦੇ ਸ਼ੁਰੂਆਤੀ ਮੁੱਲ ਨੂੰ ਨਿਸ਼ਚਿਤ ਕਰਦਾ ਹੈ, ਇਸਦਾ ਡਿਫਾਲਟ ਮੁੱਲ 0 ਹੁੰਦਾ ਹੈ, ਭਾਵ ਜੇਕਰ lower_limit ਪੈਰਾਮੀਟਰ ਦਾ ਮੁੱਲ **range( )** ਫੰਕਸ਼ਨ ਨੂੰ ਪ੍ਰਦਾਨ ਨਹੀਂ ਕੀਤੀ ਜਾਂਦਾ, ਤਾਂ ਇਸਨੂੰ ਜ਼ੀਰੋ (0) ਮੰਨਿਆ ਜਾਵੇਗਾ।
- **upper_limit** ਪੈਰਾਮੀਟਰ ਦਾ ਮੁੱਲ **range( )** ਫੰਕਸ਼ਨ ਵਿੱਚ ਪਾਸ ਕਰਨਾ ਲਾਜ਼ਮੀ ਹੁੰਦਾ ਹੈ ਜੋ ਕ੍ਰਮ ਦੇ ਅੰਤਿਮ ਮੁੱਲ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ (upper_limit ਮੁੱਲ ਕ੍ਰਮ ਵਿੱਚ ਸ਼ਾਮਲ ਨਹੀਂ ਹੁੰਦਾ)
- **step** ਇੱਕ ਆਪਸ਼ਨਲ ਪੈਰਾਮੀਟਰ ਹੈ ਜੋ **range( )** ਫੰਕਸ਼ਨ ਦੁਆਰਾ ਤਿਆਰ ਕੀਤੇ ਕ੍ਰਮ ਲਈ ਵਾਧੇ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦਾ ਹੈ। ਇਸਦਾ ਡਿਫਾਲਟ ਮੁੱਲ 1 ਹੁੰਦਾ ਹੈ। step ਪੈਰਾਮੀਟਰ ਜ਼ੀਰੋ ਨੂੰ ਛੱਡ ਕੇ ਕੋਈ ਵੀ ਪਾਜੀਟਿਵ ਜਾਂ ਨੈਗੇਟਿਵ ਇੰਟੀਜ਼ਰ ਹੋ ਸਕਦਾ ਹੈ।

ਉਦਾਹਰਣ ਲਈ:

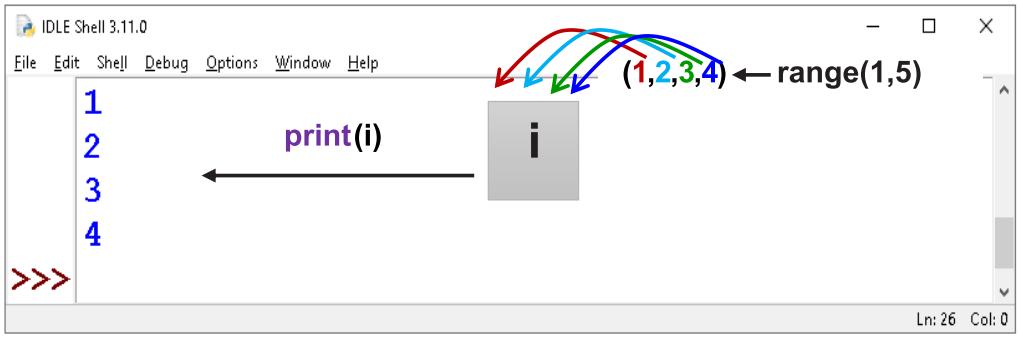
- range(5) ਇਹ ਮੁੱਲਾਂ ਦਾ ਕ੍ਰਮ (0,1,2,3,4) ਜੈਨਰੇਟ ਕਰੇਗਾ।
- range(3, 7) ਇਹ ਮੁੱਲਾਂ ਦਾ ਕ੍ਰਮ (3,4,5,6) ਜੈਨਰੇਟ ਕਰੇਗਾ।
- range(2, 10, 2) ਇਹ ਮੁੱਲਾਂ ਦਾ ਕ੍ਰਮ (2,4,6,8) ਜੈਨਰੇਟ ਕਰੇਗਾ।

ਆਓ ਹੁਣ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ 'ਤੇ ਵਿਚਾਰ ਕਰੀਏ ਜਿੱਥੇ ਅਸੀਂ ਨੰਬਰ 1, 2, 3 ਅਤੇ 4 ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਵਾ ਰਹੇ ਹਾਂ:



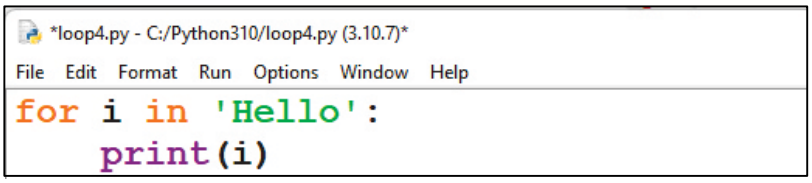
ਪ੍ਰੋਗਰਾਮ (loop3.py): range() ਫੰਕਸ਼ਨ ਨਾਲ for ਲੂਪ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ for ਲੂਪ ਫਲੋਅ-ਗ੍ਰਾਫ ਵਿਚ ਦਰਸਾਏ ਐਗਜ਼ੀਕਿਊਸ਼ਨ-ਫਲੋਅ ਅਨੁਸਾਰ ਕੰਮ ਕਰੇਗਾ। ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ **range(1,5)** ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਗਈ ਹੈ ਜੋ **lower_limit(1)** ਤੋਂ **upper_limit-1 (i.e. 5-1=4)** ਤੱਕ ਅੰਕਾਂ ਦਾ ਇਕ ਕ੍ਰਮ: 1,2,3,4 ਜੈਨਰੇਟ ਕਰੇਗਾ (ਇਸ ਵਿਚ ਡਿਫਾਲਟ ਇੰਕਰੀਮੈਂਟ ਮੁੱਲ 1 ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਵੇਗੀ); **range()** ਫੰਕਸ਼ਨ ਦੁਆਰਾ ਜੈਨਰੇਟ ਕੀਤੇ ਗਏ ਕ੍ਰਮ ਦੀ ਹਰੇਕ ਸੰਖਿਆ ਨੂੰ ਲੂਪ ਦੀ ਹਰੇਕ ਆਈਟ੍ਰੇਸ਼ਨ ਦੌਰਾਨ ਇੱਕ-ਇੱਕ ਕਰਕੇ ਵੇਰੀਏਬਲ i ਵਿੱਚ ਪ੍ਰਾਪਤ ਕੀਤਾ ਜਾਵੇਗਾ; ਜਿਸਨੂੰ for ਲੂਪ ਦੇ ਬਲਾਕ ਵਿੱਚ **print(i)** ਸਟੇਟਮੈਂਟ ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾਵੇਗਾ। ਆਉ ਹੁਣ ਇਸ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਐਗਜ਼ਕਿਊਟ ਕਰਕੇ ਇਸਦਾ ਆਉਟਪੁੱਟ ਪ੍ਰਾਪਤ ਕਰਦੇ ਹਾਂ:



ਚਿੱਤਰ 4.32: ਪ੍ਰੋਗਰਾਮ (loop3.py) ਦੀ ਆਊਟਪੁੱਟ

ਆਓ ਹੁਣ for ਲੂਪ ਦੀ ਇਕ ਹੋਰ ਉਦਾਹਰਣ ਦੇਖਦੇ ਹਾਂ ਜਿਸ ਵਿਚ ਸਟ੍ਰਿੰਗ ਵਿਚਲੇ ਕਰੈਕਟਰਜ਼ ਦੀ ਲੜੀ ਨੂੰ ਆਈਟ੍ਰੇਟ ਕੀਤਾ ਗਿਆ ਹੈ:

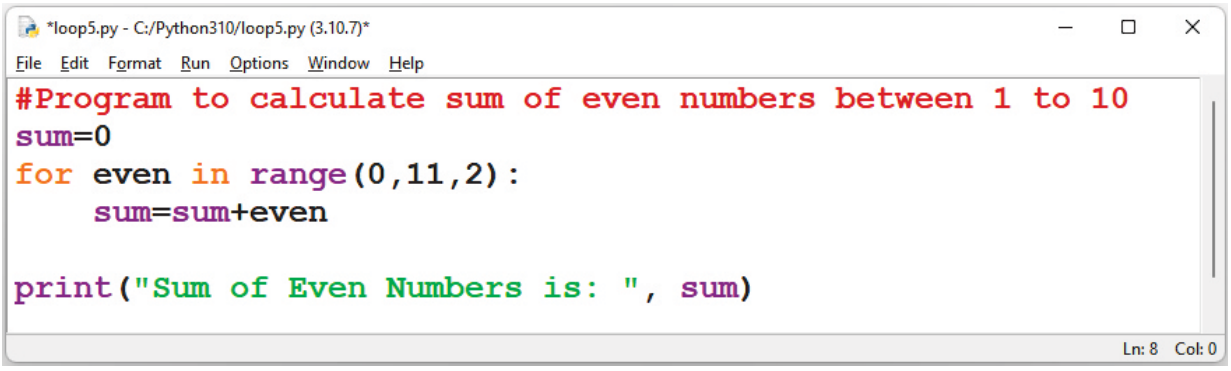


ਪ੍ਰੋਗਰਾਮ (loop4.py): ਕਰੈਕਟਰਾਂ ਦੀ ਇਕ ਲੜੀ (ਸਟ੍ਰਿੰਗ) ਨਾਲ for ਲੂਪ

ਇਹ ਪ੍ਰੋਗਰਾਮ ਫਾਰ ਲੂਪ ਦੀ ਵਰਤੋਂ ਕਰਕੇ **Hello** ਸ਼ਬਦ ਦੇ ਕਰੈਕਟਰਜ਼ ਨੂੰ ਇੱਕ-ਇੱਕ ਕਰਕੇ ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ। ਇਸ ਦਾ ਆਉਟਪੁੱਟ ਹੇਠ ਦਿਖਾਏ ਅਨੁਸਾਰ ਹੋਵੇਗਾ:

```
H
e
l
l
o
```

for ਲੂਪ ਸਬੰਧੀ ਹੁਣ ਤੱਕ ਵਿਚਾਰੀਆਂ ਗਈਆਂ ਉਦਾਹਰਨਾਂ ਬਹੁਤ ਸਰਲ ਹਨ ਜੋ ਲੂਪ ਦੇ ਮੂਲ ਦੁਹਰਾਓ ਨੂੰ ਦਰਸਾਉਂਦੀਆਂ ਹਨ। ਆਉ ਹੁਣ ਹੋਰ ਗੁੰਝਲਦਾਰ ਸਟੇਟਮੈਂਟਸ ਨਾਲ ਇੱਕ ਹੋਰ ਉਦਾਹਰਣ ਦੇਖਦੇ ਹਾਂ:



ਪ੍ਰੋਗਰਾਮ (loop5.py): 1 ਤੋਂ 10 ਦੇ ਵਿਚਕਾਰ ਮੌਜੂਦ ਸਮ (even) ਨੰਬਰਾਂ ਦਾ ਜੋੜ ਕਰਨ ਸਬੰਧੀ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਦਾ ਆਊਟਪੁੱਟ ਇਸ ਤਰ੍ਹਾਂ ਦਿਖਾਈ ਦੇਵੇਗਾ:

```
Sum of Even Numbers is:  30
```

TEST Yourself

ਪਾਈਥਨ ਦੇ ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਵਿੱਚ ਗਲਤੀਆਂ ਲੱਭੋ ਅਤੇ ਆਉਟਪੁੱਟ ਦੇ ਨਾਲ ਸਹੀ ਕੋਡ ਲਿਖੋ:

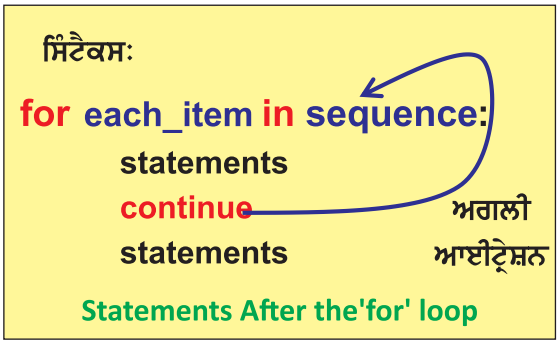
4.7

	ਸਹੀ ਕੋਡ:	ਆਊਟਪੁੱਟ:
for i in range(3) print(“ Hi ”)		
for i of range(5,2,- 1) print(i, end=” ”)		

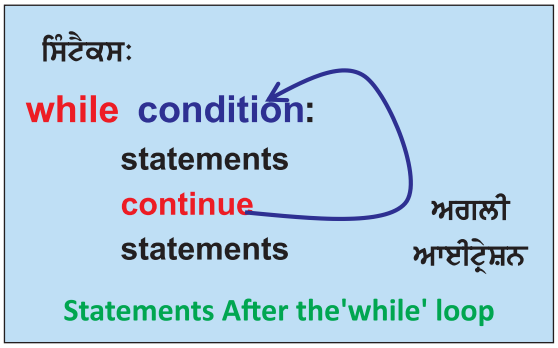
4.6 ਲੂਪ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ (Loop Control Statements)

ਪਾਈਥਨ ਸਾਨੂੰ ਕੁੱਝ ਅਜਿਹੇ ਸਟੇਟਮੈਂਟਸ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ ਜੋ ਲੂਪਸ ਦੇ ਅੰਦਰ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਫਲੋ ਨੂੰ ਕੰਟਰੋਲ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਅਜਿਹੇ ਸਟੇਟਮੈਂਟਸ ਨੂੰ **ਲੂਪ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ** ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਟੇਟਮੈਂਟਸ ਲੂਪ ਦੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਦੇ ਨਾਰਮਲ ਫਲੋਅ (normal flow) ਨੂੰ ਬਦਲਦੇ ਹਨ। ਪਾਈਥਨ ਹੇਠ ਦਿੱਤੇ ਲੂਪ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ:

- **'continue' ਸਟੇਟਮੈਂਟ ('continue Statement):** ਪਾਈਥਨ ਵਿੱਚ ਇਹ ਸਟੇਟਮੈਂਟ ਵਰਤਮਾਨ ਦੁਹਰਾਅ (current iteration) ਨੂੰ ਛੱਡਣ (skip) ਅਤੇ ਅਗਲੇ ਦੁਹਰਾਓ (next iteration) ਨਾਲ ਲੂਪ ਜਾਰੀ (continue) ਰੱਖਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੇ ਸਿੰਟੈਕਸ ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ ਕਿ ਕਿਸ ਤਰ੍ਹਾਂ ਲੂਪਸ ਵਿਚ **continue** ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੰਟਰੋਲ ਟ੍ਰਾਂਸਫਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ:

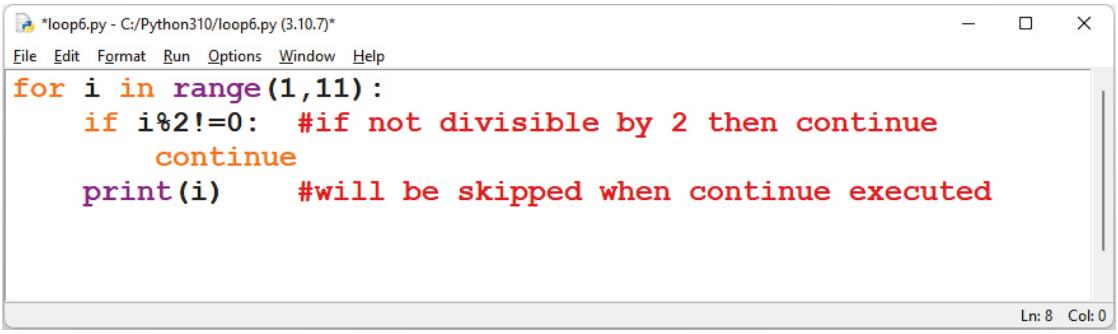


ਚਿੱਤਰ 4.33: **'for' ਲੂਪ ਵਿਚ continue**



ਚਿੱਤਰ 4.34: **'while' ਲੂਪ ਵਿਚ continue**

ਹੇਠਾਂ ਦਿੱਤਾ ਪ੍ਰੋਗਰਾਮ ਲੂਪ ਵਿੱਚ continue ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾ ਰਿਹਾ ਹੈ:



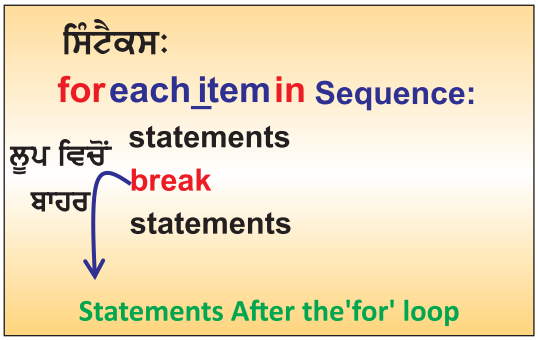
ਪ੍ਰੋਗਰਾਮ (loop6.py): ਲੂਪ ਵਿਚ continue ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ for ਲੂਪ ਨੂੰ 1 ਤੋਂ 10 ਤੱਕ ਨੰਬਰਾਂ ਦੇ ਕ੍ਰਮ, ਜੋ ਕਿ range () ਫੰਕਸ਼ਨ ਦੁਆਰਾ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਹੈ, ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਚਲਾਇਆ ਜਾ ਰਿਹਾ ਹੈ। ਲੂਪ ਵਿੱਚ ਇੱਕ ਕੰਡੀਸ਼ਨ (if i%2!=0) ਦੀ ਵਰਤੋਂ ਇਹ ਜਾਂਚ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾ ਰਹੀ ਹੈ ਕਿ ਕੀ i ਦਾ ਮੁੱਲ 2 ਨਾਲ ਵੰਡਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਾਂ ਨਹੀਂ। ਜੇਕਰ i ਦਾ ਮੁੱਲ 2 ਨਾਲ ਵੰਡਿਆ ਨਹੀਂ ਜਾ ਸਕਦਾ ਹੈ ਤਾਂ continue ਸਟੇਟਮੈਂਟ ਐਗਜ਼ੀਕਿਊਟ ਹੋ ਜਾਵੇਗੀ ਜੋ ਲੂਪ ਦੇ print(i) ਸਟੇਟਮੈਂਟ ਨੂੰ ਐਗਜ਼ੀਕਿਊਟ ਕੀਤੇ ਬਿਨਾਂ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਨੂੰ ਅਗਲੀ ਆਈਟ੍ਰੇਸ਼ਨ (next iteration) ਉਪਰ ਲੈ ਜਾਵੇਗੀ। ਇਸ ਤਰ੍ਹਾਂ, print(i) ਸਟੇਟਮੈਂਟ ਨੂੰ ਸਿਰਫ ਉਸ ਸਮੇਂ ਹੀ ਐਗਜ਼ੀਕਿਊਟ ਕੀਤਾ ਜਾਵੇਗਾ ਜਦੋਂ if ਕੰਡੀਸ਼ਨ ਗਲਤ (false) ਹੋਵੇਗੀ (i.e. ਜਦੋਂ ਵੀ i ਦਾ ਮੁੱਲ 2 ਨਾਲ ਵੰਡਿਆ ਜਾ ਸਕੇਗਾ)। ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਚਲਾਉਣ ਉਪਰੰਤ ਅੱਗੇ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਦਿਖਾਈ ਦੇਵੇਗੀ:

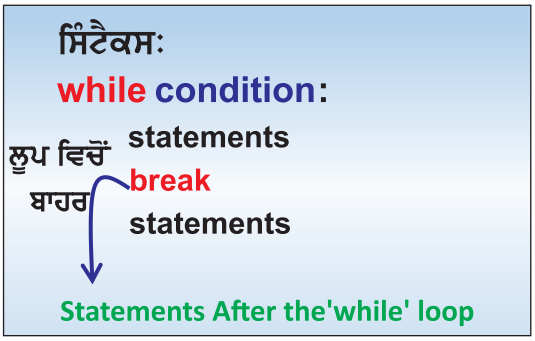
```
2
4
6
8
10
```

ਚਿੱਤਰ 4.35: ਪ੍ਰੋਗਰਾਮ (loop6.py) ਦੀ ਆਊਟਪੁੱਟ

- **break ਸਟੇਟਮੈਂਟ ('break' Statement):** ਪਾਈਥਨ ਵਿੱਚ break ਸਟੇਟਮੈਂਟ ਲੂਪ ਵਿਚੋਂ ਫਲੋਅ ਕੰਟਰੋਲ ਨੂੰ ਬਾਹਰ ਲਿਆਉਂਦਾ ਹੈ, ਭਾਵ break ਸਟੇਟਮੈਂਟਟ ਦੀ ਵਰਤੋਂ ਸਮੇਂ ਤੋਂ ਪਹਿਲਾਂ (ਭਾਵ ਲੂਪ ਪੂਰਾ ਹੋਣ ਤੋਂ ਪਹਿਲਾਂ) ਲੂਪ ਤੋਂ ਬਾਹਰ ਆਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਉਸ ਸਮੇਂ ਲਾਭਦਾਇਕ ਹੈ ਜਦੋਂ ਅਸੀਂ ਬਾਕੀ ਰਹਿੰਦੀਆਂ ਆਈਟ੍ਰੇਸ਼ਨਾਂ (remaining iterations) ਕਰਨ ਦੀ ਬਜਾਏ ਕੰਡੀਸ਼ਨ ਪੂਰੀ (fulfil) ਹੁੰਦੇ ਹੀ ਲੂਪ ਨੂੰ ਖਤਮ (terminate) ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ। ਹੇਠਾਂ ਦਿੱਤੇ ਸਿੰਟੈਕਸ ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ ਕਿ ਕਿਸ ਤਰ੍ਹਾਂ ਲੂਪ ਵਿਚ break ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੰਟਰੋਲ ਟ੍ਰਾਂਸਫਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ:

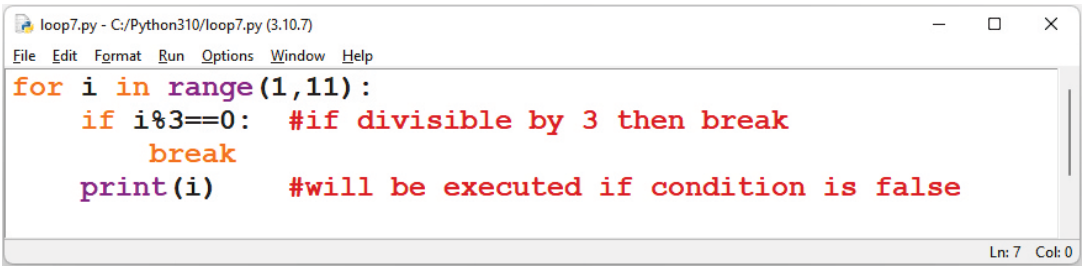


ਚਿੱਤਰ 4.36: 'for' ਲੂਪ ਵਿਚ 'break'



ਚਿੱਤਰ 4.37: 'while' ਲੂਪ ਵਿਚ 'break'

ਹੇਠਾਂ ਦਿੱਤਾ ਪ੍ਰੋਗਰਾਮ ਲੂਪ ਵਿੱਚ break ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾ ਰਿਹਾ ਹੈ:



ਪ੍ਰੋਗਰਾਮ (loo7.py): ਲੂਪ ਵਿਚ break ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ for ਲੂਪ ਨੂੰ 1 ਤੋਂ 10 ਤੱਕ ਨੰਬਰਾਂ ਦੇ ਕ੍ਰਮ, ਜੋ ਕਿ range () ਫੰਕਸ਼ਨ ਦੁਆਰਾ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਹੈ, ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਚਲਾਇਆ ਜਾ ਰਿਹਾ ਹੈ। ਲੂਪ ਵਿੱਚ ਕੰਡੀਸ਼ਨ (if i% 3= =0) ਦੀ ਵਰਤੋਂ ਇਹ ਜਾਂਚ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾ ਰਹੀ ਹੈ ਕਿ ਕੀ i ਦਾ ਮੁੱਲ 3 ਨਾਲ ਵੰਡਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਾਂ ਨਹੀਂ। ਜੇਕਰ i ਦਾ ਮੁੱਲ 3 ਨਾਲ ਵੰਡਿਆ ਜਾ ਸਕਦਾ ਹੈ ਤਾਂ break ਸਟੇਟਮੈਂਟ ਐਗਜ਼ੀਕਿਊਟ ਹੋ ਜਾਵੇਗੀ ਜੋ ਸਮੇਂ ਤੋਂ ਪਹਿਲਾਂ ਲੂਪ ਨੂੰ ਖਤਮ ਕਰ ਦੇਵੇਗੀ ਅਤੇ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਕੰਟਰੋਲ ਨੂੰ ਲੂਪ ਤੋਂ ਬਾਹਰ ਲੈ ਆਵੇਗੀ। ਇਸ ਤਰ੍ਹਾਂ ਲੂਪ ਵਿਚਲੀ print(i) ਸਟੇਟਮੈਂਟ ਨੂੰ ਉਸ ਸਮੇਂ ਹੀ ਲਾਗੂ ਕੀਤਾ ਜਾਵੇਗਾ ਜਦੋਂ ਲੂਪ ਵਿਚਲੀ if ਕੰਡੀਸ਼ਨ ਗਲਤ (false) ਹੋਵੇਗੀ (i.e. ਜਦੋਂ i ਦਾ ਮੁੱਲ 3 ਨਾਲ ਵੰਡਿਆ ਨਹੀਂ ਜਾਵੇਗਾ)। ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਚਲਾਉਣ ਉਪਰੰਤ ਹੇਠਾਂ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਦਿਖਾਈ ਦੇਵੇਗੀ:

```
1
2
```

ਚਿੱਤਰ 4.38: ਪ੍ਰੋਗਰਾਮ (look7.py) ਦੀ ਆਊਟਪੁੱਟ

- **'pass' ਸਟੇਟਮੈਂਟ ('pass' Statement):** ਅਸੀਂ ਖਾਲੀ ਲੂਪਸ (empty loops) ਲਿਖਣ ਲਈ ਪਾਈਥਨ ਵਿੱਚ pass ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। pass ਸਟੇਟਮੈਂਟ ਇੱਕ ਨੱਲ (null) ਸਟੇਟਮੈਂਟ ਹੈ, ਭਾਵ ਜਦੋਂ ਇਸ ਸਟੇਟਮੈਂਟ ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਕੁੱਝ ਨਹੀਂ (nothing) ਹੁੰਦਾ। ਕਈ ਵਾਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਿੱਚ ਅਜਿਹੀ ਸਥਿਤੀ ਆਉਂਦੀ ਹੈ ਜਿੱਥੇ ਸਾਨੂੰ ਸਿੰਟੈਕਟਿਕ ਤੌਰ 'ਤੇ (syntactically) ਇੱਕ ਖਾਲੀ ਬਲਾਕ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਪੈਂਦੀ ਹੈ, ਅਜਿਹੇ ਬਲਾਕ ਨੂੰ ਅਸੀਂ pass ਕੀਅਵਰਡ ਨਾਲ ਪਰਿਭਾਸ਼ਿਤ ਕਰ ਸਕਦੇ ਹਾਂ।

ਮਹੱਤਵਪੂਰਨ ਨੋਟ (Important Note):

'if' ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ ਦੀ ਤਰ੍ਹਾਂ for ਅਤੇ while ਲੂਪਸ ਨੂੰ ਵੀ ਇਕ ਲਾਈਨ ਵਿਚ ਹੀ ਉਸ ਸਮੇਂ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਦੋਂ ਲੂਪਸ ਦਾ ਬਲਾਕ ਸਧਾਰਨ ਸਿੰਗਲ ਸਟੇਟਮੈਂਟ ਦਾ ਹੋਵੇ। ਉਦਾਹਰਣਾਂ ਲਈ:

while ਲੂਪ ਦੀ ਸਿੰਗਲ ਲਾਈਨ ਵਿਚ ਉਦਾਹਰਣ:

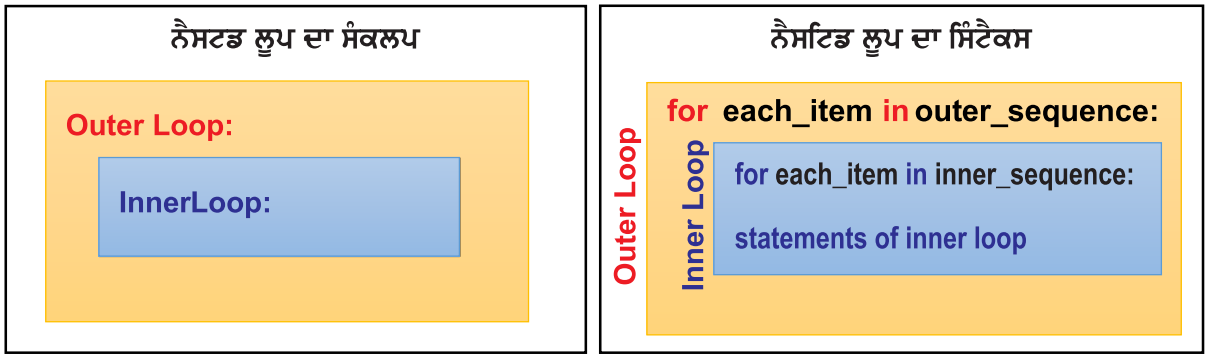
```
x =1
while x <=5:print(x); x = x+1
```

for ਲੂਪ ਦੀ ਸਿੰਗਲ ਲਾਈਨ ਵਿਚ ਉਦਾਹਰਣ:

```
for i in range(1, 5): print(i)
```

4.7 ਨੈਸਟਡ ਲੂਪਸ (Nested Loops)

ਨੈਸਟਡ ਲੂਪ ਦਾ ਅਰਥ ਹੈ ਇੱਕ ਲੂਪ ਦੇ ਅੰਦਰ ਇੱਕ ਹੋਰ ਲੂਪ। ਇਹਨਾਂ ਲੂਪਸ ਵਿਚ ਇੱਕ ਬਾਹਰੀ ਲੂਪ ਵਿੱਚ ਅੰਦਰੂਨੀ ਲੂਪ ਲਗਾਇਆ ਜਾਂਦਾ ਹੈ। ਨੈਸਟਡ ਲੂਪ ਦਾ ਸਿੰਟੈਕਸ ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਹੈ:



ਚਿੱਤਰ 4.39: ਨੈਸਟ ਲੂਪ ਦਾ ਸੰਕਲਪ ਅਤੇ ਸਿੰਟੈਕਸ

ਨੈਸਟਡ ਲੂਪਸ ਵਿੱਚ "ਅੰਦਰੂਨੀ ਲੂਪ" ਨੂੰ "ਬਾਹਰੀ ਲੂਪ" ਦੇ ਹਰੇਕ ਦੁਹਰਾਅ (each iteration) ਲਈ ਇੱਕ ਵਾਰ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਬਾਹਰੀ ਲੂਪ ਦੇ ਹਰ ਇੱਕ ਦੁਹਰਾਅ (each iteration) ਸਮੇਂ ਅੰਦਰੂਨੀ ਲੂਪ ਆਪਣੀਆਂ ਸਾਰੀਆਂ ਆਈਟ੍ਰੇਸ਼ਨਾਂ ਨੂੰ ਚਲਾਵੇਗਾ। ਬਾਹਰੀ ਲੂਪ ਦੇ ਹਰੇਕ ਦੁਹਰਾਅ ਲਈ ਅੰਦਰੂਨੀ ਲੂਪ ਮੁੜ-ਸ਼ੁਰੂ ਹੋ ਜਾਂਦਾ ਹੈ ਅਤੇ ਬਾਹਰੀ ਲੂਪ ਆਪਣੇ ਅਗਲੇ ਦੁਹਰਾਅ (next iteration) ਨੂੰ ਜਾਰੀ ਰੱਖਣ ਤੋਂ ਪਹਿਲਾਂ ਅੰਦਰੂਨੀ ਲੂਪ ਨੂੰ ਪੂਰਾ ਕਰਦਾ ਹੈ।

ਨੈਸਟਡ ਲੂਪਸ ਵਿਚ ਅਸੀਂ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੇ ਲੂਪ ਨੂੰ ਕਿਸੇ ਹੋਰ ਕਿਸਮ ਦੇ ਲੂਪ ਅੰਦਰ ਲਿਖ ਸਕਦੇ ਹਾਂ। ਉਦਾਹਰਨ ਲਈ: ਬਾਹਰੀ for ਲੂਪ ਵਿੱਚ ਇੱਕ while ਲੂਪ ਅਤੇ ਇਸਦੇ ਉਲਟ (vice versa) ਵੀ ਲੂਪਸ ਨੂੰ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਅੱਗੇ ਦਿੱਤੇ ਉਦਾਹਰਨ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ ਕਿ ਪਾਈਥਨ ਵਿੱਚ ਲੂਪਸ ਦੀ ਨੈਸਟਿੰਗ ਕਿਵੇਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ:

```
*loop8.py - C:/Python310/loop8.py (3.10.7)*
File  Edit  Format  Run  Options  Window  Help
for i in range(1,5):          #outer loop
    for j in range (1,5):     #inner loop
        print("#",end=" ")    #statement of inner loop
    print()                   #statement of outer loop
                                                    Ln: 7  Col: 0
```

ਪ੍ਰੋਗਰਾਮ (loop8.py): ਨੈਸਟਡ ਲੂਪ ਦੀ ਵਰਤੋਂ ਨਾਲ ਸਾਧਾਰਨ ਪੈਟਰਨ ਦਰਸਾਉਣ ਸਬੰਧੀ ਪ੍ਰੋਗਰਾਮ

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਬਾਹਰੀ ਲੂਪ ਦੇ ਦੁਹਰਾਓ ਦੀ ਕੁੱਲ ਸੰਖਿਆ 4 ਹੈ (i.e. i=1 ਤੋਂ 4 ਲਈ), ਜਦੋਂ ਕਿ ਅੰਦਰੂਨੀ ਲੂਪ, ਬਾਹਰੀ ਲੂਪ ਦੀ ਹਰੇਕ ਦੁਹਰਾਅ ਦੌਰਾਨ 4 ਵਾਰ (i.e. j=1 ਤੋਂ 4 ਲਈ) ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਅੰਦਰੂਨੀ ਲੂਪ ਦੇ ਹਰੇਕ ਦੁਹਰਾਅ ਵਿੱਚ ਅਸੀਂ # ਚਿੰਨ੍ਹ ਨੂੰ print('#',end= " ") ਸਟੇਟਮੈਂਟ ਨਾਲ ਪ੍ਰਿੰਟ ਕਰ ਰਹੇ ਹਾਂ। ਅੰਦਰਲਾ ਲੂਪ ਪੂਰਾ ਹੋਣ ਉਪਰੰਤ ਬਾਹਰਲੇ ਲੂਪ ਦੀ ਸਟੇਟਮੈਂਟ print() ਐਗਜ਼ੀਕਿਊਟ ਹੋਵੇਗੀ ਜੋ ਆਊਟਪੁੱਟ ਵਿਚ ਨਵੀਂ ਲਾਈਨ ਦਾਖਲ ਕਰ ਦੇਵੇਗੀ। ਇਸ ਪਰੋਗਰਾਮ ਨੂੰ ਚਲਾਉਣ 'ਤੇ, ਹੇਠਾਂ ਦਿੱਤੀ ਆਉਟਪੁੱਟ ਦਿਖਾਈ ਜਾਵੇਗੀ:

```
# # # #
# # # #
# # # #
# # # #
```

ਚਿੱਤਰ 4.40: ਪ੍ਰੋਗਰਾਮ **(loop8.py)** ਦੀ ਆਊਟਪੁੱਟ

ਲੈਬ ਐਕਟੀਵਿਟੀ

ੳ. ਬਹੁਪਸੰਦੀ ਪ੍ਰਸ਼ਨ (Multiple Choice Questions):

1. **ਜੇਕਰ ਅਸੀਂ ਪ੍ਰੋਗਰਾਮਿੰਗ ਦੌਰਾਨ ਦੋ ਜਾਂ ਉਸ ਤੋਂ ਵੱਧ ਆਪਸ਼ਨਾਂ ਵਿਚੋਂ ਕੋਈ ਇਕ ਆਪਸ਼ਨ ਸਿਲੈਕਟ ਕਰਨੀ ਹੋਵੇ, ਤਾਂ ਅਸੀਂ ਹੇਠਾਂ ਵਿਚੋਂ ਕਿਸਦੀ ਚੌਣ ਕਰਾਂਗੇ ?**

   ੳ) ਸਧਾਰਣ if　　ਅ) if elif else

   ੲ) ਸਿਕੁਐਂਸ਼ੀਅਲ ਐਗਜ਼ੀਕਿਊਸ਼ਨ　　ਸ) ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ

2. **ਤੁਸੀਂ ਹੇਠਾਂ ਦਿਤੀ ਕੰਡੀਸ਼ਨ ਲਈ ਲਾਜੀਕਲ ਐਕਸਪ੍ਰੈਸ਼ਨ ਕਿਸ ਤਰ੍ਹਾਂ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋਗੇ ?**
   "ਪ੍ਰਾਪਤ ਅੰਕ (Marks) 60 ਜਾਂ ਇਸਤੋਂ ਵੱਧ ਹਨ ਪਰੰਤੂ 80 ਤੋਂ ਘੱਟ ਹਨ"

   ੳ) if marks>=80 and marks<60:　　ਅ) if marks>60 and marks<=80:

   ੲ) if marks>=60 and marks<=80:　　ਸ) if marks>=60 and marks<80:

3. **ਜਦੋਂ ਅਸੀਂ if else ਕੰਡੀਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਿਸੇ ਹੋਰ if else ਕੰਡੀਸ਼ਨ ਵਿਚ ਕਰਦੇ ਹਾਂ ਤਾਂ ਉਸਨੂੰ ਕੀ ਕਹਿੰਦੇ ਹਨ ?**

   ੳ) else if ਲੈਡਰ　　ਅ) ਸਧਾਰਣ if else

   ੲ) ਨੈਸਟਡ if else　　ਸ) ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ

4. **ਕੰਡੀਸ਼ਨਲ ਫਲੋਅ ਸਟੇਟਮੈਂਟਸ ਨੂੰ _____ ਵੱਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।**

ੳ) ਬ੍ਰਾਂਚਿੰਗ ਸਟੇਟਮੈਂਟ
ਅ) ਫੈਂਸਲਾਂ ਕਰਨ ਵਾਲੇ ਸਟੇਟਮੈਂਟ
ੲ) ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟ
ਸ) ਉ ਤੋਂ ਅ ਦੋਵੇਂ

5. **ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟਸ ਨੂੰ _____ ਵੱਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।**

ੳ) ਆਇਟ੍ਰੇਟਿਵ ਸਟੇਟਮੈਂਟ
ਅ) ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ
ੲ) ਸਿਕੁਐਂਸ਼ੀਅਲ ਸਟੇਟਮੈਂਟ
ਸ) ਉਪਰੋਕਤ ਸਾਰੇ

6. **ਜੇਕਰ ਅਸੀਂ ਇਕ ਜਾਂ ਇਕ ਤੋਂ ਵੱਧ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਵਾਰ-ਵਾਰ ਦੁਹਰਾਉਣਾ ਹੋਵੇ ਤਾਂ ਅਸੀ ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਸ ਦੀ ਵਰਤੋਂ ਕਰਾਂਗੇ ?**

ੳ) ਆਈਟ੍ਰੇਟਿਵ ਸਟੇਟਮੈਂਟਸ
ਅ) ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟਸ
ੲ) ਸਕਿੱਪਿੰਗ ਸਟੇਟਮੈਂਟਸ
ਸ) ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ

7. **ਤੁਸੀਂ ਦਿੱਤੇ ਗਏ ਕ੍ਰਮ ਨੂੰ ਜੈਨਰੇਟ ਕਰਨ ਲਈ range ਫੰਕਸ਼ਨ ਨੂੰ ਕਿਸ ਤਰ੍ਹਾਂ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋਗੇ** (“2,4,6,8,10,12,14,16”) ?

ੳ) range(2,16)
ਅ) range(2,17)
ੲ) range(2,17,2)
ਸ) range(2,16,2)

8. **ਜਦੋਂ ਅਸੀਂ ਇਕ ਲੂਪ ਅੰਦਰ ਇਕ ਹੋਰ ਲੂਪ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦੇ ਹਾਂ ਤਾਂ ਉਸਨੂੰ ਕੀ ਕਹਿੰਦੇ ਹਨ ?**

ੳ) ਅਨੰਤ (Infinite) ਲੂਪ
ਅ) ਨੈਸਟਡ (Nested) ਲੂਪ
ੲ) for while ਲੂਪ
ਸ) ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ

9. **ਇਕ ਲੂਪ ਜਿਸ ਦਾ ਅੰਤ ਕਦੇ ਵੀ ਨਹੀਂ ਹੁੰਦਾ ?**

ੳ) ਲਗਾਤਾਰ (Continuous) ਲੂਪ
ਅ) ਅਨੰਤ (Infinite) ਲੂਪ
ੲ) ਚੱਕਰੀ (Circular) ਲੂਪ
ਸ) ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ

10. **_____ ਸਟੇਟਮੈਂਟ ਲੂਪ ਵਿਚੋਂ ਕੰਟਰੋਲ ਨੂੰ ਬਾਹਰ ਲੈ ਆਉਂਦੀ ਹੈ।**

ੳ) break
ਅ) back
ੲ) pass
ਸ) continue

11. **ਹੇਠ ਲਿਖਿਆਂ ਵਿਚੋਂ ਕਿਹੜਾ ਲੂਪ ਲਗਾਤਾਰ ਅਨੰਤ ਤੱਕ (continue infinitely) ਚਲਦਾ ਰਹੇਗਾ ?**

ੳ) while 0:
ਅ) while 1:
ੲ) while :1:
ਸ) while False:

ਅ. ਲੂਪਸ ਦੀ ਬਿਹਤਰ ਸਮਝ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਪ੍ਰੋਗਰਾਮਾਂ ਦੀ ਪ੍ਰੈਕਟਿਸ ਕਰੋ।

ਪ੍ਰੋਗਰਾਮ 1: ਕਿਸੇ ਪੋਜ਼ੀਟਿਵ ਨੰਬਰ ਦਾ ਫੈਕਟੋਰੀਅਲ **(factorial)** ਪਤਾ ਕਰਨ ਦਾ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ।

```
#Program for finding the factorial of a positive number
num=int(input("Enter a number to calculate factorial: "))
if(num>0):    #checking if the number is positive
    fact=1; i=1
    while(num>0):    #Calculating factorial for positive number
        fact=fact*num
        num=num-1

    print("Factorial of the number is: ", fact)
else:                #Shows error for negative number
    print("Error!!! Entered number is Negative...")
```

ਪ੍ਰੋਗਰਾਮ ਦੇ ਪਹਿਲੇ ਰਨ ਦਾ ਆਊਟਪੁੱਟ:

```
Enter a number to calculate factorial: 4
Factorial of the number is:  24
```

ਪ੍ਰੋਗਰਾਮ ਦੇ ਦੂਜੇ ਰਨ ਦਾ ਆਊਟਪੁੱਟ:

```
Enter a number to calculate factorial: -3
Error!!! Entered number is Negative...
```

ਪ੍ਰੋਗਰਾਮ 2: ਦਿੱਤੇ ਗਏ ਕ੍ਰਮ ਦਾ ਜੋੜ ਪਤਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ: $1+2+3+------+ \mathbf{n}$

```
#Program to find the sum of series
n=int(input("Enter value of n: "))
sum=0
for i in range(1,n+1):
    sum=sum + i

print("Sum of series is: ",sum)
```

ਆਊਟਪੁੱਟ:

```
Enter value of n: 5
Sum of series is:  15
```

ਪ੍ਰੋਗਰਾਮ 3: ਦਿੱਤੇ ਗਏ ਕ੍ਰਮ ਦਾ ਜੋੜ ਪਤਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ: $1^2+2^2+3^2+------+ \mathbf{n}^2$

```
#Program to find the sum of series
n=int(input("Enter value of n: "))
sum=0
for i in range(1,n+1):
    sum=sum + i**2

print("Sum of series is: ",sum)
```

ਆਊਟਪੁੱਟ:

```
Enter value of n: 4
Sum of series is:  30
```

ਏ. ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡ ਦੀ ਆਊਟਪੁੱਟ ਲਿਖੋ ਅਤੇ ਦਿੱਤੇ ਅਨੁਸਾਰ ਕੋਡ ਵਿੱਚ ਬਦਲਾਵ ਕਰੋ:

```
num1=_____

num2=_____

if _____________:

    print("Both Numbers are Equal")

elif _____________:

    print("Num1 is greater than Num2")

else:
    print("Num2 is greater than Num1")
```

ਉਪਰ ਦਿੱਤੇ ਗਏ ਕੋਡ ਨੂੰ ਹੇਠਾਂ ਲਿਖੇ ਅਨੁਸਾਰ ਪੂਰਾ ਕਰੋ ਜਾਂ ਉਸ ਵਿੱਚ ਤਬਦੀਲੀ ਕਰੋ:

1. ਸਭ ਤੋਂ ਪਹਿਲਾਂ ਉਪਰ ਦਿੱਤੇ ਕੋਡ ਨੂੰ ਇਸ ਤਰ੍ਹਾਂ ਪੂਰਾ ਕਰੋ ਕਿ if ਕੰਡੀਸ਼ਨ true ਹੋ ਜਾਵੇ।
2. ਹੁਣ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਸ ਤਰੀਕੇ ਨਾਲ ਬਦਲਾਵ ਕਰੋ ਕਿ elif ਕੰਡੀਸ਼ਨ ਦਾ ਬਲਾਕ ਐਗਜ਼ੀਕਿਊਟ ਹੋਵੇ।
3. ਹੁਣ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਸ ਤਰੀਕੇ ਨਾਲ ਬਦਲਾਵ ਕਰੋ ਕਿ else ਭਾਗ ਆਪਣਾ ਕੰਮ ਕਰੋ।

ਸ. ਹੇਠਾਂ ਦਿਤੇ ਕੋਡ ਨੂੰ ਪੂਰਾ ਕਰੋ/ਤਬਦੀਲੀ ਕਰੋ:

```
num1=10
num2=5
if num1>num2:
    if(num1%num2==0)
       print(num1, " is divisible by ", num2)
    else
       print(num, " is not divisible by ", num2)
else:
    print(num1*num2)
```

ਆਊਟਪੁੱਟ:

ਉਪਰ ਦਿੱਤੇ ਕੋਡ ਵਿੱਚ ਕੀ ਬਦਲਾਵ ਕੀਤੇ ਜਾਣ ਕਿ ਸਟੇਟਮੈਂਟ `print(num1*num2)` ਕੰਮ ਕਰੇ ?

ਬਦਲਾਵ :

ਹ. **ਹੇਠਾਂ ਦਿੱਤੇ ਕੋਡਸ ਦੀ ਆਊਟਪੁੱਟਸ ਲਿਖੋ:**

(i)
```
num = 10
while num> 1:
    print(num)
    num -= 2
```

(ii)
```
for i in range(1,11):
    print(i*5)
```

(iii)
```
for r in range(15,0,-2):
    print(r)
```

(iv)
```
state = 'PUNJAB'
for s in state:
    print (s)
```

(v)
```
i = 1; f = 1; n=5
while i <=n:
    f= f* i
    i = i + 1
print (f)
```

(vi)
```
for row in range(1,5):
    for col in range(1,row):
        print ("*", end="")
    print()
```

ਕ. **ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਲਈ ਪਾਈਥਨ ਵਿਚ ਬ੍ਰਾਂਚਿੰਗ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪ੍ਰੋਗਰਾਮਜ਼ ਲਿਖੋ:**

1. ਇਕ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰੋ ਜੋ ਯੂਜ਼ਰ ਦਾ ਨਾਂ (name) ਅਤੇ ਉਸਦੀ ਉਮਰ (age) ਇਨਪੁੱਟ ਕਰਨ ਬਾਰੇ ਪੁੱਛੇ ਅਤੇ ਫਿਰ ਇਹ ਮੈਸੇਜ਼ ਦਿਖਾਵੇ ਕਿ ਉਹ ਵੋਟ ਪਾਉਣ ਯੋਗ (Eligible to Vote) ਹੈ ਜਾਂ ਨਹੀਂ (Not Eligible to Vote); (ਵੋਟ ਪਾਉਣ ਯੋਗ ਉਮਰ 18 ਸਾਲ ਹੈ)
2. ਇਕ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰੋ ਜੋ ਤੁਹਾਡੇ ਦੁਆਰਾ ਇਨਪੁੱਟ ਕੀਤੇ ਗਏ ਸਾਲ (year) ਨੂੰ ਚੈਕ ਕਰਨ ਉਪਰੰਤ ਇਹ ਦਰਸਾਵੇ ਕਿ ਇਨਪੁੱਟ ਕੀਤਾ ਗਿਆ ਸਾਲ Leap Year ਹੈ ਜਾਂ ਨਹੀਂ (Not a Leap Year) ।
3. ਇਕ ਵਿਦਿਆਰਥੀ ਦੇ ਪ੍ਰਤੀਸ਼ਤ ਅੰਕ ਇਨਪੁੱਟ ਕਰਵਾਉਣ ਲਈ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰੋ। ਫਿਰ ਇਨਪੁੱਟ ਕੀਤੇ ਗਏ ਮੁੱਲ ਅਤੇ ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਕੰਡੀਸ਼ਨਾਂ ਅਨੁਸਾਰ ਵਿਦਿਆਰਥੀ ਦਾ ਗਰੇਡ ਦਰਸਾਓ।

ਪ੍ਰਤੀਸ਼ਤ ਅੰਕ	ਗਰੇਡ (Grade)
Above 90	A+
70 to 90	A
50 to 70	B
Below 50	C

4. ਦੋ ਅੰਕਾਂ ਉੱਪਰ ਮੁਢਲੇ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਸ਼ਨਾਂ ਨੂੰ ਕਰਵਾਉਣ ਲਈ ਇਕ ਸਾਧਾਰਣ ਕੈਲਕੁਲੇਟਰ ਤਿਆਰ ਕਰਨ ਸਬੰਧੀ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰੋ। ਪ੍ਰੋਗਰਾਮ ਅੱਗੇ ਦਿਤੇ ਅਨੁਸਾਰ ਕੰਮ ਕਰੇ:

a. ਯੂਜ਼ਰ ਤੋਂ ਦੋ ਅੰਕ ਪ੍ਰਾਪਤ ਕਰੋ

b. ਯੂਜ਼ਰ ਤੋਂ ਮੁੱਢਲੇ ਅਰਿੱਥਮੈਟਿਕ ਆਪਰੇਟਰਾਂ (+, -, *, /) ਵਿਚੋਂ ਕੋਈ ਆਪਰੇਟਰ ਇਨਪੁਟ ਕਰਨ ਲਈ ਪੁੱਛਿਆ ਜਾਵੇ। ਜੇਕਰ ਯੂਜ਼ਰ ਇਹਨਾਂ ਤੋਂ ਇਲਾਵਾ ਕੋਈ ਹੋਰ ਆਪਰੇਟਰ ਇਨਪੁਟ ਕਰਵਾਏ ਤਾਂ ਪ੍ਰੋਗਰਾਮ ਇਕ ਐਰਰ ਮੈਸੇਜ਼ (error message) ਦਰਸਾਵੇ।

c. ਇਨਪੁੱਟ ਕੀਤੇ ਆਪਰੇਟਰ ਅਨੁਸਾਰ ਯਜ਼ਰ ਦੁਆਰਾ ਦਿਤੇ ਗਏ ਅੰਕਾਂ ਉਪਰ ਕੰਮ ਕਰਨ ਤੋਂ ਬਾਅਦ ਨਤੀਜਾ ਦਿਖਾਈ ਦੇਵੇ।

ਖ. ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਟੇਟਮੈਂਟਸ ਲਈ ਪਾਈਥਨ ਵਿਚ ਲੂਪਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ:

1. ਦਿੱਤੇ ਗਏ ਨੰਬਰ ਦੀ ਸਾਰਣੀ (table) ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ। ਨੰਬਰ ਯੂਜ਼ਰ ਦੁਆਰਾ ਇਨਪੁੱਟ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।
2. ਲੂਪ ਅਤੇ range ਫੰਕਸ਼ਨ ਦੀ ਮਦਦ ਨਾਲ 5 ਦੇ ਸਾਰੇ ਗੁਣਜ (multiples) ਪ੍ਰਿੰਟ ਕਰੋ ਜੋ 50 ਤੋਂ ਛੋਟੇ ਹੋਣ।
3. ਹੇਠਾਂ ਦਿੱਤਾ ਕ੍ਰਮ ਬਣਾਉਣ ਲਈ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ:

   –5, –15, –25, ..., n

   ਜਿੱਥੇ n ਯੂਜ਼ਰ ਦੁਆਰਾ ਇਨਪੁੱਟ ਕੀਤਾ ਗਿਆ ਇੱਕ ਨੈਗੇਟਿਵ ਨੰਬਰ ਹੋਵੇਗਾ।
4. ਯੂਜ਼ਰ ਦੁਆਰਾ ਦਾਖਲ ਕੀਤੇ ਗਏ ਨੰਬਰ ਤੱਕ ਫਿਬੋਨਾਚੀ ਸੀਰੀਜ਼ (Fibonacci Series) ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ। Hint: 0, 1, 1, 2, 3, 5, 8, 13 .....
5. ਯੂਜ਼ਰ ਦੁਆਰਾ ਇਨਪੁਟ ਕੀਤੇ ਗਏ ਕਿਸੇ ਵੀ ਪੂਰਨ ਅੰਕ (integer) ਦੇ ਇਕੱਲੇ-ਇਕੱਲੇ ਅੰਕ ਦਾ ਜੋੜ ਪਤਾ ਕਰਨ ਲਈ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ। ਉਦਾਹਰਨ: ਜੇਕਰ ਇਨਪੁੱਟ ਕੀਤਾ ਗਿਆ ਪੂਰਨ ਅੰਕ 425 ਹੈ, ਤਾਂ ਇਸਦੇ ਇਕੱਲੇ-ਇਕੱਲੇ ਅੰਕ ਦਾ ਜੋੜ 4+2+5=11 ਹੋਵੇਗਾ।
6. ਹੇਠਾਂ ਦਿਤੇ ਪੈਟਰਨਜ਼ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਪਾਈਥਨ ਵਿਚ ਪ੍ਰੋਗਰਾਮਜ਼ ਲਿਖੋ:

ਪ੍ਰੈਟਰਨ 1:
```
*
**
***
****
```

ਪ੍ਰੈਟਰਨ 2:
```
****
***
**
*
```

ਪ੍ਰੈਟਰਨ 3:
```
1
1 2
1 2 3
1 2 3 4
```

ਪ੍ਰੈਟਰਨ 4:
```
1 2 3 4
1 2 3
1 2
1
```

ਡਾਟਾਬੇਸ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ

DBMS ਨਾਲ ਜਾਣ-ਪਛਾਣ

DBMS ਹਰ ਸੰਸਥਾ ਦਾ ਜ਼ਰੂਰੀ ਹਿੱਸਾ ਹੁੰਦਾ ਹੈ। ਇਹ ਹਰੇਕ ਵੱਡੇ ਅਤੇ ਛੋਟੇ ਵਪਾਰਕ ਸੰਸਥਾਵਾਂ ਲਈ ਬੁਨਿਆਦ ਦੇ ਤੌਰ ਤੇ ਕੰਮ ਕਰਦਾ ਹੈ। ਇਹ ਹਰ ਕੰਮ ਲਈ ਹੇਠਲੇ ਪੱਧਰ ਤੋਂ ਲੈ ਕੇ ਉਪਰਲੇ ਪੱਧਰ ਤੱਕ ਪ੍ਰਬੰਧਨ ਦੀ ਅਹਿਮ ਭੂਮਿਕਾ ਨਿਭਾਉਂਦਾ ਹੈ। ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਸਿਲੇਬਸ ਦੇ ਇਸ ਅਧਿਆਏ ਵਿੱਚ DBMS ਦੀਆਂ ਸਾਰੀਆਂ ਮੂਲ ਧਾਰਨਾਵਾਂ ਸ਼ਾਮਲ ਹਨ ਅਤੇ ਇਹ ਵਿਦਿਆਰਥੀਆਂ ਨੂੰ ਡਾਟਾ ਅਤੇ ਸੂਚਨਾ ਨੂੰ ਸਾਰਥਕ ਤਰੀਕੇ ਨਾਲ ਵਿਵਸਥਿੱਤ ਕਰਨ ਵਿੱਚ ਮਦਦ ਕਰੇਗਾ।

ਪਾਠ ਦਾ ਉਦੇਸ਼

- ✓ ਡਾਟਾ ਅਤੇ ਸੂਚਨਾ ਨਾਲ ਜਾਣ ਪਛਾਣ, ਡਾਟਾਬੇਸ ਧਾਰਨਾਵਾਂ, ਡਾਟਾਬੇਸ ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ, ਡਾਟਾਬੇਸ ਦੇ ਭਾਗ ਅਤੇ ਉਹਨਾਂ ਦੀ ਮਹੱਤਤਾ, ਪੁਰਾਤਨ ਫਾਈਲ ਪ੍ਰੋਸੈਸਿੰਗ ਅਤੇ ਆਧੁਨਿਕ ਡਾਟਾਬੇਸ ਤਕਨੀਕਾਂ ਦੀ ਇੱਕ ਸੰਖੇਪ ਜਾਣ-ਪਛਾਣ।
- ✓ DBMS ਦੀ ਜਾਣ-ਪਛਾਣ, ਡਾਟਾਬੇਸ ਉਪਭੋਗਤਾ, ਡਾਟਾ ਮਾਡਲਾਂ ਦੀਆਂ ਮੂਲ ਗੱਲਾਂ, ਡਾਟਾ ਮਾਡਲਾਂ ਦੀਆਂ ਕਿਸਮਾਂ ਅਤੇ ਉਹਨਾਂ ਦੀ ਮਹੱਤਤਾ, ਸਾਰੇ ਡਾਟਾ ਮਾਡਲਾਂ ਵਿੱਚ ਅੰਤਰ।
- ✓ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾ ਮਾਡਲ ਬਾਰੇ ਵਿਸਥਾਰ ਪੂਰਵਕ ਜਾਣਕਾਰੀ ਜਿਵੇਂ ਕਿ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ, ਟਪਲਸ, ਰਿਲੇਸ਼ਨ, ਡੋਮੇਨ, ਡਿਗਰੀ, ਕਾਰਡੀਨੈਲਿਟੀ, ਡਾਟਾਬੇਸ ਕੀਅਜ਼ ਆਦਿ।
- ✓ SQL ਅਤੇ ਇਸ ਦੀਆਂ ਭਾਸ਼ਾਵਾਂ ਦੀ ਜਾਣ-ਪਛਾਣ: DDL, DML ਅਤੇ DCL
- ✓ MySQL ਵਿੱਚ DDL ਅਤੇ DML ਕਮਾਂਡਾਂ ਦੀ ਵਰਤੋਂ ਕਰਨਾ

ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ✓ ਵਿਦਿਆਰਥੀ ਡਾਟਾਬੇਸ ਅਤੇ DBMS ਦੀਆਂ ਮੂਲ ਧਾਰਨਾਵਾਂ ਨੂੰ ਸਮਝਣਗੇ।
- ✓ ਉਹ DBMS ਦੇ ਵੱਖ-ਵੱਖ ਹਿੱਸਿਆਂ ਅਤੇ ਉਹਨਾਂ ਦੀ ਮਹੱਤਤਾ ਨੂੰ ਸਮਝਣਗੇ।
- ✓ ਉਹ ਵੱਖ-ਵੱਖ ਡਾਟਾ ਮਾਡਲਾਂ ਅਤੇ ਡਾਟਾਬੇਸ ਡਿਜ਼ਾਈਨਿੰਗ ਵਿੱਚ ਉਹਨਾਂ ਦੀ ਮਹੱਤਤਾ ਨੂੰ ਸਮਝਣਗੇ।
- ✓ ਉਹਨਾਂ ਨੂੰ SQL ਅਤੇ ਇਸ ਦੀਆਂ ਭਾਸ਼ਾਵਾਂ DDL ਅਤੇ DML ਦੀ ਮੁਢਲੀ ਸਮਝ ਹੋਵੇਗੀ।
- ✓ ਉਹ ਸਿੱਖਣਗੇ ਕਿ MySQL ਵਿੱਚ DDL ਅਤੇ DML ਕਮਾਂਡਾਂ ਦੀ ਵਰਤੋਂ ਕਿਵੇਂ ਕਰਨੀ ਹੈ?

05 ਪਾਠ ਡਾਟਾਬੇਸ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ ਦੀਆਂ ਮੂਲ ਧਾਰਨਾਵਾਂ ਬਾਰੇ ਜਾਣ-ਪਛਾਣ

ਕਿਸੇ ਵੀ ਸੰਸਥਾ ਦੁਆਰਾ ਆਪਣੇ ਸਫਲਤਾ ਟੀਚਿਆਂ ਨੂੰ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਲਗਭਗ ਹਰੇਕ ਖੇਤਰ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਕਿਸਮ ਦੇ ਸਬੰਧਤ ਤੱਥਾਂ ਅਤੇ ਸੰਖਿਆਤਮਕ ਅੰਕੜਿਆਂ ਨੂੰ ਰਿਕਾਰਡ ਕਰਨਾ ਜ਼ਰੂਰੀ ਹੁੰਦਾ ਹੈ। ਇੱਥੇ, ਅਸੀਂ ਵੱਖ-ਵੱਖ ਤੱਥਾਂ ਦੇ ਅਜਿਹੇ ਸੰਗ੍ਰਹਿ ਨੂੰ "ਡਾਟਾ" ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕਰ ਸਕਦੇ ਹਾਂ, ਜਿਸ ਵਿੱਚ ਸੰਖਿਆਵਾਂ, ਮੁੱਲ ਅਤੇ ਤੱਥ ਸ਼ਾਮਲ ਹੋ ਸਕਦੇ ਹਨ। ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਢੰਗ ਨਾਲ ਪ੍ਰਬੰਧਤ ਡਾਟਾ ਕਿਸੇ ਵੀ ਸੰਸਥਾ ਦੇ ਕੰਮ-ਕਾਜ ਨੂੰ ਪਰਭਾਵਿਤ ਕਰਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਅਸੀਂ ਕੰਮ ਕਰਦੇ ਹਾਂ। ਮਹੱਤਵਪੂਰਨ ਡਾਟਾ ਪ੍ਰਬੰਧਨ ਦੇ ਕੁਝ ਮਾਪਦੰਡ ਹੋਣੇ ਜ਼ਰੂਰੀ ਹਨ ਜੋ ਭਵਿੱਖ ਵਿੱਚ ਬਣਾਈਆਂ ਜਾਣ ਵਾਲੀਆਂ ਨੀਤੀਆਂ ਦੇ ਅਧਾਰ ਵਜੋਂ ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ।

ਡਾਟਾ ਮੂਲ ਰੂਪ ਵਿੱਚ ਕੱਚਾ ਅਤੇ ਗੈਰ-ਪ੍ਰੋਸੈਸਡ ਸਮੱਗਰੀ ਹੁੰਦੀ ਹੈ ਜੋ ਤੱਥਾਂ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਉਦਾਹਰਨ ਲਈ-ਨਾਮ, ਕਲਾਸ, ਨੰਬਰ, ਆਦਿ। ਜਾਣਕਾਰੀ ਦਾ ਇੱਕ ਭਾਗ ਜੋ ਕਿ ਪ੍ਰਕਿਰਿਆ ਲਈ ਮੂਲ ਰੂਪ ਵਿੱਚ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ, ਨੂੰ ਡਾਟਾ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਡਾਟਾ ਪਰਿਵਰਤਨਯੋਗ ਜਾਣਕਾਰੀ ਹੁੰਦੀ ਹੈ।

ਡਾਟਾ, ਡਾਟਾਬੇਸ ਅਤੇ ਡੀ.ਬੀ.ਐੱਮ.ਐੱਸ ਨਾਲ ਸਬੰਧਤ ਕਈ ਤੱਥ ਹਨ ਜਿਨ੍ਹਾਂ ਤੋਂ ਡਿਜੀਟਲ ਐਪਲੀਕੇਸ਼ਨਾਂ ਅਤੇ ਉਹਨਾਂ ਦੇ ਢਾਂਚੇ ਦੀ ਵਰਤੋਂ ਨੂੰ ਸਮਝਣ ਲਈ ਜਾਣੂ ਹੋਣ ਦੀ ਜ਼ਰੂਰਤ ਹੈ। ਆਉ ਡਾਟਾਬੇਸ ਦੇ ਸਾਰੇ ਨਿਯਮਾਂ ਅਤੇ ਤੱਥਾਂ ਬਾਰੇ ਵਿਸਥਾਰ ਵਿੱਚ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰੀਏ।

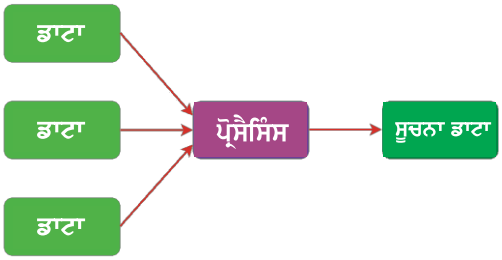
5.1 ਡਾਟਾਬੇਸ ਦੀ ਧਾਰਨਾ (DATABASE CONCEPT)

ਕੋਈ ਵੀ ਜਾਣਕਾਰੀ ਜੋ ਕੰਪਿਊਟਰ ਜਾਂ ਕਿਸੇ ਡਿਜੀਟਲ ਡਿਵਾਈਸ ਦੁਆਰਾ ਪ੍ਰੋਸੈਸਿੰਗ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ, ਹਰੇਕ ਸੰਸਥਾ ਲਈ ਬਹੁਤ ਹੀ ਅਹਿਸ ਹੁੰਦੀ ਹੈ। ਅਸੀਂ ਜਾਣਕਾਰੀ ਦੀ ਸਟੋਰੇਜ, ਪ੍ਰੋਸੈਸਿੰਗ ਅਤੇ ਲੋੜੀਂਦੀ ਸੂਚਨਾ ਦੀ ਪ੍ਰਾਪਤੀ ਨਾਲ ਸਬੰਧਤ ਸਾਰੇ ਕੰਮਾਂ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਲਈ ਇੱਕ ਸਿਸਟਮ/ਪ੍ਰਕਿਰਿਆ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ ਜੋ ਕਿ "ਡਾਟਾਬੇਸ" ਵਜੋਂ ਜਾਣੀ ਜਾਂਦੀ ਹੈ। ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਬੈਂਕਾਂ, ਹਸਪਤਾਲਾਂ ਆਦਿ ਵਿੱਚ ਜਾ ਸਕਦੀ ਹੈ।

ਇੱਕ ਡਾਟਾਬੇਸ ਸਬੰਧਤ ਡਾਟਾ ਦਾ ਇੱਕ ਸੰਗ੍ਰਹਿ ਹੁੰਦਾ ਹੈ ਜੋ ਸੰਗਠਿਤ ਹੁੰਦਾ ਹੈ ਅਤੇ ਡਾਟਾ ਬਿਨਾਂ ਕਿਸੇ ਰਿਡੰਡੈਂਸੀ ਦੇ ਸੁਰੱਖਿਅਤ ਤਰੀਕੇ ਨਾਲ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

ਡਾਟਾਬੇਸ ਨਾਲ ਸਬੰਧਤ ਦੋ ਮੁੱਖ ਭਾਗ ਹਨ ਜੋ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਹਨ।

1. **ਡਾਟਾ (Data) :** ਸ਼ਬਦ "ਡਾਟਾ" ਤੋਂ ਭਾਵ ਹੈ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਤੱਥਾਂ ਦਾ ਸੰਗ੍ਰਹਿ। ਕਿਸੇ ਵੀ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸੂਚਨਾ ਤਿਆਰ ਕਰਨ ਲਈ ਡਾਟਾ ਕੱਚੇ ਮਾਲ ਦੇ ਤੌਰ ਤੇ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ, ਪ੍ਰੋਸੈੱਸ ਅਤੇ ਸ਼ੇਅਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। "ਇੱਕਜੁੱਟ" ਤੋਂ ਸਾਡਾ ਭਾਵ ਹੈ ਕਿ ਡਾਟਾ ਘੱਟੋ-ਘੱਟ ਰਿਡੰਡੈਂਸੀ ਨਾਲ ਕਈ ਵੱਖ-ਵੱਖ ਫਾਈਲਾਂ ਤੋਂ ਏਕੀਕ੍ਰਿਤ ਹੈ। "ਸ਼ੇਅਰ" ਤੋਂ ਸਾਡਾ ਭਾਵ ਹੈ ਕਿ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਡਾਟਾ ਦੇ ਅਲੱਗ ਅਲੱਗ ਭਾਗ ਕਈ ਵੱਖ-ਵੱਖ ਯੂਜ਼ਰਜ਼ ਨਾਲ ਸਾਂਝੇ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ।
2. **ਸੂਚਨਾ (Information) :** ਸੂਚਨਾ ਨੂੰ ਇੱਕ ਸੰਗਠਿਤ ਜਾਂ ਵਰਗੀਕ੍ਰਿਤ ਡਾਟਾ ਵਜੋਂ ਵੀ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ, ਜਿਸ ਵਿੱਚ ਯੂਜ਼ਰ ਲਈ ਕੁਝ ਅਰਥਪੂਰਨ ਜਾਣਕਾਰੀ ਸ਼ਾਮਿਲ ਹੁੰਦੀ ਹੈ। ਸੂਚਨਾ ਇੱਕ ਪ੍ਰੋਸੈਸਡ ਡਾਟਾ ਹੁੰਦਾ ਹੈ ਜਿਸਦੀ ਵਰਤੋਂ ਫੈਸਲੇ ਲੈਣ ਲਈ ਜਾਂ ਨਵੀਆਂ ਪਾਲਸੀਆਂ ਬਣਾਉਣ ਲਈ ਹੁੰਦੀ ਹੈ। ਸੂਚਨਾ ਪੂਰੀ ਤਰ੍ਹਾਂ ਡਾਟਾ ਤੇ ਨਿਰਭਰ ਕਰਦੀ ਹੈ।



ਚਿੱਤਰ 5.1 ਡਾਟਾ ਅਤੇ ਸੂਚਨਾ

Activity No: 5.1

TEST YOURSELF

ਮੰਨ ਲਓ ਅਸੀਂ ਆਪਣੇ ਸਕੂਲ ਵਿੱਚ ਇੱਕ ਡਾਟਾਬੇਸ ਬਾਰੇ ਵਿਚਾਰ ਕਰ ਰਹੇ ਹਾਂ। ਆਉ ਹੇਠਾਂ ਦਿੱਤੇ ਸ਼ਬਦਾਂ ਦੀ ਪਛਾਣ ਕਰੀਏ ਕਿ ਕਿਹੜਾ ਡਾਟਾ ਨਾਲ ਸਬੰਧਤ ਹੈ ਜਾਂ ਕਿਹੜਾ ਜਾਣਕਾਰੀ ਦੀ ਇਕ ਉਦਾਹਰਨ ਹੈ:

ਢੁੱਕਵੇਂ ਵਿਕਲਪ ਤੇ (✓) ਨਿਸ ਲਗਾਓ:

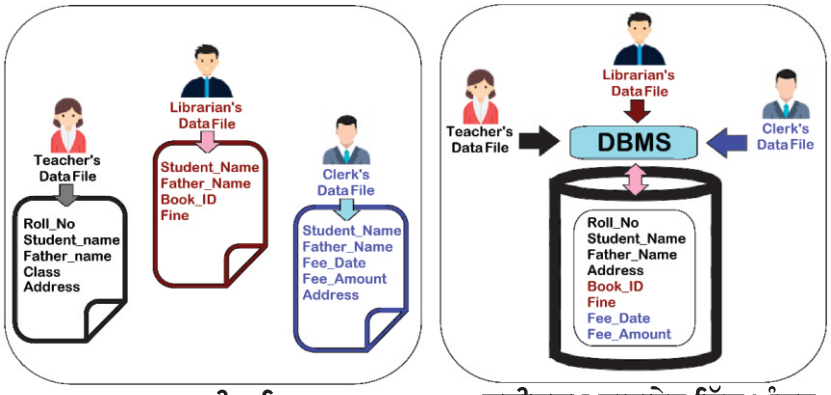
	Data	Information
1. ਅੰਕਾਂ ਦੀ ਸੂਚੀ।	☐	☐
2. ਅੰਗਰੇਜ਼ੀ ਵਿੱਚ ਪ੍ਰਾਪਤ ਅੰਕਾਂ ਦੀ ਔਸਤ।	☐	☐
3. ਸਾਡੀ ਕੰਪਿਊਟਰ ਲੈਬ ਵਿੱਚ ਕੰਪਿਊਟਰਾਂ ਦੇ ਸੀਰੀਅਲ ਨੰਬਰਾਂ ਦੀ ਸੂਚੀ।	☐	☐
4. ਰੋਲ ਨੰਬਰ 101 ਦੇ ਅੰਕਾਂ ਦੀ ਪ੍ਰਤੀਸ਼ਤਤਾ।	☐	☐

5.1.1 ਪਰੰਪਰਾਗਤ ਫਾਈਲ ਸਿਸਟਮ ਅਤੇ ਡਾਟਾਬੇਸ ਵਿੱਚ ਅੰਤਰ :

ਫਾਈਲ-ਆਧਾਰਿਤ ਡਾਟਾ ਸਟੋਰੇਜ਼ ਸਿਸਟਮ ਮੈਨੂੰਅਲ ਤਰੀਕੇ ਨਾਲ ਤੱਥਾਂ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਪੁਰਾਤਨ ਢੰਗ ਹੁੰਦੇ ਸਨ। ਇਸ ਕਿਸਮ ਦੀ ਸਟੋਰੇਜ ਵਿੱਚ ਹਰੇਕ ਮੋਡੀਊਲ ਇੱਕ ਡਾਟਾ ਪ੍ਰੋਸੈਸਿੰਗ ਮਾਹਰ ਦੀ ਮਦਦ ਨਾਲ ਆਪਣੇ ਖੁਦ ਦੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਅਤੇ ਕੰਟਰੋਲ ਕਰਦਾ ਹੈ। ਇੱਕ ਡਾਟਾ ਪ੍ਰੋਸੈਸਿੰਗ ਮਾਹਰ ਦੀ ਮੁੱਖ ਭੂਮਿਕਾ ਜ਼ਰੂਰੀ ਕੰਪਿਊਟਰ ਫਾਈਲ ਸਟਰਕਚਰ ਨੂੰ ਬਣਾਉਣਾ, ਢਾਂਚੇ ਦੇ ਅੰਦਰ ਡਾਟਾ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨਾ ਅਤੇ ਕੁਝ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨਾ ਜੋ ਫਾਈਲ ਆਧਾਰਿਤ ਡਾਟਾ ਦੀਆਂ ਰਿਪੋਰਟਾਂ ਬਣਾਉਣ ਦੇ ਯੋਗ ਹੋਣ।

ਇਸ ਤੋਂ ਇਲਾਵਾ ਇੱਕ "ਡਾਟਾਬੇਸ" ਡਾਟਾ ਦਾ ਸੰਗਠਿਤ ਸੰਗ੍ਰਹਿ ਹੁੰਦਾ ਹੈ ਜੋ ਇੱਕ ਅਰਥਪੂਰਨ ਤਰੀਕੇ ਨਾਲ ਸੰਬੰਧਤ ਹੈ ਜਿਸਨੂੰ ਵੱਖ-ਵੱਖ ਯੂਜ਼ਰਜ਼ ਦੁਆਰਾ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਬਿਨਾਂ ਕਿਸੇ ਰਿਡੰਡੈਂਸੀ ਦੇ ਐਕਸੈਸ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। DBMS ਸਿਸਟਮ ਦੁਆਰਾ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਵੱਖ-ਵੱਖ ਕਾਰਜ ਹਨ: ਨਵਾਂ ਡਾਟਾ ਭਰਨਾ, ਡਾਟਾ ਮਿਟਾਉਣਾ, ਡਾਟੇ ਦੀ ਚੋਣ ਕਰਨਾ, ਡਾਟਾ ਦੀ ਤਰਤੀਵ ਬਦਲਣਾ ਆਦਿ।

ਅਸੀਂ ਡਾਇਗ੍ਰਾਮ ਦੀ ਮਦਦ ਨਾਲ ਇਹਨਾਂ ਦੋਵਾਂ ਕਿਸਮਾਂ ਦੇ ਡਾਟਾ ਸਟੋਰੇਜ ਨੂੰ ਵੱਖ ਕਰ ਸਕਦੇ ਹਾਂ।



ਪੁਰਾਤਨ ਫਾਈਲ ਸਿਸਟਮ ਨਵੀਨਤਮ ਡਾਟਾਬੇਸ ਵਿੱਚ ਅੰਤਰ

ਚਿੱਤਰ 5.2 : ਪੁਰਾਤਨ ਫਾਈਲ ਸਿਸਟਮ ਅਤੇ ਨਵੀਨਤਮ ਡਾਟਾਬੇਸ ਵਿੱਚ ਅੰਤਰ

5.1.2 ਡਾਟਾਬੇਸ ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ:

ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਬਹੁਤ ਸਾਰੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਹੁੰਦੀਆਂ ਹਨ। ਇੱਕ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੀਆਂ ਕੁਝ ਲੋੜੀਂਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਇਸ ਤਰ੍ਹਾਂ ਹੋਣੀਆਂ ਚਾਹੀਦੀਆਂ ਹਨ:

1. **ਘੱਟ ਤੋਂ ਘੱਟ ਡਾਟਾ ਡੁਪਲੀਕੇਸੀ (Minimal Redundancy):** ਰਵਾਇਤੀ ਫਾਈਲ ਸਿਸਟਮ ਵਿੱਚ ਹਰੇਕ ਐਪਲੀਕੇਸ਼ਨ (Application) ਦੀਆਂ ਆਪਣੀਆਂ ਨਿੱਜੀ ਫਾਈਲਾਂ ਹੁੰਦੀਆਂ ਹਨ। ਇਹਨਾਂ ਫਾਈਲਾਂ ਨੂੰ ਕਿਸੇ ਐਪਲੀਕੇਸ਼ਨ ਵਿਚਕਾਰ ਸਾਂਝਾ (Share) ਨਹੀਂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਲਈ ਇਸ ਕਿਸਮ ਦੇ ਸਿਸਟਮ ਵਿੱਚ ਇੱਕੋ ਡਾਟਾ ਨੂੰ ਵੱਖ ਵੱਖ ਫਾਈਲਾਂ ਵਿੱਚ ਕਾਪੀ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਟੋਰ ਕੀਤੇ ਡਾਟਾ ਵਿੱਚ ਕਾਫ਼ੀ ਰਿਡੰਡੈਂਸੀ ਪੈਦਾ ਹੁੰਦੀ ਹੈ। ਇਸ ਨਾਲ ਸਟੋਰੇਜ ਸਪੇਸ ਦੀ ਬਰਬਾਦੀ ਵੀ ਹੁੰਦੀ ਹੈ। ਪਰੰਤੂ ਡਾਟਾਬੇਸ ਦੀ ਮਦਦ ਨਾਲ ਕੇਂਦਰੀਕ੍ਰਿਤ ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸ ਤੋਂ ਬਚਿਆ ਜਾ ਸਕਦਾ ਹੈ।
2. **ਨਾਬਰਾਬਰਤਾ (Inconsistency) ਤੋਂ ਬਚਿਆ ਜਾ ਸਕਦਾ ਹੈ:** ਰਵਾਇਤੀ ਫਾਈਲ ਸਿਸਟਮ ਵਿੱਚ ਇੱਕ ਡਾਟਾ ਨੂੰ ਵੱਖ-ਵੱਖ ਫਾਈਲਾਂ ਵਿੱਚ ਡੁਪਲੀਕੇਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਇਸ ਡਾਟਾ ਨੂੰ ਕਿਸੇ ਵੀ ਇੱਕ ਫਾਈਲ ਵਿੱਚ ਬਦਲਿਆ ਜਾਂਦਾ ਹੈ ਪਰੰਤੂ ਇਸਨੂੰ ਦੂਜੀਆਂ ਫਾਈਲਾਂ ਵਿੱਚ ਅਪਡੇਟ ਨਹੀਂ ਕੀਤਾ ਜਾਂਦਾ, ਤਾਂ ਇਹ ਅਸੰਗਤਤਾ (Inconsistency) ਨੂੰ ਜਨਮ ਦਿੰਦਾ ਹੈ। ਫਾਈਲ ਸਿਸਟਮ ਦੀ ਇਸ ਕਮੀ ਨੂੰ ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਹੱਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜਿੱਥੇ ਸਾਰੇ ਮੋਡੀਊਲਜ਼ ਦੁਆਰਾ ਡਾਟਾ ਦੀ ਇੱਕ ਸਾਂਝੀ ਕਾਪੀ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਕੀਤੇ ਗਏ ਬਦਲਾਅ ਸਾਰੇ ਮੋਡੀਊਲਜ਼ ਵਿੱਚ ਵਰਤੋਂ ਯੋਗ ਹੋਣਗੇ।
3. **ਡਾਟਾ ਦਾ ਸਾਂਝਾਕਰਨ (Sharing of data):** ਰਵਾਇਤੀ ਫਾਈਲ ਸਿਸਟਮ ਵਿੱਚ ਹਰੇਕ ਐਪਲੀਕੇਸ਼ਨ ਦੀਆਂ ਆਪਣੀਆਂ ਨਿੱਜੀ ਫਾਈਲਾਂ ਹੁੰਦੀਆਂ ਹਨ। ਇਹਨਾਂ ਫਾਈਲਾਂ ਨੂੰ ਇੱਕੋ ਸਮੇਂ ਕਈ ਐਪਲੀਕੇਸ਼ਨਾਂ ਵਿਚਕਾਰ ਸਾਂਝਾ (Sharing) ਨਹੀਂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਪਰ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਨੂੰ ਵੱਖ-ਵੱਖ ਉਪਭੋਗਤਾਵਾਂ ਅਤੇ ਐਪਲੀਕੇਸ਼ਨਾਂ ਵਿੱਚ ਸਾਂਝਾ ਕੀਤਾ ਜਾਣਾ ਸੰਭਵ ਹੁੰਦਾ ਹੈ।
4. **ਲੋੜੀਂਦਾ ਡਾਟਾ ਖੋਜਣ/ਸਰਚ ਕਰਨ ਯੋਗ ਹੁੰਦਾ ਹੈ (Search Capability):** ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਨੂੰ ਤੇਜ਼ੀ ਨਾਲ ਖੋਜਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਸਰਚ ਕਾਰਜ ਨੂੰ ਕਈ ਵੱਖ-ਵੱਖ ਤਰੀਕਿਆਂ ਨਾਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਾਰਾ ਡਾਟਾ ਇੱਕ ਕੇਂਦਰੀ ਸਟੋਰੇਜ ਤੇ ਉਪਲਬਧ ਹੁੰਦਾ ਹੈ ਅਤੇ ਲੋੜ ਪੈਣ ਤੇ ਸਕਿੰਟਾਂ ਦੇ ਅੰਦਰ ਲੋੜੀਂਦਾ ਡਾਟਾ ਖੋਜਿਆ ਜਾ ਸਕਦਾ ਹੈ।
5. **ਇੱਕਜੁੱਟਤਾ (Integrity):** ਡਾਟਾ ਦੀ ਇੱਕਜੁੱਟਤਾ ਤੋਂ ਭਾਵ ਹੈ ਕਿ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਹਮੇਸ਼ਾਂ ਸਹੀ ਤੌਰ ਤੇ ਸਟੋਰ ਹੁੰਦਾ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਗਲਤ ਜਾਣਕਾਰੀ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ ਨਹੀਂ ਕੀਤੀ ਜਾ ਸਕਦੀ। ਡਾਟਾ ਦੀ ਇਕਸਾਰਤਾ ਨੂੰ ਬਣਾਈ ਰੱਖਣ ਲਈ, ਡਾਟਾਬੇਸ 'ਤੇ ਕੁਝ ਇਕਸਾਰਤਾ ਪਾਬੰਦੀਆਂ (integrity constraints) ਲਾਗੂ ਕੀਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ।
6. **ਪ੍ਰਾਈਵੇਸੀ (Privacy) ਅਤੇ ਸੁਰੱਖਿਆ (Security):** ਜਦੋਂ ਬਹੁਤ ਸਾਰੇ ਯੂਜ਼ਰਜ਼ ਇੱਕ ਸਾਂਝੇ ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਨ ਤਾਂ ਇਹ ਸੰਭਵ ਹੈ ਕਿ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਤੱਕ ਪਹੁੰਚ ਕਰਨ ਦੇ ਵੱਖੋ ਵੱਖਰੇ ਅਧਿਕਾਰ ਤੈਅ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹੋਣ। ਅਸੀਂ ਆਪਣੇ ਕੀਮਤੀ ਡਾਟਾ ਨੂੰ ਸੁਰੱਖਿਅਤ ਕਰਨ ਲਈ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਪ੍ਰਮਾਣਿਕਤਾ (authentication) ਵਿਧੀਆਂ ਨੂੰ ਵੀ ਆਸਾਨੀ ਨਾਲ ਲਾਗੂ ਕਰ ਸਕਦੇ ਹਾਂ ਤਾਂ ਜੋ ਡਾਟਾਬੇਸ ਵਿੱਚ ਮੌਜੂਦ ਡਾਟਾ ਦੀ ਪ੍ਰਾਈਵੇਸੀ ਅਤੇ ਸੁਰੱਖਿਆ ਪ੍ਰਦਾਨ ਕੀਤੀ ਜਾ ਸਕੇ।
7. **ਆਸਾਨੀ ਨਾਲ ਪਹੁੰਚਯੋਗ (Easily Accessible):** ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਨੂੰ ਵੱਖ-ਵੱਖ ਤਰੀਕਿਆਂ ਨਾਲ ਸਾਰੇ ਯੂਜ਼ਰਜ਼ ਵਿੱਚ ਸਾਂਝਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਲੋੜੀਂਦੇ ਡਾਟਾ ਦੀ ਉਪਲਬਧਤਾ ਯਕੀਨੀ ਬਣਾਉਂਦਾ ਹੈ ਤਾਂ ਜੋ ਜਦੋਂ ਵੀ ਅਤੇ ਜਿਥੇ ਵੀ ਉਸਦੀ ਲੋੜ ਹੋਵੇ, ਡਾਟਾ ਮੁਹੱਈਆ ਕਰਵਾਇਆ ਜਾ ਸਕੇ।
8. **ਆਸਾਨ ਬੈਕਅੱਪ/ਰਿਕਵਰੀ (Easy Backup/Recovery):** ਡਾਟਾਬੇਸ ਬੈਕਅੱਪ ਰੱਖਣਾ ਹਰ ਸੰਸਥਾ ਲਈ ਜ਼ਰੂਰੀ ਹੈ। ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋ ਨਿਯਮਤ ਬੈਕਅੱਪ ਲੈਣਾ ਬਹੁਤ ਸੁਵਿਧਾਜਨਕ (convenient) ਅਤੇ ਆਸਾਨ ਹੈ। ਕਿਸੇ ਵੀ ਹਾਰਡਵੇਅਰ ਦੀ ਅਸਫਲਤਾ ਦੇ ਮਾਮਲੇ ਵਿੱਚ ਡਾਟਾ ਜਾਂ ਸੂਚਨਾ ਦੇ ਨੁਕਸਾਨ ਨੂੰ ਘੱਟ ਕਰਨ ਲਈ ਬੈਕਅੱਪ ਬਹੁਤ ਜ਼ਰੂਰੀ ਹੈ ਤਾਂ ਜੋ ਰਿਕਵਰੀ ਕਰਨ ਉਪਰੰਤ ਡਾਟਾ ਮੁੜ ਪ੍ਰਾਪਤ ਕੀਤਾ ਜਾ ਸਕੇ।

ਨੋਟ: ਇਹ ਸਾਰੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਸਿਰਫ਼ ਕੰਪਿਊਟਰਾਈਜ਼ਡ ਡਾਟਾਬੇਸ ਪ੍ਰਬੰਧਨ ਸਿਸਟਮ ਨਾਲ ਉਪਲਬਧ ਹਨ। ਪੁਰਾਤਨ ਫਾਈਲ ਸਿਸਟਮ ਡਾਟਾ ਹੈਂਡਲਿੰਗ ਦੀਆਂ ਇਹਨਾਂ ਸ਼ਾਨਦਾਰ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਤੋਂ ਬਹੁਤ ਦੂਰ ਸੀ।

ਐਕਟੀਵਿਟੀ: 5.2

TEST Yourself ਉਹਨਾਂ ਸਥਾਨਾਂ ਦਾ ਨਾਮ ਲਿਖੋ ਜਿੱਥੇ ਡਾਟਾਬੇਸ ਵਰਤਿਆ ਜਾ ਰਿਹਾ ਹੈ:

1. _____ 2. _____
3. _____ 4. _____
5. _____ 6. _____
7. _____ 8. _____
9. _____ 10. _____

5.2 DBMS ਨਾਲ ਜਾਣ-ਪਛਾਣ (INTRODUCTION OF DBMS)

ਡਾਟਾਬੇਸ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ (DBMS) ਇੱਕ ਸਾਫਟਵੇਅਰ ਹੈ ਜੋ ਲੋੜੀਂਦੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਕਰਨ, ਲੋੜ ਅਨੁਸਾਰ ਪ੍ਰਾਪਤ ਕਰਨ ਅਤੇ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇੱਕ DBMS ਇੱਕ ਐਂਡ-ਯੂਜ਼ਰ ਅਤੇ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਦੇ ਤੌਰ ਤੇ ਕੰਮ ਕਰਦਾ ਹੈ। DBMS ਯੂਜ਼ਰ ਨੂੰ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਬਣਾਉਣ, ਪੜ੍ਹਨ, ਅਪਡੇਟ ਕਰਨ ਅਤੇ ਬੇਲੋੜਾ ਡਾਟਾ ਮਿਟਾਉਣ ਦੀ ਆਗਿਆ ਦੇਣ ਲਈ ਇੰਟਰਫੇਸ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਹ ਡਾਟਾ ਸੁਰੱਖਿਆ (security), ਡਾਟਾ ਅਖੰਡਤਾ (data integrity), ਇੱਕੋ ਸਮੇਂ ਡਾਟਾ ਦੀ ਵਧੇਰੇ ਥਾਵਾਂ ਤੇ ਵਰਤੋਂ (concurrency) ਅਤੇ ਇਕਸਾਰ ਡਾਟਾ ਪ੍ਰਸ਼ਾਸਨ (uniform data administration) ਪ੍ਰਕਿਰਿਆਵਾਂ ਪ੍ਰਦਾਨ ਕਰਨ ਵਿੱਚ ਵੀ ਮਦਦ ਕਰਦਾ ਹੈ।

ਇੱਕ DBMS ਇੱਕ ਸਾਫਟਵੇਅਰ ਹੈ ਜੋ ਡਾਟਾਬੇਸ ਬਣਾਉਣ ਅਤੇ ਪ੍ਰਬੰਦਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਟ ਡਾਟਾ ਨੂੰ ਬਣਾਉਣਾ, ਪ੍ਰਾਪਤ ਕਰਨਾ, ਅੱਪਡੇਟ ਕਰਨਾ, ਮਿਟਾਉਣਾ ਅਤੇ ਸੁਰੱਖਿਅਤ ਕਰਨ ਵਰਗੇ ਬੁਨਿਆਦੀ ਕਾਰਜ ਕਰਦਾ ਹੈ।

ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਬਹੁਤ ਸਾਰੇ ਯੂਜ਼ਰਜ਼ ਦੁਆਰਾ ਵੱਖ-ਵੱਖ ਉਦੇਸ਼ਾਂ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਕੁਝ ਯੂਜ਼ਰਜ਼ ਡਾਟਾ ਨੂੰ ਪ੍ਰਾਪਤ ਕਰ ਕੇ ਵਰਤੋਂ ਕਰਨ ਵਿੱਚ ਸ਼ਾਮਲ ਹੋ ਸਕਦੇ ਹਨ ਅਤੇ ਕੁਝ ਯੂਜ਼ਰਜ਼ ਡਾਟਾਬੇਸ ਦਾ ਬੈਕਅੱਪ ਲੈਣ ਨਾਲ ਸੰਬੰਧਤ ਹੋ ਸਕਦੇ ਹਨ। ਯੂਜ਼ਰਜ਼ ਦੇ ਇਹਨਾਂ ਕੰਮਾਂ ਨੂੰ ਪੂਰਾ ਕਰਨ ਵਿੱਚ ਨਿਭਾਈਆਂ ਜਾਣ ਵਾਲੀਆਂ ਉਹਨਾਂ ਦੀਆਂ ਭੂਮਿਕਾਵਾਂ ਦੇ ਅਧਾਰ ਤੇ ਹੀ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਉਹਨਾਂ ਵਿੱਚੋਂ ਕੁਝ ਦਾ ਵਰਣਨ ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਹੈ:

5.2.1 ਡਾਟਾਬੇਸ ਇਨਵਾਇਰਨਮੈਂਟ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਰੋਲਜ਼/ਯੂਜ਼ਰਜ਼ (Roles/Users):

ਅਸੀਂ ਡਾਟਾਬੇਸ ਦੇ ਉਪਭੋਗਤਾਵਾਂ ਨੂੰ ਹੇਠ ਲਿਖੀਆਂ ਕਿਸਮਾਂ ਵਿੱਚ ਸ਼੍ਰੇਣੀਬੱਧ ਕਰ ਸਕਦੇ ਹਾਂ:

1. **ਡਾਟਾਬੇਸ ਐਡਮੀਨੀਸਟ੍ਰੇਟਰ (Database Administrator):** DBA ਇੱਕ ਵਿਅਕਤੀ ਜਾਂ ਵਿਅਕਤੀਆਂ ਦਾ ਸਮੂਹ ਹੁੰਦਾ ਹੈ। ਇਹ ਯੂਜ਼ਰ DBMS ਦੇ ਰੱਖ ਰਖਾਵ ਦਾ ਮੁੱਖ ਕਾਰਜ ਕਰਦੇ ਹਨ। ਇਹ ਡਾਟਾਬੇਸ ਦੇ ਪ੍ਰਬੰਧਨ ਅਤੇ ਇਸ ਦੀ ਵਰਤੋਂ ਦੀ ਦੇਖਭਾਲ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦੇ ਹਨ। ਇਹ ਡੀ.ਬੀ.ਏ. ਦੁਆਰਾ ਤੈਅ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਕਿ ਕਿਹੜਾ ਯੂਜ਼ਰ ਡਾਟਾਬੇਸ ਤੇ ਕੀ ਕੀ ਕੰਮ ਕਰ ਸਕਦਾ ਹੈ? DBA (ਡਾਟਾ-ਬੇਸ ਐਡਮਿਨਿਸਟ੍ਰੇਟਰ) ਦੇ ਮੁੱਖ-ਕਾਰਜ ਹੇਠ ਦਰਸਾਏ ਅਨੁਸਾਰ ਹਨ:

- **ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ ਹੋਣ ਵਾਲੇ ਡਾਟਾ ਨਾਲ ਸਬੰਧਤ ਫੈਸਲੇ:** ਇਹ ਫੈਸਲਾ ਕਰਨਾ DBA ਦਾ ਕੰਮ ਹੈ ਕਿ **ਡਾਟਾਬੇਸ** ਵਿੱਚ ਕਿਸ ਕਿਸਮ ਦੀ ਜਾਣਕਾਰੀ ਸਟੋਰ ਕੀਤੀ ਜਾਣੀ ਹੈ।
- **ਡਾਟਾਬੇਸ ਦੇ ਸਟੋਰੇਜ਼ ਢਾਂਚੇ ਅਤੇ ਡਾਟਾ ਪਹੁੰਚ ਸੰਬੰਧੀ ਯੋਜਨਾਵਾਂ:** ਇਹ ਫੈਸਲਾ ਵੀ DBA ਦੁਆਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਕਿ **ਡਾਟਾਬੇਸ** ਵਿੱਚ **ਡਾਟਾ** ਨੂੰ ਕਿਵੇਂ ਦਰਸਾਇਆ ਜਾਣਾ ਹੈ।
- **ਯੂਜ਼ਰ ਨੂੰ ਸਹਾਇਤਾ ਪ੍ਰਦਾਨ ਕਰਨਾ:** ਯੂਜ਼ਰ ਨੂੰ ਸਹਾਇਤਾ ਪ੍ਰਦਾਨ ਕਰਨਾ ਅਤੇ ਪ੍ਰਬੰਧਨ ਦੇ ਸਾਰੇ ਪੱਧਰਾਂ ਤੇ ਉਹਨਾਂ ਨਾਲ ਸੰਚਾਰ ਕਰਨਾ DBA ਦੀ ਜ਼ਿੰਮੇਵਾਰੀ ਹੁੰਦੀ ਹੈ।
- **ਸੁਰੱਖਿਆ ਅਤੇ ਇਕਜੁੱਟਤਾ ਜਾਂਚਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨਾ:** DBA ਪ੍ਰਮਾਣਿਕਤਾ (authorization) ਚੈੱਕ ਨਿਰਧਾਰਿਤ ਕਰਨ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦਾ ਹੈ। ਇਹ ਸੁਨਿਸ਼ਚਿਤ ਕਰਦਾ ਹੈ ਕਿ ਕੋਈ ਵੀ ਅਣਅਧਿਕਾਰਤ ਯੂਜ਼ਰ **ਡਾਟਾਬੇਸ** ਤੱਕ ਪਹੁੰਚ ਨਾ ਕਰ ਸਕਦਾ ਹੋਵੇ ਅਤੇ ਡਾਟਾਬੇਸ ਪੂਰੀ ਤਰ੍ਹਾਂ ਸੁਰੱਖਿਅਤ ਰਹੇ। DBA **ਡਾਟਾਬੇਸ** ਦੀ ਇੱਕ ਜੁੱਟਤਾ ਨੂੰ ਵੀ ਯਕੀਨੀ ਬਣਾਉਂਦਾ ਹੈ।
- **ਬੈਕਅੱਪ ਅਤੇ ਰਿਕਵਰੀ ਪਾਲਸੀ ਤਿਆਰ ਕਰਨਾ:** ਜੇਕਰ **ਡਾਟਾਬੇਸ** ਵਿੱਚ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੀ ਕੋਈ ਵੀ ਸਮੱਸਿਆ ਆਉਂਦੀ ਹੈ ਤਾਂ ਇੱਕ DBA **ਡਾਟਾ** ਦੀ ਮੁਰੰਮਤ ਕਰਨ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦਾ ਹੈ। ਇਹ ਫੈਸਲਾ ਕਰਦਾ ਹੈ ਕਿ **ਡਾਟਾਬੇਸ** ਦਾ ਬੈਕਅੱਪ ਕਦੋਂ ਲੈਣਾ ਹੈ ਅਤੇ ਇਸਨੂੰ ਕਿਵੇਂ ਰਿਕਵਰ ਕਰਨਾ ਹੈ।
- **ਕਾਰਗੁਜ਼ਾਰੀ ਦੀ ਨਿਗਰਾਨੀ:** DBA ਡਾਟਾਬੇਸ ਦੇ ਸੰਬੰਧਤ ਸਾਰੇ ਕੰਮਾਂ ਦੀ ਨਿਗਰਾਨੀ ਕਰਨ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦਾ ਹੈ। ਇਸਦੇ ਲਈ ਉਹ **ਡਾਟਾਬੇਸ** ਵਿੱਚ ਸੁਧਾਰ ਕਰਦਾ ਹੈ।

2. **ਡਿਜ਼ਾਈਨਰ:** ਇਹ ਅਜਿਹੇ ਯੂਜ਼ਰਜ਼ ਦਾ ਇਕ ਸਮੂਹ ਹੁੰਦਾ ਹੈ ਜੋ ਅਸਲ ਵਿੱਚ ਡਾਟਾਬੇਸ ਦੇ ਸਾਰੇ ਹਿੱਸਿਆਂ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਦੇ ਹਨ। ਡਾਟਾਬੇਸ ਬਣਾਉਣ ਦੀ ਪ੍ਰਕੀਰਿਆ ਦੀ ਸ਼ੁਰੂਆਤ ਡਾਟਾਬੇਸ ਤੋਂ ਵੱਖ-ਵੱਖ ਯੂਜ਼ਰਜ਼ ਦੀਆਂ ਲੋੜਾਂ ਦਾ ਵਿਸ਼ਲੇਸ਼ਣ ਕਰਨ ਉਪਰੰਤ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਤੋਂ ਬਾਅਦ ਹੀ ਇੱਕ ਚੰਗਾ ਡਿਜ਼ਾਈਨ ਤਿਆਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਡਿਜ਼ਾਇਨਰ ਡਾਟਾਬੇਸ ਦੀਆਂ ਸਾਰੀਆਂ ਇਕਾਈਆਂ (entities), ਰਿਲੇਸ਼ਨਾਂ (relations), ਨਿਯਮਾਂ (constraints) ਅਤੇ ਵਿਊਜ਼ (views) ਦੇ ਪੂਰੇ ਸਮੂਹ ਦੀ ਪਹਿਚਾਣ ਅਤੇ ਡਿਜ਼ਾਈਨ ਕਰਦੇ ਹਨ।

3. **ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮਰ:** ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮਰ ਉਹ ਡਾਟਾਬੇਸ ਯੂਜ਼ਰਜ਼ ਹੁੰਦੇ ਹਨ ਜੋ ਡਾਟਾ ਪ੍ਰਬੰਧ ਕਰਨ ਲਈ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਤਿਆਰ ਕਰਨ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦੇ ਹਨ। ਆਮ ਤੌਰ ਤੇ ਉਹਨਾਂ ਦੁਆਰਾ ਐਪਲੀਕੇਸ਼ਨ ਤਿਆਰ ਕਰਨ ਲਈ COBOL ਜਾਂ PL/1 ਆਦਿ ਭਾਸ਼ਾਵਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਡਾਟਾ ਤੇ ਕੰਮ ਕਰਦੇ ਹਨ। ਇਹਨਾਂ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮਜ਼ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵੱਖ-ਵੱਖ ਯੂਜ਼ਰਜ਼ ਆਪਣੀ ਲੋੜੀਂਦੀ ਸੂਚਨਾ ਪ੍ਰਾਪਤ ਕਰ ਸਕਦੇ ਹਨ, ਨਵੀਂ ਸੂਚਨਾ ਤਿਆਰ ਕਰਨ ਸਕਦੇ ਹਨ, ਮੌਜੂਦਾ ਸੂਚਨਾ ਨੂੰ ਮਿਟਾ ਵੀ ਸਕਦੇ ਹਨ ਜਾਂ ਬਦਲ ਸਕਦੇ ਹਨ।

4. **ਐਂਡ ਯੂਜ਼ਰਜ਼ (End-Users):** ਯੂਜ਼ਰਜ਼ ਦੇ ਇਸ ਸਮੂਹ ਵਿੱਚ ਉਹ ਵਿਅਕਤੀ ਸ਼ਾਮਿਲ ਹੁੰਦੇ ਹਨ ਜੋ ਅਸਲ ਵਿੱਚ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੀ ਵਰਤੋਂ ਆਪਣੇ ਕੰਮਾਂ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਕਰਦੇ ਹਨ। ਇਹ ਇੱਕ ਟਰਮੀਨਲ (terminal) ਤੋਂ ਡਾਟਾਬੇਸ ਤੱਕ ਪਹੁੰਚ ਕਰਦੇ ਹਨ। ਐਂਡ ਯੂਜ਼ਰਜ਼ ਦੀਆਂ ਦੋ ਕਿਸਮਾਂ ਹਨ।
   - **ਆਮ ਯੂਜ਼ਰਜ਼/ਸੋਫ਼ਿਸਟਿਕੇਟਿਡ ਯੂਜ਼ਰ (Casual User/Sphisticated Users):** ਆਮ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਡਾਟਾਬੇਸ ਕੁਏਰੀ ਭਾਸ਼ਾ (database query language) ਦੀ ਵਰਤੋਂ ਲਈ ਸਿਖਲਾਈ ਦਿੱਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਯੂਜ਼ਰਜ਼ ਕਿਸੇ ਵੀ ਟਰਮੀਨਲ ਤੋਂ ਕੁਏਰੀ ਦਾਖਲ ਕਰਕੇ ਡਾਟਾ ਤੱਕ ਪਹੁੰਚ ਕਰਦੇ ਹਨ।
   - **ਨੇਵ ਯੂਜ਼ਰਜ਼ (Naive User):** ਨੇਵ ਯੂਜ਼ਰ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮਾਂ ਰਾਹੀਂ ਡਾਟਾ ਤੱਕ ਪਹੁੰਚ ਕਰਦੇ ਹਨ। ਉਨ੍ਹਾਂ ਲਈ ਕਈ ਵਿਸ਼ੇਸ਼ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਲਿਖੇ ਗਏ ਹੁੰਦੇ ਹਨ। ਉਹਨਾਂ ਨੂੰ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੀ ਕੁਏਰੀ ਭਾਸ਼ਾ ਦੀ ਜਾਣਕਾਰੀ ਹੋਣ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਹੁੰਦੀ।

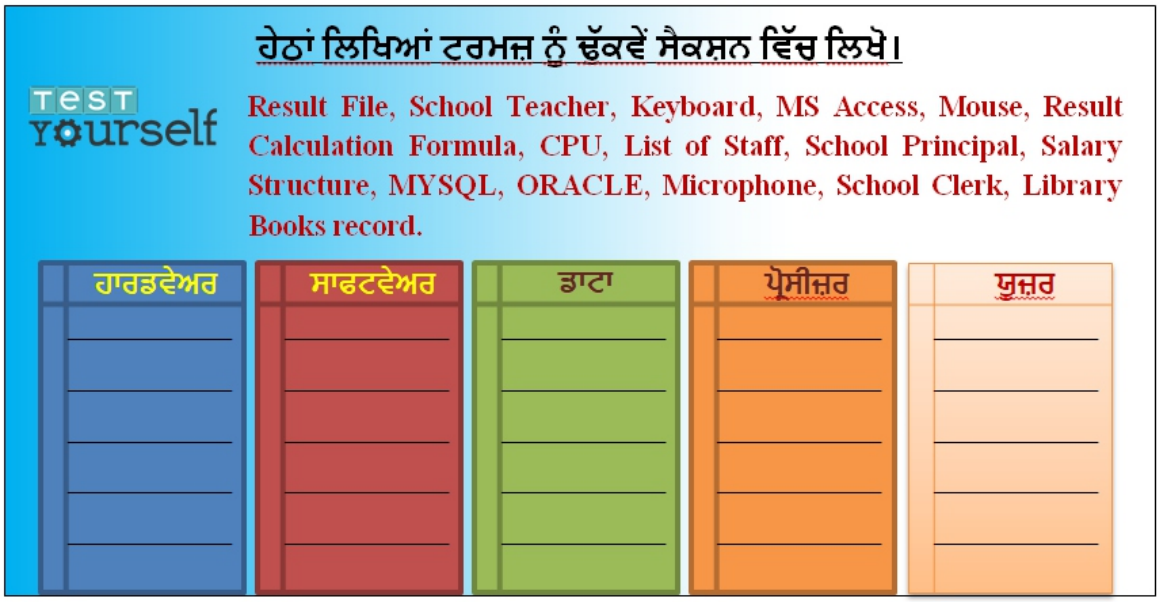
ਨੋਟ: ਉਦਾਹਰਨ ਵਜੋਂ ਜੇਕਰ ਸਾਡੇ ਕੋਲ ਡਾਟਾਬੇਸ ਲਈ ਇੱਕ ਸਕੂਲ ਹੈ ਤਾਂ ਸਾਰੇ ਵਿਦਿਆਰਥੀਆਂ ਨੂੰ ਨਿਰਪੱਖ ਉਪਭੋਗਤਾ ਮੰਨਿਆ ਜਾ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਉਹ ਆਪਣੀਆਂ ਸੇਵਾਵਾਂ ਜਿਵੇਂ ਕਿ ਨਤੀਜਾ, ਹਾਜ਼ਰੀ ਆਦਿ ਲਈ ਡਾਟਾਬੇਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਨ। ਸਾਰੇ ਅਧਿਆਪਕਾਂ/ਵਿਦਿਆਰਥੀਆਂ ਨੂੰ ਆਮ ਉਪਭੋਗਤਾਵਾਂ ਵਜੋਂ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਉਹ ਡਾਟਾਬੇਸ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸੰਚਾਲਿਤ ਕਰਦੇ ਹਨ। ਸਕੂਲ ਦੇ ਪ੍ਰਿੰਸੀਪਲ/ਹੈੱਡ ਮਾਸਟਰ ਨੂੰ ਡੀਬੀਏ ਵਜੋਂ ਜਾਣਿਆ ਜਾ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਉਹ ਸਾਰੇ ਕਾਰਜਾਂ ਦੀ ਨਿਗਰਾਨੀ ਕਰਦਾ ਹੈ।

5.2.2 ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੇ ਹਿੱਸੇ (Components of Database System Environment):

ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਸ਼ਬਦ ਡਾਟਾਬੇਸ ਨਾਲ ਸੰਬੰਧਤ ਵੱਖ ਵੱਖ ਭਾਗਾਂ ਦੇ ਸੰਗਠਨ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇਹ ਭਾਗ ਡਾਟਾਬੇਸ ਵਾਤਾਵਰਣ ਦੇ ਅੰਦਰ ਡਾਟਾ ਦੇ ਇਕੱਠ (collection), ਸਟੋਰੇਜ (storage), ਪ੍ਰਬੰਧਨ (management) ਅਤੇ ਵਰਤੋਂ ਨੂੰ ਪਰਿਭਾਸ਼ਤ ਅਤੇ ਕੰਟਰੋਲ ਕਰਦੇ ਹਨ। ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਵਾਤਾਵਰਨ ਦੇ ਇਹ ਭਾਗ ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਹੋ ਸਕਦੇ ਹਨ:-

1. **ਹਾਰਡਵੇਅਰ:** ਇਹ ਭਾਗ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੇ ਸਾਰੇ ਭੌਤਿਕ ਯੰਤਰਾਂ ਦਾ ਸਮੂਹ ਹੁੰਦੇ ਹਨ। ਇਸ ਭਾਗ ਦਾ ਸਭ ਤੋਂ ਮਹੱਤਵਪੂਰਨ ਅਤੇ ਜ਼ਰੂਰੀ ਹਿੱਸਾ ਕੰਪਿਊਟਰ ਹੈ। ਇਹ ਇੱਕ ਮਾਈਕ੍ਰੋਪ੍ਰੋਸੈਸਰ (microprocessor), ਇੱਕ ਮਿਨੀਕੰਪਿਊਟਰ (minicomputer) ਜਾਂ ਇਕ ਮੇਨਫ੍ਰੇਮ (mainframe) ਕੰਪਿਊਟਰ ਹੋ ਸਕਦਾ ਹੈ। ਇਸ ਵਿੱਚ ਕੀਬੋਰਡ, ਮਾਊਸ, ਮਾਡਮ, ਪ੍ਰਿੰਟਰ ਆਦਿ ਵਰਗੇ ਇਨਪੁੱਟ ਅਤੇ ਆਉਟਪੁੱਟ ਉਪਕਰਨ ਵੀ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ।
2. **ਸਾਫਟਵੇਅਰ:** ਸਾਫਟਵੇਅਰ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਵਿੱਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਪ੍ਰੋਗਰਾਮਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇਸ ਵਿੱਚ ਆਪ੍ਰੇਟਿੰਗ (operating) ਸਿਸਟਮ, DBMS ਸਾਫਟਵੇਅਰ, ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਅਤੇ ਯੂਟੀਲਈ (utility) ਸਾਫਟਵੇਅਰ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ।
3. **ਯੂਜ਼ਰਜ਼ (Users):** ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੇ ਇਸ ਭਾਗ ਵਿੱਚ ਡਾਟਾਬੇਸ ਨਾਲ ਸੰਬੰਧਤ ਸਾਰੇ ਉਪਭੋਗਤਾ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਅਸੀਂ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਵਿੱਚ ਮੌਜੂਦ ਸਾਰੇ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਪੰਜ ਕਿਸਮਾਂ ਵਿੱਚ ਵੰਡ ਸਕਦੇ ਹਾਂ:
   - ਸਿਸਟਮ ਪ੍ਰਸ਼ਾਸਕ (System Administrators)
   - ਡਾਟਾ ਮਾਡਲਰ (Data Modelers)
   - ਡਾਟਾਬੇਸ ਐਡਮੀਨੀਸਟ੍ਰੇਟਰ (Database Administrators)
   - ਸਿਸਟਮ ਵਿਸ਼ਲੇਸ਼ਕ ਅਤੇ ਪ੍ਰੋਗਰਾਮਰ (System Analysts and Programmers)
   - ਐਂਡ ਯੂਜ਼ਰਜ਼ (End Users)
4. **ਪ੍ਰੋਸੀਜ਼ਰਜ਼ (Procedures):** ਪ੍ਰੋਸੀਜ਼ਰਜ਼ ਹਦਾਇਤਾਂ ਅਤੇ ਨਿਯਮ ਹੁੰਦੇ ਹਨ ਜੋ ਡਾਟਾਬੇਸ ਸਿਸਟਮ ਦੇ ਡਿਜ਼ਾਈਨ ਅਤੇ ਵਰਤੋਂ ਦਾ ਪ੍ਰਬੰਦਨ ਕਰਦੇ ਹਨ।
5. **ਡਾਟਾ (Data):** ਸ਼ਬਦ "ਡਾਟਾ" ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਤੱਥਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਡਾਟਾ ਕਿਸੇ ਵੀ ਸੂਚਨਾ ਲਈ ਇੱਕ ਕੱਚੇ ਮਾਲ ਦੇ ਤੌਰ ਤੇ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਏਕੀਕਿਰਤ (integrated) ਅਤੇ ਸਾਂਝਾ (shared) ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਭਾਗ ਡਾਟਾਬੇਸ ਦਾ ਮੂਲ ਹੈ।

ਐਕਟੀਵਿਟੀ 5.3



5.3 ਡਾਟਾ ਮਾਡਲ (DATA MODELS)

ਡਾਟਾ ਮਾਡਲ ਨੂੰ ਡਾਟਾਬੇਸ ਦੀ ਲਾਜ਼ੀਕਲ ਬਣਤਰ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਨੂੰ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਵੱਖ ਵੱਖ ਡਾਟਾ ਦੇ ਵਿਚਕਾਰ ਸਬੰਧਾਂ ਅਤੇ ਡਾਟਾ ਪਾਬੰਦੀਆਂ ਦੇ ਵੱਖ-ਵੱਖ ਨਿਯਮਾਂ ਅਤੇ ਵਿਵਸਥਿਕ ਕਰਨ ਨਾਲ ਸੰਬੰਧਤ ਸੰਕਲਪਾਂ (concepts) ਦੇ ਸਮੂਹ ਵਜੋਂ ਵੀ ਪ੍ਰਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਇੱਕ ਡਾਟਾ-ਮਾਡਲ 'ਅਸਲ ਸੰਸਾਰ' ਦੀਆਂ ਮੌਜੂਦਾ ਵਸਤੂਆਂ (objects), ਘਟਨਾਵਾਂ (events) ਅਤੇ ਐਸੋਸੀਏਸ਼ਨਾਂ (associations) ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇੱਕ ਡਾਟਾ ਮਾਡਲ ਆਪਣੇ ਆਪ ਵਿੱਚ ਇੱਕ ਡਾਟਾ ਸੰਗਠਨ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।

ਇੱਕ ਡਾਟਾ ਮਾਡਲ ਵਿੱਚ ਤਿੰਨ ਭਾਗ ਹੁੰਦੇ ਹਨ:

1. **ਢਾਂਚਾਗਤ ਭਾਗ (A structural part):** ਇਸ ਵਿੱਚ ਡਾਟਾਬੇਸ ਬਣਾਉਣ ਲਈ ਨਿਯਮਾਂ ਦਾ ਇੱਕ ਸਮੂਹ ਹੁੰਦਾ ਹੈ।
2. **ਲਾਗੂਕਰਨ ਨਿਰਧਾਰਿਤ ਕਰਨ ਵਾਲਾ ਭਾਗ (A manipulative part):** ਇਹ ਉਹਨਾਂ ਸਾਰੇ ਓਪਰੇਸ਼ਨਜ਼ ਦੀਆਂ ਕਿਸਮਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦਾ ਹੈ ਜੋ ਡਾਟਾ ਤੇ ਲਾਗੂ ਕੀਤੇ ਜਾਣ ਯੋਗ ਹੁੰਦੇ ਹਨ।
3. **ਇੱਕਜੁੱਟਤਾ ਨਿਯਮਾਂ ਦਾ ਸੈੱਟ (A set of integrity rules):** ਇਹ ਯਕੀਨੀ ਬਣਾਉਂਦਾ ਹੈ ਕਿ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਾਰਾ ਡਾਟਾ ਸਹੀ ਰਹੇ।

5.3.1 ਰਿਕਾਰਡ (RECORD) ਆਧਾਰਿਤ ਲਾਜ਼ੀਕਲ ਮਾਡਲ

ਰਿਕਾਰਡ ਅਧਾਰਤ ਡਾਟਾ ਮਾਡਲ ਦੀ ਵਰਤੋਂ ਲੌਜੀਕਲ ਅਤੇ ਬਾਹਰੀ ਪੱਧਰ (external levels) ਤੇ ਡਾਟਾ ਦਾ ਵਰਣਨ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਮਾਡਲ ਦੀ ਵਰਤੋਂ ਡਾਟਾਬੇਸ ਦੀ ਸਮੁੱਚੀ ਲਾਜ਼ੀਕਲ ਬਣਤਰ ਨੂੰ ਤਿਆਰ ਕਰਨ ਅਤੇ ਦਰਸਾਉਣ ਲਈ ਹੁੰਦੀ ਹੈ। ਇਹ ਮਾਡਲ ਡਾਟਾਬੇਸ ਦੇ ਲਾਗੂਕਰਨ (implementation) ਦਾ ਉੱਚ-ਪੱਧਰੀ ਵਰਣਨ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਸ ਮਾਡਲ ਵਿੱਚ ਡਾਟਾਬੇਸ ਨੂੰ ਰਿਕਾਰਡਜ਼ ਦੇ ਇੱਕ ਪਹਿਲਾਂ ਤੋਂ ਨਿਰਦਾਰਿਤ ਲਾਜੀਕਲ ਢਾਂਚੇ ਦੇ ਰੂਪ ਵਿੱਚ ਤਿਆਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਹਰੇਕ ਰਿਕਾਰਡ ਦੀ ਕਿਸਮ ਕਈ ਫੀਲਡਜ਼ (fields) ਜਾਂ ਐਟਰੀਬਿਊਟਜ਼ (attributes) ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਸਭ ਤੋਂ ਵੱਧ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਰਿਕਾਰਡ ਅਧਾਰਤ ਡਾਟਾ ਮਾਡਲ ਨਿਮਨ ਦਰਸਾਏ ਅਨੁਸਾਰ ਹਨ:

1. ਹਾਈਰੈਰਕੀਕਲ (Hierarchical) ਮਾਡਲ – ਟਰੀ ਸਟ੍ਰਕਚਰ
2. ਨੈੱਟਵਰਕ ਮਾਡਲ – ਪਲੇਕਸ ਸਟ੍ਰੱਕਚਰ
3. ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ – ਨਾਰਮਲਾਈਜ਼ (Normalize) ਸਟ੍ਰਕਚਰ

5.3.1.1 ਹਾਈਰੈਰਕੀਕਲ (Hierarchical) ਮਾਡਲ:

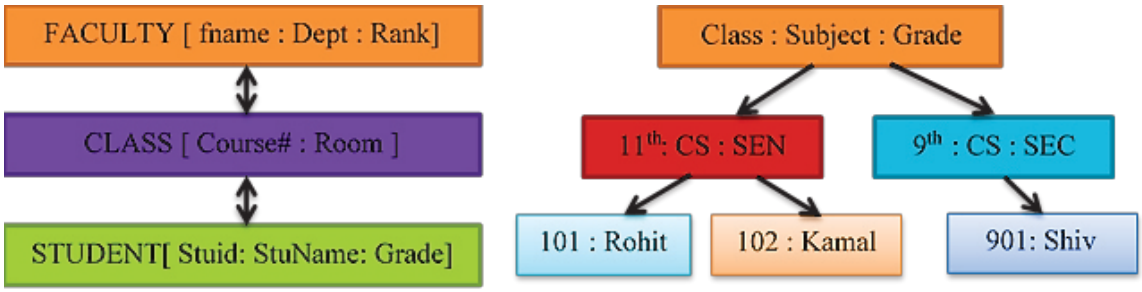
ਹਾਈਰੈਰਕੀਕਲ ਮਾਡਲ ਤਿੰਨਾਂ ਰਿਕਾਰਡ-ਅਧਾਰਿਤ ਡਾਟਾ ਮਾਡਲਜ਼ ਵਿੱਚੋਂ ਸਭ ਤੋਂ ਪੁਰਾਣਾ ਮਾਡਲ ਹੈ। ਹਾਈਰੈਰਕੀਕਲ ਮਾਡਲ ਟ੍ਰੀ ਸਟ੍ਰਕਚਰ ਨੂੰ ਇਸਦੀ ਮੂਲ ਬਣਤਰ ਵਜੋਂ ਵਰਤਦਾ ਹੈ। ਇੱਕ ਟ੍ਰੀ ਇੱਕ ਡਾਟਾ ਹੁੰਦਾ ਹੈ ਜੋ ਕਿ ਹਾਈਰੈਰਕੀਕਲ ਸਟ੍ਰਕਚਰ ਵਿੱਚ ਬਣੀ ਇੱਕ ਲੜੀ ਹੁੰਦੀ ਹੈ। ਉੱਚਤਮ ਪੱਧਰ ਤੇ ਮੌਜੂਦ ਨੋਡ ਨੂੰ "ਰੂਟ" ਨੋਡ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਹਾਈਰੈਰਕੀਕਲ ਮਾਡਲ ਦੋ ਮੁੱਖ ਡਾਟਾ ਸਟ੍ਰਕਚਰਿੰਗ ਸੰਕਲਪ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ:

- ਰਿਕਾਰਡਜ਼
- ਪੇਰੇਂਟ-ਚਾਈਲਡ ਰਿਲੇਸ਼ਨ।

ਰਿਕਾਰਡ: ਇੱਕ ਰਿਕਾਰਡ ਫੀਲਡ ਮੁੱਲਾਂ ਦਾ ਸੰਗ੍ਰਹਿ ਹੁੰਦਾ ਹੈ। ਇਹ ਕਿਸੇ ਐਂਟੀਟੀ (entity) ਜਾਂ ਰਿਲੇਸ਼ਨ (relationship) ਬਾਰੇ ਜਾਣਕਾਰੀ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ।

ਪੇਰੇਂਟ-ਚਾਈਲਡ ਰਿਲੇਸ਼ਨ: ਇਸ ਮਾਡਲ ਵਿੱਚ ਇੱਕ ਪੇਰੇਂਟ-ਚਾਈਲਡ ਰਿਲੇਸ਼ਨ ਦੇ ਰੂਪ ਵਿੱਚ ਦੋ ਜਾਂ ਦੋ ਤੋਂ ਵੱਧ ਰਿਕਾਰਡ ਵਿਚਕਾਰ ਸਬੰਧ ਸਥਾਪਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਸ ਨੂੰ 1:N ਰਾਹੀਂ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। 1 ਰਾਹੀਂ ਦਰਸ਼ਾਏ ਗਏ ਰਿਕਾਰਡ ਦੀ ਕਿਸਮ ਨੂੰ ਪੇਰੇਂਟ ਰਿਕਾਰਡ ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ N ਰਾਹੀਂ ਦਰਸ਼ਾਏ ਗਏ ਰਿਕਾਰਡਜ਼ ਨੂੰ ਚਾਈਲਡ ਰਿਕਾਰਡ ਦੇ ਰੂਪ ਵਿੱਚ ਪੇਸ਼ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

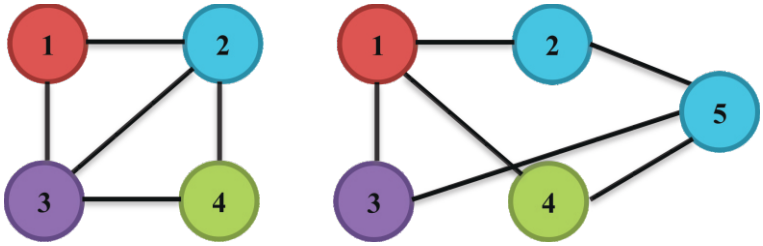
ਇਸ ਤਰ੍ਹਾਂ ਬਣਾਇਆ ਗਿਆ ਪੂਰਾ ਟ੍ਰੀ ਇੱਕ ਇਕਾਈ ਨੂੰ ਦਰਸਾਉਂਦਾ ਇੱਕ ਸਟ੍ਰਕਚਰ ਹੋ ਸਕਦਾ ਹੈ ਜਾਂ ਇੱਕ ਟ੍ਰੀ ਦੇ ਅੰਦਰ ਮੌਜੂਦ ਵੱਖ-ਵੱਖ ਹਿੱਸਿਆਂ ਨੂੰ ਪੇਸ਼ ਕਰਦਾ ਸਟ੍ਰਕਚਰ ਵੀ ਹੋ ਸਕਦਾ ਹੈ।



ਚਿੱਤਰ 5.3 ਹਾਈਰੈਰਕੀਕਲ (Hierarchical) ਮਾਡਲ

5.3.1.2 ਨੈੱਟਵਰਕ ਮਾਡਲ (Network Model):

ਨੈੱਟਵਰਕ ਮਾਡਲ ਨੈੱਟਵਰਕ ਜਾਂ ਪਲੈਕਸ (Plex) ਡਾਟਾ ਸਟ੍ਰਕਚਰ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਇੱਕ ਨੈੱਟਵਰਕ ਮਾਡਲ ਇੱਕ ਗ੍ਰਾਫ ਦੇ ਰੂਪ ਵਿੱਚ ਪੇਸ਼ ਕੀਤਾ ਹੁੰਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਲਿੰਕ ਜਾਂ ਲਾਈਨਾਂ (directed arcs) ਦੁਆਰਾ ਜੁੜੇ ਨੋਡ ਹੁੰਦੇ ਹਨ। ਨੋਡ ਰਿਕਾਰਡ ਕਿਸਮਾਂ ਅਤੇ ਪੁਆਇੰਟਰਾਂ ਦੇ ਲਿੰਕ ਨਾਲ ਮੇਲ ਖਾਂਦੇ ਹਨ। ਨੈੱਟਵਰਕ ਡਾਟਾ ਬਣਤਰ ਵੀ ਇੱਕ ਟ੍ਰੀ ਬਣਤਰ ਵਰਗਾ ਹੀ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ। ਪਰੰਤੂ ਇਸ ਮਾਡਲ ਵਿੱਚ ਚਾਈਲਡ ਨੋਡ ਨਾਲ ਜੁੜੇ ਇੱਕ ਤੋਂ ਵੱਧ ਪੇਰੇਂਟ ਨੋਡ ਹੋ ਸਕਦੇ ਹਨ। ਹੇਠਾਂ ਦਿੱਤੀ ਤਸਵੀਰ ਨੈੱਟਵਰਕ ਮਾਡਲ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ।



ਚਿੱਤਰ 5.4 ਨੈੱਟਵਰਕ ਡਾਟਾ ਮਾਡਲ

5.3.1.3 ਰਿਲੇਸ਼ਨਲ ਡਾਟਾ ਮਾਡਲ:

ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ ਡਾਟਾ ਅਤੇ ਵੱਖ-ਵੱਖ ਡਾਟਾ ਦੇ ਆਪਣੀ ਸਬੰਧਾਂ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਟੇਬਲਾਂ ਦੇ ਸਮੂਹ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਇੱਕ ਟੇਬਲ ਰੋਅਜ਼ ਅਤੇ ਕਾਲਮਜ਼ ਦਾ ਸਮੂਹ ਹੁੰਦਾ ਹੈ। ਹਰੇਕ ਕਾਲਮ ਦਾ ਇੱਕ ਵਿਲੱਖਣ ਨਾਮ ਹੁੰਦਾ ਹੈ। ਟੇਬਲ ਵਿੱਚ ਹਰੇਕ ਰੋਅ ਸਬੰਧਿਤ ਡਾਟਾ ਮੁੱਲਾਂ ਦੇ ਇਕੱਠ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ ਵਿੱਚ ਇੱਕ ਰੋਅ ਨੂੰ ਇੱਕ ਟਪਲ **(tuple)** ਕਿਹਾ ਜਾਂਦਾ ਹੈ, ਇੱਕ ਕਾਲਮ ਹੈਡਰ ਨੂੰ ਇੱਕ ਐਟਰੀਬਿਊਟ **(attribute)** ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਟੇਬਲ ਨੂੰ ਇੱਕ ਰਿਲੇਸ਼ਨ **(relation)** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ, ਹੇਠਾਂ ਦਿੱਤਾ ਚਿੱਤਰ ਵਿਦਿਆਰਥੀ ਨਾਂ ਦੇ ਇੱਕ ਟੇਬਲ ਦੀ ਇੱਕ ਉਦਾਹਰਣ ਦਿਖਾਉਂਦਾ ਹੈ।

ਰਿਲੇਸ਼ਨ: STUDENT

STUID	SNAME	SUBJECT	MARKS
1015	Mary	Math	42
1005	Jones	History	13
1001	Smith	History	90

ਚਿੱਤਰ 5.5: ਰਿਲੇਸ਼ਨਲ ਡਾਟਾ ਮਾਡਲ

ਰਿਲੇਸ਼ਨਲ ਡਾਟਾ ਮਾਡਲ ਦੇ ਲਾਭ:

- ਇਹ ਮਾਡਲ ਸਮਝਣਾ ਅਤੇ ਵਰਤੋਂ ਕਰਨਾ ਆਸਾਨ ਹੁੰਦਾ ਹੈ।
- ਇਹ ਮਾਡਲ ਬਹੁਤ ਲਚਕੀਲਾ ਹੁੰਦਾ ਹੈ।
- ਇਹ ਮਾਡਲ ਸਭ ਤੋਂ ਵਧੇਰੇ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
- ਇਹ ਮਾਡਲ ਐਡ-ਹਾਕ (ad-hoc) ਕੁਏਰੀਜ਼ (queries) ਲਈ ਉੱਤਮ ਸੇਵਾਵਾਂ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ।
- ਇਸ ਮਾਡਲ ਵਿੱਚ ਡਾਟਾ ਤੱਕ ਪਹੁੰਚਣ ਲਈ ਯੂਜ਼ਰ ਨੂੰ ਡਾਟਾ ਦੇ ਸਟੋਰੇਜ ਸਟ੍ਰਕਚਰ ਦੀ ਸਮਝ ਜ਼ਰੂਰੀ ਨਹੀਂ ਹੁੰਦੀ।
- ਇਸ ਮਾਡਲ ਵਿੱਚ ਸੁਰੱਖਿਆ ਨਿਯੰਤਰਣ (Security control) ਅਤੇ ਅਧਿਕਾਰ (authorization) ਨੂੰ ਹੋਰ ਆਸਾਨੀ ਨਾਲ ਲਾਗੂ ਕੀਤਾ ਜਾ ਸਕਦਾਹੈ।
- ਇਸ ਮਾਡਲ ਰਾਹੀਂ ਨਾਰਮਲਾਈਜੇਸ਼ਨ ਨਾਲ ਡਾਟਾ ਸੁਤੰਤਰਤਾ (Data independence) ਵਧੇਰੇ ਆਸਾਨੀ ਨਾਲ ਪ੍ਰਾਪਤ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ ਦੇ ਨੁਕਸਾਨ:

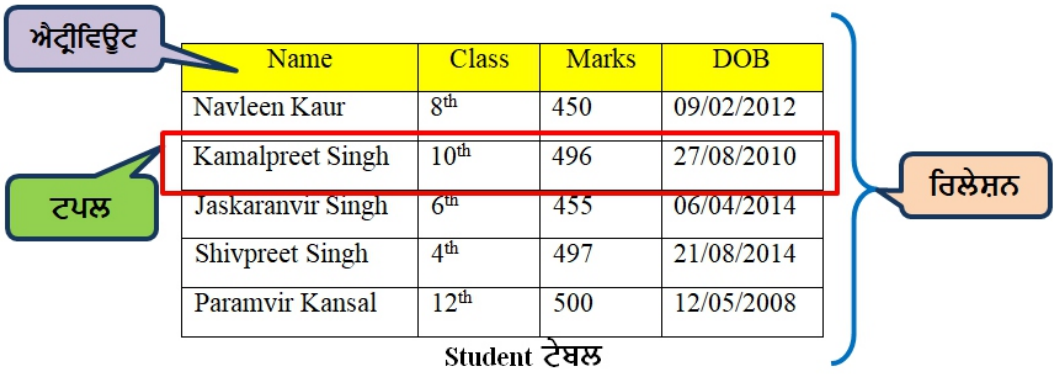
- ਵੱਡੇ ਡਾਟਾਬੇਸ ਲਈ ਇਸ ਕਿਸਮ ਦੇ ਮਾਡਲ ਦੀ ਪੇਸ਼ਕਾਰੀ ਔਖੀ ਹੁੰਦੀ ਹੈ।
- ਪ੍ਰੋਸੈਸਿੰਗ ਲੋੜਾਂ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਇੰਡੈਕਸ (index) ਬਣਾਉਣ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ।
- ਅਸਲ ਫਾਈਲ ਰਿਕਾਰਡ ਪ੍ਰਾਪਤ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਫਾਈਲ ਇੰਡੈਕਸ ਵਿੱਚ ਸਰਚਿੰਗ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜਿਸ ਨਾਲ ਕਾਫੀ ਸਮਾਂ ਬਰਬਾਦ ਹੁੰਦਾ ਹੈ।

5.4 RDBMS ਟਰਮਿਨੋਲੋਜੀ (RDBMS TERMINOLOGY)

RDBMS ਦਾ ਅਰਥ ਹੈ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ। ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਸਭ ਤੋਂ ਵੱਧ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਡਾਟਾਬੇਸ ਹੈ। ਇਸ ਵਿੱਚ ਬਹੁਤ ਸਾਰੇ ਟੇਬਲਜ਼ ਦੇ ਰੂਪ ਵਿੱਚ ਡਾਟਾਬੇਸ ਤਿਆਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਹਰੇਕ ਟੇਬਲ ਦੀ ਆਪਣੀ ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਹੁੰਦੀ ਹੈ। ਇੱਕ ਟੇਬਲ ਰੋਅਜ਼ ਅਤੇ ਕਾਲਮਜ਼ ਦਾ ਸਮੂਹ ਹੁੰਦਾ ਹੈ। ਹਰੇਕ ਕਾਲਮ ਦਾ ਇੱਕ ਵਿਲੱਖਣ ਨਾਮ ਹੁੰਦਾ ਹੈ। ਹਰੇਕ ਰੋਅ ਸੰਬੰਧਿਤ ਡਾਟਾ ਮੁੱਲਾਂ ਦਾ ਇਕੱਠ ਹੁੰਦੀ ਹੈ। ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ ਵਿੱਚ, ਇੱਕ ਰੋਅ ਨੂੰ ਇੱਕ ਟਪਲ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ, ਇੱਕ ਕਾਲਮ ਹੈਡਰ ਨੂੰ ਇੱਕ ਐਟਰੀਬਿਊਟ ਦੇ ਤੌਰ ਤੇ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਟੇਬਲ ਨੂੰ ਇੱਕ ਰਿਲੇਸ਼ਨ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਟੇਬਲਾਂ ਦੇ ਸਮੂਹ ਹੋਣ ਕਰਕੇ ਹੀ RDBMS ਵਿੱਚ ਡਾਟਾ ਨੂੰ ਆਸਾਨੀ ਨਾਲ ਐਕਸੈਸ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

ਅਸੀਂ RDBMS ਨਾਲ ਸੰਬੰਧਿਤ ਕੁਝ ਸ਼ਬਦਾਂ ਦੀ ਵਿਆਖਿਆ ਇਸ ਤਰ੍ਹਾਂ ਕਰ ਸਕਦੇ ਹਾਂ:

1. **ਐਟਰੀਬਿਊਟ (Attribute):** ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਵਿੱਚ ਕਿਸੇ ਵੀ ਟੇਬਲ ਵਿੱਚ ਸਟੋਰ ਕੀਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਦਾ ਵਰਣਨ ਕਰਨ ਵਾਲੇ ਕਾਲਮਾਂ ਨੂੰ ਐਟਰੀਬਿਊਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜੋ ਕਿ ਸੰਬੰਧਤ ਐਨਟਿਟੀ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦੇ ਹਨ। ਅਸੀਂ DBMS ਵਿੱਚ ਇੱਕ ਟੇਬਲ ਦੇ ਵੱਖਰੇ-ਵੱਖਰੇ ਕਾਲਮਾਂ ਦੇ ਰੂਪ ਵਿੱਚ ਕਈ ਐਟੀਰਬਿਊਟ ਦਰਸਾ ਸਕਦੇ ਹਾਂ। ਹਰੇਕ ਕਾਲਮ ਕਿਸੇ ਵਿਸ਼ੇਸ਼ ਗੁਣ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਅਤੇ ਸਟੋਰ ਕੀਤੇ ਜਾ ਰਹੇ ਮੁੱਲਾਂ ਦੀ ਕਿਸਮ ਦੇ ਅਨੁਸਾਰ ਇਹਨਾਂ ਗੁਣਾਂ ਨੂੰ ਨਿਰਧਾਰਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਇੱਕ ਵਿਦਿਆਰਥੀ ਦੇ ਰੋਲ ਨੰਬਰ ਨੂੰ ਇੱਕ ਸਧਾਰਨ ਐਟਰੀਬਿਊਟ ਮੰਨਿਆ ਜਾ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਇਸਨੂੰ ਅੱਗੇ ਨਹੀਂ ਵੰਡਿਆ ਜਾ ਸਕਦਾ। ਜਨਮ ਮਿਤੀ ਨੂੰ ਇੱਕ ਬਹੁਮੁੱਲ ਐਟਰੀਬਿਊਟ ਕਿਹਾ ਜਾ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਇਸ ਨੂੰ ਦਿਨ, ਮਹੀਨੇ ਅਤੇ ਸਾਲ ਵਿੱਚ ਵੰਡਿਆ ਜਾ ਸਕਦਾ ਹੈ।
2. **ਟਪਲ (Tuple) :** DBMS ਵਿੱਚ ਇੱਕ ਟਪਲ ਮੁੱਲਾਂ ਦਾ ਇੱਕ ਸਮੂਹ ਹੁੰਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਕਿਸੇ ਵੀ ਦੋ ਵਿਲੱਖਣ ਆਈਟਮਾਂ ਦਾ ਇੱਕੋ ਨਾਮ ਨਹੀਂ ਹੁੰਦਾ। ਇੱਕ ਟਪਲ ਵਿੱਚ ਇੱਕ ਵਿਸ਼ੇਸ਼ ਐਂਟੀਟੀ ਦੀ ਸਾਰੀ ਜਾਣਕਾਰੀ ਸ਼ਾਮਲ ਹੁੰਦੀ ਹੈ। ਟੇਬਲ ਵਿੱਚ ਫੀਲਡਜ਼ ਅਤੇ ਟਪਲਜ਼ ਸ਼ਾਮਿਲ ਹੁੰਦੇ ਹਨ। ਇੱਕ ਟੇਬਲ ਵਿੱਚ ਇੱਕ ਟਪਲ ਕਿਸੇ ਵੀ ਐਂਟੀਟੀ ਨਾਲ ਸੰਬੰਧਤ ਰੋਅ ਦੇ ਡਾਟਾ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।
3. **ਰਿਲੇਸ਼ਨ (Relation):** ਇੱਕ ਰਿਲੇਸ਼ਨ ਇੱਕ ਟੇਬਲ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਰੋਅਜ਼ ਅਤੇ ਕਾਲਮਜ਼ ਦੇ ਰੂਪ ਵਿੱਚ ਕੁਝ ਮੁੱਲ ਸਟੋਰ ਕੀਤੇ ਹੁੰਦੇ ਹਨ, ਜੱਥੇ ਇੱਕ ਰੋਅ ਜਾਂ ਟਪਲ ਸੰਬੰਧਿਤ ਡਾਟਾ ਮੁੱਲਾਂ ਦਾ ਸੰਗ੍ਰਹਿ ਹੁੰਦੀ ਹੈ। ਹਰੇਕ ਰੋਅ, ਕਾਲਮ ਅਤੇ ਟੇਬਲ ਦੇ ਅੰਦਰ ਸਟੋਰ ਕੀਤਾ ਹਰੇਕ ਮੁੱਲ ਟੇਬਲ ਦੇ ਨਾਮ ਨੂੰ ਨਿਰਦਾਰਤ ਕਰਨ ਲਈ ਮਹੱਤਵਪੂਰਨ ਹੰਦਾ ਹੈ। ਹਾਲਾਂਕਿ ਡਾਟਾ ਨੂੰ RDBMS ਵਿੱਚ ਟੇਬਲਜ਼ ਵਿੱਚ ਵਿਵਸਥਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਪਰੰਤੂ ਡਾਟਾ ਸਟੋਰੇਜ ਡਾਟਾ ਦੇ ਲਾਜ਼ੀਕਲ ਢਾਂਚੇ ਤੇ ਨਿਰਭਰ ਨਹੀਂ ਕਰਦਾ ਹੈ।
4. **ਡੋਮੇਨ (Domain):** ਇੱਕ ਰਿਲੇਸ਼ਨ ਦੇ ਕਿਸੇ ਐਟਰੀਬਿਊਟ ਲਈ ਸਾਰੇ ਸਵੀਕਾਰ ਕੀਤੇ ਜਾ ਸਕਣ ਵਾਲੇ ਮੁੱਲਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਨੂੰ ਐਟਰੀਬਿਊਟ ਦਾ ਡੋਮੇਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਡੋਮੇਨ ਨੂੰ ਡਾਟਾ ਕਿਸਮ ਦੇ ਰੂਪ ਵਿੱਚ ਵੀ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜੋ ਐਟਰੀਬਿਊਟ ਦੇ ਮੁੱਲਾਂ ਦੀ ਕਿਸਮ ਦਾ ਵਰਣਨ ਕਰਦਾ ਹੈ। ਇਹ ਸਾਰੇ ਮੁੱਲ ਹੀ ਉਸ ਕਾਲਮ ਵਿੱਚ ਭਰੇ ਜਾ ਸਕਦੇ ਹੁੰਦੇ ਹਨ। ਉਦਾਹਰਨ ਲਈ {“Male”, “Female”} ਨੂੰ Gender ਨਾਮ ਦੇ ਐਟਰੀਬਿਊਟ ਲਈ ਇੱਕ ਡੋਮੇਨ ਵਜੋਂ ਦਰਸ਼ਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।
5. **ਡਿਗਰੀ (Degree)**: ਕਿਸੇ ਰਿਲੇਸ਼ਨ/ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਐਟਰੀਬਿਊਟਜ਼/ਕਾਲਮਜ਼ ਦੀ ਕੁੱਲ ਗਿਣਤੀ ਨੂੰ ਉਸ ਟੇਬਲ ਦੀ ਡਿਗਰੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਟੇਬਲ ਦੀ ਡਿਗਰੀ ਨੂੰ d(R) ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।
6. **ਕਾਰਡੀਨੈਲਿਟੀ (Cardianlity):** ਕਿਸੇ ਰਿਲੇਸ਼ਨ/ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਟਪਲਜ਼/ਰੋਅਜ਼ ਦੀ ਕੁੱਲ ਗਿਣਤੀ ਨੂੰ ਉਸ ਟੇਬਲ ਦੀ ਕਾਰਡੀਨੈਲਿਟੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਟੇਬਲ ਦੀ ਕਾਰਡੀਨੈਲਿਟੀ ਨੂੰ ।R। ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।



Name	Class	Marks	DOB
Navleen Kaur	8th	450	09/02/2012
Kamalpreet Singh	10th	496	27/08/2010
Jaskaranvir Singh	6th	455	06/04/2014
Shivpreet Singh	4th	497	21/08/2014
Paramvir Kansal	12th	500	12/05/2008

Student ਟੇਬਲ

ਚਿੱਤਰ 5.6 ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ ਟਰਮਜ਼

Name	Class	Marks	DOB	Gender
Navleen Kaur	8th	450	09/02/2012	Female
Kamalpreet Singh	10th	496	27/08/2010	Male
Jaskaranvir Singh	6th	455	06/04/2014	Male
Paramvir Kansal	12th	500	12/05/2008	Male

Cardinality

Degree

ਚਿੱਤਰ 5.7 ਟੇਬਲ ਦੀ ਡਿਗਰੀ ਅਤੇ ਕਾਰਡੀਨੈਲਿਟੀ

"Gender" ਐਟਰੀਬਿਊਟ ਦਾ ਡੋਮੇਨ: Male, Female (ਕਿਉਂਕਿ ਕੇਵਲ Male, Female ਹੀ ਮਨਜ਼ੂਰ ਹਨ)
"Student ਟੇਬਲ" ਰਿਲੇਸ਼ਨ ਦੀ ਡਿਗਰੀ: 5 (ਕਿਉਂਕਿ ਇੱਥੇ 5 ਕਾਲਮਜ਼/ਐਟਰੀਬਿਊਟਜ਼ ਹਨ)
"ਚਿੱਤਰ 5.7 ਦੇ ਟੇਬਲ" ਰਿਲੇਸ਼ਨ ਦੀ ਕਾਰਡੀਨਲਿਟੀ: 4 (ਕਿਉਂਕਿ ਟੇਬਲ ਵਿੱਚ 4 ਰੋਅਜ਼/ਟਪਲਜ਼ ਹਨ)

Activity No: 5.4 ਦਿਤੇ ਗਏ Teacher Table ਦੇ ਅਨੁਸਾਰ ਹੇਠਾਂ ਲਿਖੇ ਸਵਾਲਾਂ ਦੇ ਜਵਾਬ ਦਿਓ।

Teacher ID	Name	SUBJECT	Salary
T1	Keshav	Math	18000
T2	Navleen	Hindi	15000
T7	Karan	English	19000

TEST Yourself

- ਇਸ ਰਿਲੇਸ਼ਨ ਦੀ ਕਾਰਡੀਨੈਲਿਟੀ ਕੀ ਹੈ?
- "SUBJECT" ਕਾਲਮ ਦਾ ਡੋਮੇਨ ਕੀ ਹੋ ਸਕਦਾ ਹੈ?
- ਇਸ ਰਿਲੇਸ਼ਨ ਦੀ ਡਿਗਰੀ ਪਤਾ ਕਰੋ।
- ਰਿਲੇਸ਼ਨ ਦਾ ਨਾਂ ਕੀ ਹੈ?
- ਰਿਲੇਸ਼ਨ ਵਿੱਚ ਕਿੰਨੇ ਟਪਲਜ਼ ਹਨ?
- ਰਿਲੇਸ਼ਨ ਦੇ ਐਟ੍ਰੀਵਿਊਟਜ਼ ਦੇ ਨਾਂ ਦੱਸੋ:

5.4.1 DBMS ਵਿਚ ਕੀਅਜ਼ (Keys in DBMS):

ਅਸੀਂ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਇੱਕਜੁੱਟ (Integrity) ਨਿਯਮਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ DBMS ਵਿੱਚ ਕੀਅਜ਼ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਇੱਕ ਟੇਬਲ ਵੱਖ-ਵੱਖ ਸੰਬੰਧਤ ਰਿਕਾਰਡਾਂ ਦੇ ਸਮੂਹ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇਸ ਲਈ ਇੱਕ ਟੇਬਲ ਵਿੱਚ ਹਜ਼ਾਰਾਂ ਰਿਕਾਰਡ ਹੋ ਸਕਦੇ ਹਨ ਅਤੇ ਇਹਨਾਂ ਵਿੱਚੋਂ ਕੁਝ ਡੁਪਲੀਕੇਟ (duplicated) ਵੀ ਹੋ ਸਕਦੇ ਹਨ। ਇਸ ਤਰ੍ਹਾਂ ਸਾਨੂੰ ਇੱਕ ਅਜਿਹੇ ਤਰੀਕੇ ਦੀ ਜ਼ਰੂਰਤ ਹੈ ਜਿਸ ਵਿੱਚ ਇਹਨਾਂ ਸਾਰੇ ਰਿਕਾਰਡਾਂ ਦੀ ਵਿਲੱਖਣਤਾ ਅਤੇ ਵੱਖਰੇ ਤੌਰ ਤੇ ਪਛਾਣ ਕਾਇਮ ਰੱਖੀ ਜਾ ਸਕੇ। ਜਿਸ ਤੋਂ ਭਾਵ ਹੈ ਕਿ ਡਾਟਾ ਡੁਪਲੀਕੇਸ਼ਨ ਦੀ ਪਰੇਸ਼ਾਨੀ ਕੀਅਜ਼ ਦੀ ਮਦਦ ਨਾਲ ਦੂਰ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

DBMS ਵਿੱਚ ਕੀਅਜ਼ ਕਈ ਐਟਰੀਬਿਊਟਜ਼ (ਜਾਂ ਕਾਲਮਾਂ) ਦਾ ਸੁਮੇਲ ਹੋ ਸਕਦੀਆਂ ਹਨ ਜਾਂ ਸਿਰਫ਼ ਇੱਕ ਹੀ ਐਟਰੀਬਿਊਟ ਤੇ ਨਿਰਭਰ ਹੋ ਸਕਦੀਆਂ ਹਨ। ਕੀਅਜ਼ ਦਾ ਮੁੱਖ ਉਦੇਸ਼ ਹਰ ਰਿਕਾਰਡ ਨੂੰ ਆਪਣੀ ਵਿਲੱਖਣ ਪਛਾਣ ਪ੍ਰਦਾਨ ਕਰਨਾ ਹੈ। ਅਸੀਂ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੀਆਂ ਕੀਅਜ਼ ਨੂੰ ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਦਰਸ਼ਾ ਸਕਦੇ ਹਾਂ।

1. **ਪ੍ਰਾਇਮਰੀ ਕੀਅ (Primary Key):** ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਇੱਕ ਕਾਲਮ ਜਾਂ ਇੱਕ ਟੇਬਲ ਦੇ ਇੱਕ ਤੋਂ ਵੱਧ ਕਾਲਮਾਂ ਦੇ ਸਮੂਹ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ ਜੋ ਸਾਰੇ ਰਿਕਾਰਡਜ਼ ਨੂੰ ਵਿਲੱਖਣ ਰੂਪ ਵਿੱਚ ਪਛਾਨਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੀ ਹੈ। ਇੱਕ ਟੇਬਲ ਵਿੱਚ ਸਿਰਫ਼ ਇੱਕ ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਹੋ ਸਕਦੀ ਹੈ। ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਦੇ ਸਾਰੇ ਮੁੱਲ ਵਿਲੱਖਣ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ ਅਤੇ ਕੋਈ ਦੁਹਰਾਓ ਨਹੀਂ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਅਸੀਂ **EmpID, RollNo, BookID** ਆਦਿ ਨੂੰ ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਐਟਰੀਬਿਊਟਜ਼ ਦੇ ਕੁਝ ਉਦਾਹਰਣ ਵਜੋਂ ਦਰਸ਼ਾ ਸਕਦੇ ਹਾਂ।
2. **ਫੌਰਨ ਕੀਅ (Foreign Key):** ਅਸੀਂ ਕਿਸੇ ਵੀ ਦੋ ਵੱਖ-ਵੱਖ ਟੇਬਲਾਂ ਵਿਚਕਾਰ ਸਬੰਧ ਸਥਾਪਤ ਕਰਨ ਲਈ ਇੱਕ ਫੌਰਨ ਕੀਅ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਫੌਰਨ ਕੀਅ ਨੂੰ ਰੈਫਰੈਂਸ਼ੀਅਲ ਟੇਬਲ ਦੀ ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਨਾਲ ਮੇਲ ਕਰਨ ਲਈ ਕਾਲਮ/ਕਾਲਮਾਂ ਦੇ ਸੈੱਟ ਵਿੱਚ ਮੌਜੂਦ ਹਰੇਕ ਮੁੱਲ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਇੱਕ ਫੌਰਨ ਕੀਅ ਡਾਟਾ ਦੇ ਨਾਲ-ਨਾਲ ਕਿਸੇ ਦੂਸਰੇ ਟੇਬਲ ਵਿੱਚ ਇਕਸਾਰਤਾ ਨੂੰ ਬਣਾਈ ਰੱਖਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੀ ਹੈ। ਉਦਾਹਰਨ ਲਈ, Student Table ਨਾਂ ਦੇ ਟੇਬਲ ਦਾ ਕੋਈ ਵੀ ਰਿਕਾਰਡ Library ਟੇਬਲ ਵਿੱਚ ਇਸਦੇ ਨਾਲ ਸੰਬੰਧਤ ਐਂਟਰੀਆਂ ਰੱਖ ਸਕਦਾ ਹੈ। ਕਿਸੇ ਵੀ ਨਵੇਂ ਵਿਦਿਆਰਥੀ ਦੀ Library ਟੇਬਲ ਵਿੱਚ ਕਿਸੇ ਵੀ ਪੁਸਤਕ ਦੀ ਐਂਟਰੀ ਨਹੀਂ ਹੋ ਸਕਦੀ ਜਦੋਂ ਤੱਕ ਉਹੀ ਵਿਦਿਆਰਥੀ Student ਟੇਬਲ ਵਿੱਚ ਰਜਿਸਟਰ ਨਹੀਂ ਹੁੰਦਾ।

5.5 SQL ਨਾਲ ਜਾਣ ਪਛਾਣ (INTRODUCTION TO SQL)

SQL ਦਾ ਅਰਥ ਹੈ ਸਟ੍ਰਕਚਰਡ ਕੁਐਰੀ ਲੈਂਗੁਐਜ (Structure Query Language)। ਇਹ ਭਾਸ਼ਾ ਸਿੱਖਣੀ ਬਹੁਤ ਆਸਾਨ ਹੈ ਕਿਉਂਕਿ ਇਹ ਆਪਣੀਆਂ ਕਮਾਂਡਜ਼ ਵਿੱਚ ਆਮ ਅੰਗਰੇਜ਼ੀ ਕੀਵਰਡਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੀ ਹੈ। ਰਿਲੇਸ਼ਨਲ ਡਾਟਾ ਮਾਡਲ ਦੇ ਆਧਾਰ ਤੇ 1970 ਵਿੱਚ SQL ਦੀ ਖੋਜ ਕੀਤੀ ਗਈ ਸੀ। ਇਸਨੂੰ ਸ਼ੁਰੂ ਵਿੱਚ ਸਟ੍ਰਕਚਰਡ ਇੰਗਲਿਸ਼ ਕਿਊਰੀ ਲੈਂਗੁਏਜ (SEQUEL) ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਸੀ। ਇਸ ਸ਼ਬਦ ਨੂੰ ਬਾਅਦ ਵਿੱਚ SQL ਵਿੱਚ ਛੋਟਾ ਕਰ ਦਿੱਤਾ ਗਿਆ ਸੀ।

ਇਹ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਨੂੰ ਐਕਸੈਸ ਕਰਨ ਅਤੇ ਲੋੜੀਂਦੇ ਬਦਲਾਵ ਕਰਨ ਲਈ ਇੱਕ ਆਮ ਭਾਸ਼ਾ ਹੈ। ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਇਸ ਅਧਿਆਇ ਵਿੱਚ ਪੜ੍ਹਿਆ ਹੈ, ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਸਾਰੇ ਡਾਟਾ ਨੂੰ ਟੇਬਲ ਦੇ ਰੂਪ ਵਿੱਚ ਸਟੋਰ ਕਰਦਾ ਹੈ। ਅਸੀਂ ਡਾਟਾਬੇਸ ਦੇ ਰਿਲੇਸ਼ਨਲ ਟੇਬਲਾਂ ਵਿਚ ਜਾਣਕਾਰੀ ਨੂੰ ਸਟੋਰ ਕਰਨ, ਅੱਪਡੇਟ ਕਰਨ, ਡਲੀਟ ਕਰਨ, ਸਰਚ ਕਰਨ ਅਤੇ ਲੋੜ ਅਨੁਸਾਰ ਡਾਟਾ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ SQL ਕਮਾਂਡਜ਼ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। MySQL, SQL ਸਰਵਰ, MS Access, Oracle, Sybase, Informix, Postgres, ਆਦਿ ਆਮ ਤੌਰ 'ਤੇ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਦੀਆਂ ਉਦਾਹਰਣਾਂ ਹਨ ਜਿੱਥੇ SQL ਦੀ ਵਰਤੋਂ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਸਟੋਰ ਕਰਨ ਅਤੇ ਤਬਦੀਲ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। SQL ਨੂੰ ਅਕਸਰ ਸਾਰੀਆਂ ਕਿਸਮਾਂ ਦੀਆਂ ਐਪਲੀਕੇਸ਼ਨਾਂ ਵਿੱਚ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਸਾਫਟਵੇਅਰ ਅਤੇ ਵੈੱਬ ਡਿਵੈਲਪਰ ਲਈ ਇਸ ਭਾਸ਼ਾ ਨੂੰ ਵੱਖ-ਵੱਖ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ, ਜਿਵੇਂ ਕਿ ਪਾਈਥਨ (Python), ਜਾਵਾ (Java) ਆਦਿ ਨਾਲ ਜੋੜਨ ਲਈ SQL ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਨ।

5.5.1 **SQL ਦੀਆਂ ਉਪਭਾਸ਼ਾਵਾਂ (Sub Languages of SQL):** SQL ਭਾਸ਼ਾ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਦੇ ਸੰਬੰਧਤ ਕਾਰਜ ਕਰਨ ਲਈ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੀਆਂ ਉਪ-ਭਾਸ਼ਾਵਾਂ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ ਜੋ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਹਨ:

5.5.1.1 **ਡਾਟਾ ਡੈਫੀਨੇਸ਼ਨ ਲੈਂਗੁਏਜ਼ (Data Definition Language (DDL)) :** DDL ਇੱਕ ਉਪਭਾਸ਼ਾ ਹੈ ਜੋ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰੇਜ ਗਰੁੱਪ, ਵੱਖ-ਵੱਖ ਸਟਰਕਚਰ ਅਤੇ ਹੋਰ ਆਬਜੈਕਟ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਕਮਾਂਡਜ਼ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। DDL ਸਟੇਟਮੈਂਟ ਮੁੱਖ ਤੌਰ ਤੇ ਡਾਟਾਬੇਸ ਆਬਜੈਕਟ ਜਿਵੇਂ ਕਿ ਟੇਬਲ, ਇੰਡੈਕਸ ਆਦਿ ਨੂੰ ਬਣਾਉਣ, ਬਦਲਣ ਅਤੇ ਡਲੀਟ ਕਰਨ ਨਾਲ ਸਬੰਧਤ ਹਨ। DDL ਵਿੱਚ ਡਾਟਾਬੇਸ ਆਬਜੈਕਟ ਜਿਵੇਂ ਟੇਬਲ (tables), ਸਿਕੁਐਂਸ (sequences), ਅਲਾਇਸ (aliases) ਅਤੇ ਇੰਡੈਕਸ (indexes) ਬਣਾਉਣ ਅਤੇ ਡਲੀਟ ਕਰਨ ਲਈ ਸਟ੍ਰਕਚਰਡ ਕਿਊਰੀ ਲੈਂਗੁਏਜ (SQL) ਸਟੇਟਮੈਂਟ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ। DDL ਕਮਾਂਡਜ਼ ਵਿਚ ਡਾਟਾ ਦਾ ਵਰਣਨ ਕਰਨ ਲਈ ਪਹਿਲਾਂ ਤੋਂ ਹੀ ਨਿਰਧਾਰਿਤ ਸਿੰਟੈਕਸ ਪਰਿਭਾਸ਼ਿਤ ਹੁੰਦਾ ਹੈ। ਡਾਟਾ ਡੈਫੀਨੇਸ਼ਨ ਲੈਂਗੁਏਜ (DDL) ਦੀਆਂ ਕਮਾਂਡਜ਼ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਮੌਜੂਦ ਕਰਨ ਦੇ ਤਰੀਕੇ ਨਾਲ ਸੰਬੰਧਤ ਹੁੰਦੀਆਂ ਹਨ। MySQL ਵਿੱਚ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਕੁਝ DDL ਕਮਾਂਡਾਂ CREATE, ALTER, DROP ਅਤੇ TRUNCATE ਹਨ। ਅਸੀਂ ਇਨ੍ਹਾਂ ਕਮਾਂਡਜ਼ ਨੂੰ ਇਸ ਪਾਠ ਦੇ ਪ੍ਰੈਕਟੀਕਲ ਭਾਗ ਵਿਚ ਪੜ੍ਹਾਂਗੇ।

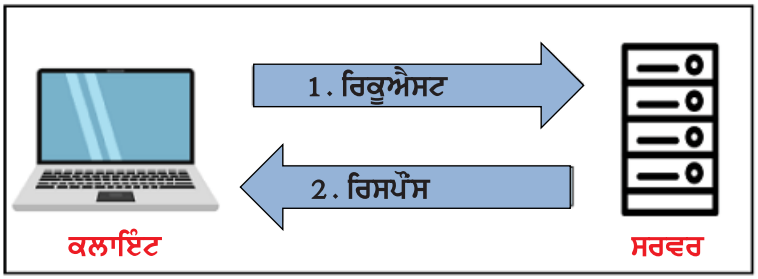
5.5.1.2 **ਡਾਟਾ ਮੈਨੀਪੁਲੇਸ਼ਨ ਲੈਂਗੁਜ਼ (DML (Data Manipulation Language):** ਡਾਟਾ ਮੈਨੀਪੁਲੇਸ਼ਨ ਲੈਂਗੁਏਜ਼ ਦੀ ਵਰਤੋਂ ਵੱਖ ਵੱਖ ਕਮਾਂਡਜ਼ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਨੂੰ ਤਬਦੀਲ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਉਪਭਾਸ਼ਾ ਵਿੱਚ, ਅਸੀਂ ਟੇਬਲ ਤੋਂ ਡਾਟਾ ਨੂੰ ਇਨਸਰਟ ਕਰਨਾ, ਅੱਪਡੇਟ ਜਾਂ ਡਲੀਟ ਕਰਨ ਲਈ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਾਂਗੇ। ਅਸੀਂ ਇਹਨਾਂ DML ਕਮਾਂਡਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਡਾਟਾ ਤੇ ਕਈ ਹੋਰ ਆਪ੍ਰੇਸ਼ਨ ਵੀ ਲਾਗੂ ਕਰ ਸਕਦੇ ਹਾਂ। MySQL ਵਿੱਚ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਕੁਝ DML ਕਮਾਂਡਜ਼ INSERT, DELETE, UPDATE, SELECT ਆਦਿ ਹਨ। ਅਸੀਂ ਇਨ੍ਹਾਂ ਕਮਾਂਡਜ਼ ਨੂੰ ਇਸ ਪਾਠ ਦੇ ਪ੍ਰੈਕਟੀਕਲ ਭਾਗ ਵਿਚ ਪੜ੍ਹਾਂਗੇ।

5.5.1.3 **ਡਾਟਾ ਕੰਟਰੋਲ ਲੈਂਗੁਏਜ਼ (DCL Data Control Language)):** ਡਾਟਾ ਕੰਟਰੋਲ ਲੈਂਗੁਏਜ਼ ਦੀ ਵਰਤੋਂ ਸਟੋਰ ਕੀਤੇ ਡਾਟਾ ਉੱਪਰ ਲੋੜੀਂਦੇ ਕਾਰਜ ਕਰਨ ਦੀ ਪ੍ਰਵਾਨਗੀ ਸੈੱਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਮੁੱਖ ਤੌਰ ਤੇ ਯੂਜ਼ਰ ਨੂੰ ਡਾਟਾਬੇਸ ਉੱਪਰ ਕੀਤੇ ਕੰਮਾਂ ਨੂੰ ਪੱਕੇ ਤੌਰ ਤੇ ਲਾਗੂ ਕਰਨ ਜਾਂ ਬਦਲਾਵ ਵਾਪਿਸ ਲੈਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਡਾਟਾਬੇਸ ਦੀ ਇਸ ਭਾਸ਼ਾ ਵਿੱਚ ਰੋਲਬੈਕ ਦੀ ਵਿਸ਼ੇਸ਼ਤਾ ਨਹੀਂ ਹੈ।

5.5.2 **MySQL ਨਾਲ ਜਾਣ ਪਛਾਣ (Introduction to MySQL):**

MySQL ਇੱਕ ਤੇਜ਼ ਅਤੇ ਵਰਤੋਂ ਵਿੱਚ ਆਸਾਨ RDBMS ਹੈ ਜੋ ਬਹੁਤ ਸਾਰੇ ਛੋਟੇ ਜਾਂ ਵੱਡੇ ਕਾਰੋਬਾਰਾਂ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। MySQL ਨੂੰ MySQL AB ਦੁਆਰਾ ਵਿਕਸਤ (developed), ਮਾਰਕੀਟਿੰਗ (marketed) ਅਤੇ ਸਮਰਥਿਤ (supported) ਕੀਤੀ ਗਈ ਜੋ ਕਿ ਇੱਕ ਸਵੀਡਿਸ਼ (Swedish) ਕੰਪਨੀ ਸੀ। MySQL ਆਪਣੇ ਆਪ ਵਿੱਚ ਇੱਕ ਬਹੁਤ ਸ਼ਕਤੀਸਾਲੀ ਪ੍ਰੋਗਰਾਮ ਹੈ। ਇਹ ਸਭ ਤੋਂ ਮਹਿੰਗੇ ਅਤੇ ਸ਼ਕਤੀਸ਼ਾਲੀ ਡਾਟਾਬੇਸ ਪੈਕੇਜਾਂ ਦੀ ਕਾਰਜਕੁਸ਼ਲਤਾ ਦੇ ਬਰਾਬਰ ਦੀਆਂ ਯੋਗਤਾਵਾਂ ਰੱਖਦੀ ਹੈ। ਇਹ ਜਾਣੀ-ਪਛਾਣੀ SQL ਕੁਏਰੀਜ਼ ਦੀ ਵਰਤੋਂ ਮੂਲ ਭਾਸ਼ਾ ਦੇ ਤੌਰ ਤੇ ਕਰਦੀ ਹੈ। MySQL ਇੱਕ ਅਜਿਹੀ ਭਾਸ਼ਾ ਹੈ ਜਿਸਦੀ ਵਰਤੋਂ ਲੋੜ ਅਨੁਸਾਰ ਬਦਲਾਵ ਕਰਕੇ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਇਸਦਾ ਓਪਨ-ਸੋਰਸ GPL ਲਾਇਸੈਂਸ ਪ੍ਰੋਗਰਾਮਰਾਂ ਨੂੰ ਉਹਨਾਂ ਦੇ ਆਪਣੇ ਖਾਸ ਵਾਤਾਵਰਨ ਵਿੱਚ ਆਪਣੀਆਂ ਲੋੜਾਂ ਪੂਰੀਆਂ ਕਰਨ ਲਈ MySQL ਸਾਫਟਵੇਅਰ ਵਿੱਚ ਤਬਦੀਲੀਆਂ ਕਰਕੇ ਪ੍ਰਯੋਗ ਕਰਨ ਦੀ ਇਜਾਜ਼ਤ ਦਿੰਦਾ ਹੈ।

MySQL ਕਲਾਇੰਟ-ਸਰਵਰ ਆਰਕੀਟੈਕਟਰ ਤੇ ਆਦਾਰਿਤ ਐਪਲੀਕੇਸ਼ਨਾਂ ਵਿੱਚ ਪ੍ਰਯੋਗ ਹੋਣ ਦੇ ਪੂਰੀ ਤਰ੍ਹਾਂ ਢੁੱਕਵੀਂ ਹੁੰਦੀ ਹੈ। ਇਹ ਮਾਡਲ ਐਂਡ ਯੂਜ਼ਰਜ਼ (end-users) ਲਈ ਡਿਜ਼ਾਇਨ ਕੀਤਾ ਗਿਆ ਹੈ ਜਿਨ੍ਹਾਂ ਨੂੰ ਕਲਾਇੰਟਸ (clients) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਜੋ ਨੈਟਵਰਕ ਸੇਵਾਵਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸਰਵਰ ਵਜੋਂ ਜਾਣੇ ਜਾਂਦੇ ਕੇਂਦਰੀ ਕੰਪਿਊਟਰ ਤੋਂ ਸਰੋਤਾਂ ਤੱਕ ਪਹੁੰਚ ਕੀਤੀ ਜਾ ਸਕੇ। ਇੱਥੇ ਕਲਾਇੰਟ ਇੱਕ ਗ੍ਰਾਫਿਕਲ ਯੂਜ਼ਰ ਇੰਟਰਫਸ (GUI) ਦੁਆਰਾ ਆਪਣੀਆਂ ਕੁਏਰੀਜ਼ ਭੇਜਦੇ ਹਨ ਅਤੇ ਸਰਵਰ ਭੇਜੀਆਂ ਹਦਾਇਤਾਂ ਦੀ ਪਾਲਣਾ ਕਰਦਾ ਹੋਇਆ ਲੋੜੀਂਦੀਆਂ ਆਉਟਪੁੱਟ ਦਿੰਦਾ ਹੈ। MySQL ਦੇ ਅਜਿਹੇ ਹੀ ਕੰਮ ਕਰਨ ਦੇ ਮਾਡਲ ਨੂੰ ਕਲਾਇੰਟ-ਸਰਵਰ ਮਾਡਲ ਦੇ ਰੂਪ ਵਿੱਚ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।



ਚਿੱਤਰ 5.8: **MySQL** ਕੁਐਰੀ ਪ੍ਰੋਸੈਸਿੰਗ ਮਾਡਲ

MySQL ਡਾਟਾਬੇਸ ਦਾ ਸਭ ਤੋਂ ਮਹੱਤਵਪੂਰਨ ਭਾਗ MySQL ਸਰਵਰ ਹੁੰਦਾ ਹੈ। ਇਹ ਸਰਵਰ ਇੱਕ ਵੱਖਰੇ ਪ੍ਰੋਗਰਾਮ ਦੇ ਰੂਪ ਵਿੱਚ ਉਪਲਬਦ ਹੁੰਦਾ ਹੈ ਅਤੇ ਡਾਟਾਬੇਸ ਦੀਆਂ ਸਾਰੀਆਂ ਹਦਾਇਤਾਂ, ਸਟੇਟਮੈਂਟਸ ਜਾਂ ਕਮਾਂਡਜ਼ ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦਾ ਹੈ। ਇਸ ਪਾਠ ਦੀਆਂ ਲੈਬ ਗਤੀਵਿਧੀਆਂ ਵਿੱਚ ਅਸੀਂ ਸਿੱਖਾਂਗੇ ਕਿ MySQL ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਡਾਟਾਬੇਸ ਉੱਤੇ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਕਾਰਜ ਕਰਨ ਲਈ SQL ਕਮਾਂਡਾਂ ਦੀ ਵਰਤੋਂ ਕਿਵੇਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਯਾਦ ਰੱਖਣ ਯੋਗ ਗੱਲਾਂ

- 'ਡਾਟਾ' ਨੂੰ ਵੱਖ-ਵੱਖ ਤੱਥਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਮੁੱਲ, ਮਾਪ ਅਤੇ ਸੰਖਿਆਵਾਂ ਸ਼ਾਮਲ ਹੋ ਸਕਦੀਆਂ ਹਨ।
- ਇੱਕ 'ਡਾਟਾਬੇਸ' ਨੂੰ ਸੰਬੰਧਿਤ ਡਾਟਾ ਦੇ ਇੱਕ ਸਮੂਹ ਦੇ ਰੂਪ ਵਿੱਚ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜੋ ਅਤੇ ਬਿਨਾਂ ਰਿਡੰਡੈਂਸੀ ਦੇ ਇਕੱਠੇ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।
- 'ਸੂਚਨਾ' ਨੂੰ ਸੰਗਠਿਤ ਜਾਂ ਵਰਗੀਕ੍ਰਿਤ ਡਾਟਾ ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜੋ ਕਿ ਐਂਡ-ਯੂਜ਼ਰ ਲਈ ਕੋਈ ਵੀ ਅਹਿਮ ਫੈਸਲਾ ਲੈਣ ਲਈ ਉਪਯੋਗੀ ਹੁੰਦਾ ਹੈ।
- ਫਾਈਲ-ਆਧਾਰਿਤ ਸਿਸਟਮ ਇਕੱਲੇ ਇਕੱਲੇ ਰਿਕਾਰਡ ਦੇ ਰੂਪ ਵਿੱਚ ਤੱਥਾਂ ਨੂੰ ਸਟੋਰ ਕਰਨ ਦਾ ਪੁਰਾਤਨ ਸਿਸਟਮ ਹੁੰਦਾ ਹੈ।
- ਇੱਕ ਡਾਟਾਬੇਸ ਡਾਟਾ ਦਾ ਇੱਕ ਸਿਸਟੇਮੈਟਿਕ ਸੰਗਠਿਤ ਸੰਗ੍ਰਹਿ ਹੈ ਜੋ ਇੱਕ ਅਰਥਪੂਰਨ ਤਰੀਕੇ ਨਾਲ ਆਪਸ ਵਿੱਚ ਸੰਬੰਧਿਤ ਹੁੰਦਾ ਹੈ। ਇਸਨੂੰ ਵੱਖ-ਵੱਖ ਯੂਜ਼ਰਜ਼ ਦੁਆਰਾ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਬਿਨਾਂ ਕਿਸੇ ਰਿਡੰਡੈਂਸੀ ਦੇ ਐਕਸੈਸ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।
- ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਬਹੁਤ ਸਾਰੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਹੁੰਦੀਆਂ ਹਨ ਜਿਵੇਂ ਕਿ ਘੱਟੋ-ਘੱਟ ਰਿਡੰਡੈਂਸੀ, ਨਾਬਰਾਬਰਤਾ (Incosistency) ਤੋਂ ਬਚਣਾ, ਡਾਟਾ ਨੂੰ ਸਾਂਝਾ ਕਰਨਾ, ਸਰਚ ਕਰਨ ਦੀ ਸਮਰੱਥਾ, ਇਕਸਾਰਤਾ, ਸੇਫਟੀ ਅਤੇ ਸੁਰੱਖਿਆ, ਆਸਾਨੀ ਨਾਲ ਪਹੁੰਚਣਯੋਗਤਾ, ਆਸਾਨ ਬੈਕਅੱਪ/ਰਿਕਵਰੀ ਆਦਿ।
- ਡਾਟਾਬੇਸ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ (DBMS) ਲੋੜੀਂਦੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਕਰਨਾ, ਡਾਟਾ ਪ੍ਰਾਪਤ ਕਰਨਾ ਅਤੇ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਸਾਫਟਵੇਅਰ ਸਿਸਟਮ ਹੁੰਦੇ ਹਨ।
- DBA ਡਾਟਾਬੇਸ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਲਈ ਜ਼ਿੰਮੇਵਾਰ ਵਿਅਕਤੀ ਜਾਂ ਵਿਅਕਤੀਆਂ ਦਾ ਸਮੂਹ ਹੁੰਦਾ ਹੈ।
- ਹਾਰਡਵੇਅਰ, ਸਾਫਟਵੇਅਰ, ਯੂਜ਼ਰ, ਕਿਰਿਆਵਾਂ ਅਤੇ ਡਾਟਾ ਡਾਟਾਬੇਸ ਦੇ ਕੁਝ ਮੁੱਖ ਭਾਗ ਹੁੰਦੇ ਹਨ।
- ਡਾਟਾ ਮਾਡਲ ਨੂੰ ਇੱਕ ਡਾਟਾਬੇਸ ਦੀ ਲਾਜ਼ੀਕਲ ਬਣਤਰ ਵਜੋਂ ਜਾਣਿਆ ਜਾ ਸਕਦਾ ਹੈ।
- ਸਭ ਤੋਂ ਆਮ ਤੌਰ ਤੇ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਰਿਕਾਰਡ-ਅਧਾਰਿਤ ਡਾਟਾ ਮਾਡਲ ਹਨ ਹੈਰਾਰੀਕਲ ਮਾਡਲ (ਟ੍ਰੀ ਸਟ੍ਰਕਟਰ), ਨੈੱਟਵਰਕ ਮਾਡਲ (ਪਲੈਕਸ ਸਟ੍ਰਕਚਰ) ਅਤੇ ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ (ਸਧਾਰਨ ਢਾਂਚੇ)।
- RDBMS ਦਾ ਅਰਥ ਹੈ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ।
- ਐਟਰੀਬਿਊਟਜ਼ ਕਿਸੇ ਵੀ ਆਇਟਮ ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਦਾ ਵਰਨਣ ਕਰਨ ਵਾਲੇ ਗੁਣ ਹੁੰਦੇ ਹਨ।

- DBMS ਵਿੱਚ ਇੱਕ ਟਪਲ ਕਿਸੇ ਵੀ ਐਂਟੀਟੀ ਦੇ ਸੰਬੰਧਤ ਐਟਰੀਬਿਊਟਜ਼ ਦੇ ਮੁੱਲਾਂ ਦਾ ਇੱਕ ਸੰਗ੍ਰਹਿ ਹੁੰਦਾ ਹੈ ਜੋ ਕਿ ਸਾਰੇ ਆਪਸ ਵਿੱਚ ਵਿਲੱਖਣ ਹੁੰਦੇ ਹਨ।
- ਇੱਕ ਰਿਲੇਸ਼ਨ ਇੱਕ ਟੇਬਲ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਰੋਅਜ਼ ਅਤੇ ਕਾਲਮਾਂ ਦੇ ਰੂਪ ਵਿੱਚ ਕੁਝ ਮੁੱਲ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਜਿੱਥੇ ਇੱਕ ਰੋਅ ਜਾਂ ਸੰਬੰਧਿਤ ਡਾਟਾ ਮੁੱਲਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਨੂੰ ਟਪਲ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
- ਰਿਲੇਸ਼ਨ (ਟੇਬਲ) ਦੇ ਕਿਸੇ ਵੀ ਐਟਰੀਬਿਊਟ ਲਈ ਸਾਰੇ ਸਵੀਕਾਰ ਕਾਤੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਨੂੰ ਡੋਮੇਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
- ਕਿਸੇ ਰਿਲੇਸ਼ਨ/ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਐਟਰੀਬਿਊਟਜ਼/ਕਾਲਮਾਂ ਦੀ ਕੁੱਲ ਸੰਖਿਆ ਨੂੰ ਰਿਲੇਸ਼ਨ ਦੀ ਡਿਗਰੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
- ਕਿਸੇ ਰਿਲੇਸ਼ਨ/ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਟਪਲਾਂ/ਰਿਕਾਰਡਾਂ ਦੀ ਕੁੱਲ ਸੰਖਿਆ ਨੂੰ ਰਿਲੇਸ਼ਨ ਦੀ ਕਾਰਡੀਨੈਲਿਟੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
- DBMS ਵਿੱਚ ਕੀਅਜ਼ ਕਿਸੇ ਵੀ ਐਟਰੀਬਿਊਟ (ਜਾਂ ਕਾਲਮਾਂ) ਜਾਂ ਇਕ ਤੋਂ ਵੱਧ ਐਟਰੀਬਿਊਟਜ਼ ਨਾਲ ਸੰਬੰਧਤ ਹੋ ਸਕਦੀਆਂ ਹਨ।
- ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਇੱਕ ਕਾਲਮ ਜਾਂ ਇੱਕ ਟੇਬਲ ਦੇ ਕਾਲਮਾਂ ਦੇ ਸਮੂਹ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ ਜੋ ਸਾਰੇ ਰਿਕਾਰਡਾਂ ਨੂੰ ਵਿਲੱਖਣ ਰੱਖਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੀ ਹੈ।
- ਹਰੇਕ ਫੋਰਨ ਕੀਅ ਨੂੰ ਮੌਜੂਦਾ ਟੇਬਲ ਦੇ ਮੁੱਲ ਨੂੰ ਮੁੱਢਲੇ ਟੇਬਲ ਦੀ ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਨਾਲ ਮੇਲ ਕਰਨ ਲਈ ਕਾਲਮ/ਕਾਲਮਾਂ ਦੇ ਸਮੂਹ ਵਿੱਚ ਹਰੇਕ ਮੁੱਲ ਦੇ ਮੌਜੂਦ ਹੋਣ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ।
- ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਨੂੰ ਐਕਸੈਸ ਕਰਨ ਅਤੇ ਬਦਲਣ ਲਈ SQL ਇੱਕ ਆਮ ਭਾਸ਼ਾ ਹੈ।
- SQL ਰਿਲੇਸ਼ਨਲ ਡਾਟਾਬੇਸ ਤੇ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਆਪ੍ਰੇਸ਼ਨ ਕਰਨ ਲਈ DDL, DML, DCL ਵਰਗੀਆਂ ਉਪ-ਭਾਸ਼ਾਵਾਂ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ।
- MySQL ਸਭ ਤੋਂ ਮਹਿੰਗੇ ਅਤੇ ਸ਼ਕਤੀਸਾਲੀ ਡਾਟਾਬੇਸ ਪੈਕੇਜਾਂ ਦੇ ਇੱਕ ਵੱਡੇ ਸਮੂਹ ਨੂੰ ਸੰਭਾਲਣ ਲਈ ਇੱਕ ਬਹੁਤ ਸ਼ਕਤੀਸਾਲੀ ਪ੍ਰੋਗਰਾਮ ਹੈ।

ਅਭਿਆਸ

ੳ. ਬਹੁਪਸੰਦੀ ਪ੍ਰਸ਼ਨ

1. ਕਿਹੜਾ ਨੈੱਟਵਰਕ ਮਾਡਲ ਪਲੈਕਸ ਸਟ੍ਰਕਟਰ ਦੇ ਰੂਪ ਵਿੱਚ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਹੈ ?
   ੳ. ਹਾਇਰੈਰੀਕਲ ਮਾਡਲ (Hierarchical Model) ਅ. ਨੈੱਟਵਰਕ ਮਾਡਲ (Network Model)
   ੲ. ਰਿਲੇਸ਼ਨਲ ਮਾਡਲ (Relational Model) ਸ. ਇਹਨਾਂ ਵਿਚੋਂ ਕੋਈ ਨਹੀਂ
2. ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਟਰਮਜ਼ ਵਿੱਚੋਂ ਕਿਹੜਾ ਇੱਕ ਟੇਬਲ ਵਿੱਚਲੇ ਕਾਲਮ ਦੇ ਨਾਮ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ ?
   ੳ. ਰਿਲੇਸ਼ਨ (Relation) ਅ. ਐਟਰੀਬਿਊਟ (Attribute)
   ੲ. ਡਿਗਰੀ (Degree) ਸ. ਟਪਲ (Tuple)
3. ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਟਰਮਜ਼ ਵਿੱਚੋਂ ਕਿਹੜਾ ਡਾਟਾਬੇਸ ਦਾ ਹਿੱਸਾ (Component) ਨਹੀਂ ਹੈ ?
   ੳ. ਹਾਰਡਵੇਅਰ (Hardware) ਅ. ਸਾਫਟਵੇਅਰ (Software)
   ੲ. ਨੈਟਵਰਕ (Network) ਸ. ਯੂਜ਼ਰ (User)

4. ਡਾਟਾਬੇਸ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਲਈ ਕੌਣ ਜ਼ਿੰਮੇਵਾਰ ਹੈ ?

ੳ. ਐਂਡ ਯੂਜ਼ਰ (End User)　　ਅ. ਸਿਸਟਮ ਪ੍ਰੋਗਰਾਮਰ

ੲ. ਡਾਟਾਬੇਸ ਐਡਮੀਨਿਸਟ੍ਰੇਟਰ　　ਸ. ਡਾਟਾ ਮਾਡਲਰ

5. ਇੱਕ ਸੰਗਠਿਤ ਜਾਂ ਵਰਗੀਕ੍ਰਿਤ ਡਾਟਾ ਨੂੰ ਕੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ:

ੳ. ਸੂਚਨਾ (Information)　　ਅ. ਟੇਬਲ (Table)

ੲ. ਰਿਜਲਟ (Result)　　ਸ. ਇਹਨਾਂ ਵਿੱਚੋਂ ਕੋਈ ਨਹੀਂ

ਅ. ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ:

1. ..... ਸਿਸਟਮ ਤੱਥਾਂ ਨੂੰ ਮੈਨੂੰਅਲ ਤਰੀਕੇ ਨਾਲ ਸਟੋਰ ਕਰਨ ਲਈ ਪੁਰਾਤਨ ਤਰੀਕੇ ਸਨ।
2. ..... ਨੂੰ ਡੇਟਾਬੇਸ ਦੀ ਇੱਕ ਲਾਜ਼ੀਕਲ ਬਣਤਰ ਵਜੋਂ ਜਾਣਿਆ ਜਾ ਸਕਦਾ ਹੈ।
3. ਇੱਕ ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਰਿਕਾਰਡਾਂ ਦੀ ਸੰਖਿਆ ਨੂੰ ਉਸ ਟੇਬਲ ਦੀ ..... ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।
4. ..... ਕੀਅ ਕਿਸੇ ਖਾਸ ਕਾਲਮ ਵਿੱਚ ਸਾਰੇ ਰਿਕਾਰਡਾਂ ਨੂੰ ਵਿਲੱਖਣ ਰੂਪ ਵਿੱਚ ਰੱਖਣ ਦੀ ਇਜ਼ਾਜਤ ਦਿੰਦੀ ਹੈ।
5. ..... ਇੱਕ ਆਮ ਭਾਸ਼ਾ ਹੈ ਜੋ ਇੱਕ ਡੇਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰੇਜ਼ ਗਰੁੱਪਜ਼, ਵੱਖ-ਵੱਖ ਢਾਂਚੇ ਅਤੇ ਆਬਜੈਕਟਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਕਮਾਂਡਜ਼ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।

ੲ. ਹੇਠਾਂ ਲਿਖੇ ਸ਼ਬਦਾਂ ਦੇ ਪੂਰੇ ਨਾਂ ਲਿਖੋ:

1. RDBMS
2. DML
3. DDL
4. DBMS
5. DBA
6. SQL

ਸ. ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ

1. ਡਾਟਾ ਅਤੇ ਸੂਚਨਾ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋ।
2. DBMS ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਭਾਵ ਹੈ ?
3. ਡਾਟਾਬੇਸ ਦੇ ਹਾਰਡਵੇਅਰ ਕੰਪੋਨੈਂਟਜ਼ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।
4. ਡਾਟਾ ਮਾਡਲਾਂ ਤੇ ਇੱਕ ਛੋਟਾ ਨੋਟ ਲਿਖੋ।
5. ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਰਿਲੇਸ਼ਨਜ਼ ਕੀ ਹੁੰਦੇ ਹਨ ?
6. ਰਿਲੇਸ਼ਨ ਦੀ ਡਿਗਰੀ ਅਤੇ ਕਾਰਡੀਨੈਲਿਟੀ ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਭਾਵ ਹੈ ?
7. ਇੱਕ ਡਾਟਾਬੇਸ ਦੀ ਪ੍ਰਾਇਮਰੀ ਕੀਅ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋ।

ਹ. ਵੱਡੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ

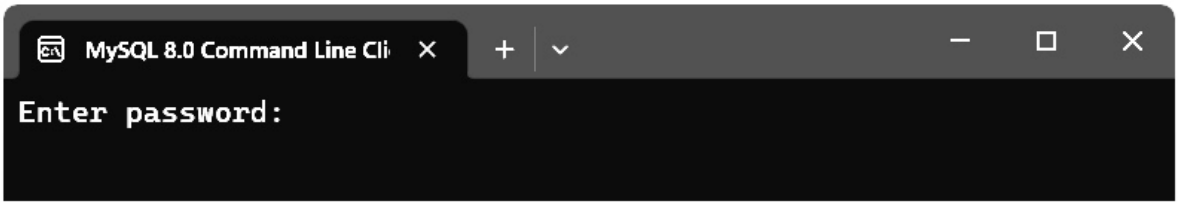
1. ਇੱਕ ਡਾਟਾਬੇਸ ਦੇ ਕਿਹੜੇ ਕਿਹੜੇ ਭਾਗ ਹੁੰਦੇ ਹਨ ? ਵਿਸਤਾਰ ਵਿੱਚ ਸਮਝਾਓ।
2. ਇੱਕ ਡਾਟਾਬੇਸ ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋ।
3. ਇੱਕ ਡਾਟਾਬੇਸ ਦਾ ਰਿਲੇਸ਼ਨਲ ਡਾਟਾ ਮਾਡਲ ਕੀ ਹੈ ? ਵਿਸਥਾਰ ਵਿੱਚ ਸਮਝਾਓ.
4. SQL ਕੀ ਹੈ ? SQL ਦੀਆਂ ਉਪ-ਭਾਸ਼ਾਵਾਂ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।

MySQL ਲੈਬ ਐਕਟੀਵਿਟੀ

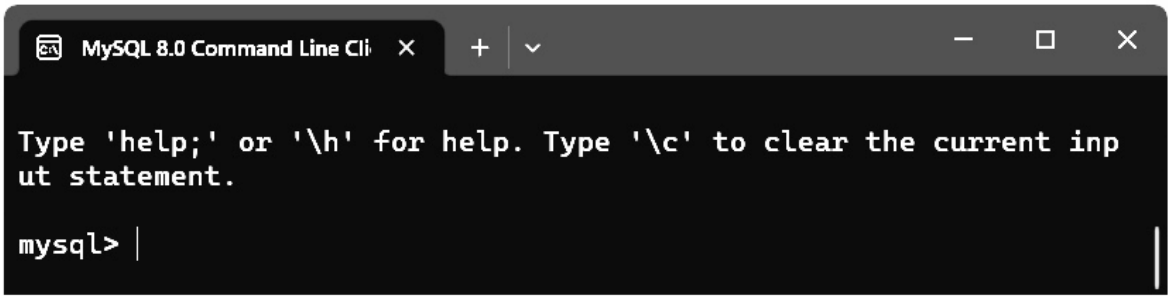
ਲੈਬ 5.1 MySQL ਨੂੰ ਸਟਾਰਟ ਕਰਨਾ (Starting MySQL):

ਅਸੀਂ ਵਿੰਡੋਜ਼ ਦੇ ਸਟਾਰਟ ਮੀਨੂੰ ਤੋਂ GUI ਮੋਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ MySQL ਨੂੰ ਸਟਾਰਟ ਕਰ ਸਕਦੇ ਹਾਂ। ਜੇਕਰ MySQL ਕਿਸੇ ਵੀ ਕੰਪਿਊਟਰ ਤੇ ਸਫਲਤਾਪੂਰਵਕ ਸਥਾਪਿਤ ਹੋ ਗਈ ਹੈ, ਤਾਂ ਅਸੀਂ MySQL ਦੇ ਕੰਸੋਲ ਨੂੰ ਚਲਾਉਣ ਲਈ ਹੇਠਾਂ ਦੱਸੇ ਗਏ ਕਦਮਾਂ ਦੀ ਪਾਲਣਾ ਕਰ ਸਕਦੇ ਹਾਂ, ਜੋ ਕਿ MySQL ਸਰਵਰ ਅਤੇ ਇੱਕ ਕਲਾਇੰਟ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਹੈ। ਜੇਕਰ MySQL ਸਥਾਪਤ ਨਹੀਂ ਹੈ ਤਾਂ ਪੁਸਤਕ ਦੇ Appendix-I ਦੀ ਪਾਲਨਾ ਕੀਤੀ ਦਾ ਸਕਦੀ ਹੈ।

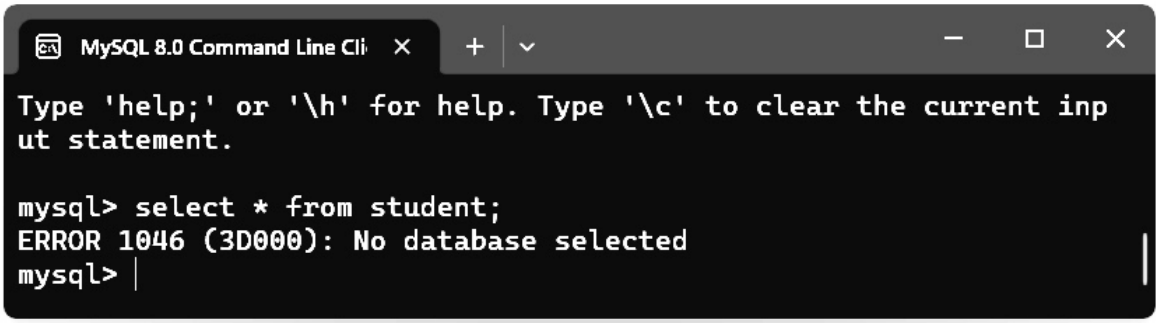
Start -> "MySQL Command Line Client" ਨੂੰ ਸਰਚ ਕਰੋ -> ਪ੍ਰੋਗਰਾਮ ਆਈਕਨ 'ਤੇ ਕਲਿੱਕ ਕਰੋ। ਇਸ ਤਰੀਕੇ ਨਾਲ ਸਾਡੇ ਕੋਲ ਇੱਕ ਸਕ੍ਰੀਨ ਨਜ਼ਰ ਆਵੇਗੀ ਜੋ ਕਿ ਹੇਠਾਂ ਦਿਖਾਈ ਗਈ ਹੈ:



ਇੱਥੇ ਅਸੀਂ MySQL ਦੀ ਇੰਸ਼ਟਾਲੇਸਨ ਦੌਰਾਨ ਪਰਿਭਾਸ਼ਿਤ ਪਾਸਵਰਡ ਦਰਜ ਕਰਾਂਗੇ। ਜੇਕਰ ਅਸੀਂ ਇਸਨੂੰ ਸਕੂਲ ਦੀ ਕੰਪਿਊਟਰ ਲੈਬ ਵਿੱਚ ਵਰਤ ਰਹੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਆਪਣੇ ਅਧਿਆਪਕ ਤੋਂ ਇਸ ਬਾਰੇ ਪੁੱਛ ਸਕਦੇ ਹਾਂ। ਸਫ਼ਲ ਲੌਗਇਨ ਤੋਂ ਬਾਅਦ ਹੇਠਾਂ ਦਿੱਤੀ ਸਕ੍ਰੀਨ ਦਿਖਾਈ ਦੇਵੇਗੀ:



ਹੁਣ ਅਸੀਂ ਆਪਣੇ ਡਾਟਾਬੇਸ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਅਤੇ ਆਪਣੇ MySQL ਕਲਾਇੰਟ ਦੀ ਵਰਤੋਂ ਕਰਨ ਲਈ ਤਿਆਰ ਹਾਂ। ਲੌਗਇਨ ਕਰਨ ਤੋਂ ਬਾਅਦ ਪਹਿਲਾ ਕਦਮ ਡਾਟਾਬੇਸ ਦੀ ਚੋਣ ਕਰਨਾ ਹੁੰਦਾ ਹੈ। ਇੱਕ ਡਾਟਾਬੇਸ ਇਕ ਸਟਰਕਚਰਡ ਡਾਟਾ ਦਾ ਸਮੂਹ ਹੁੰਦਾ ਹੈ। ਇਹ ਇੱਕ ਅਜਿਹੀ ਸਟੋਰੇਜ਼ ਹੈ ਜਿੱਥੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਅਤੇ ਸੰਗਠਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਜੇਕਰ ਅਸੀਂ MySQL ਵਿੱਚ ਡਾਟਾਬੇਸ ਦੀ ਚੋਣ ਨਹੀਂ ਕਰਦੇ ਹਾਂ ਤਾਂ ਹੇਠਾਂ ਦਰਸਾਇਆ ਐਰਰ ਮੈਸੇਜ਼ (error message) ਦਿਖਾਈ ਦੇਵੇਗੀ।



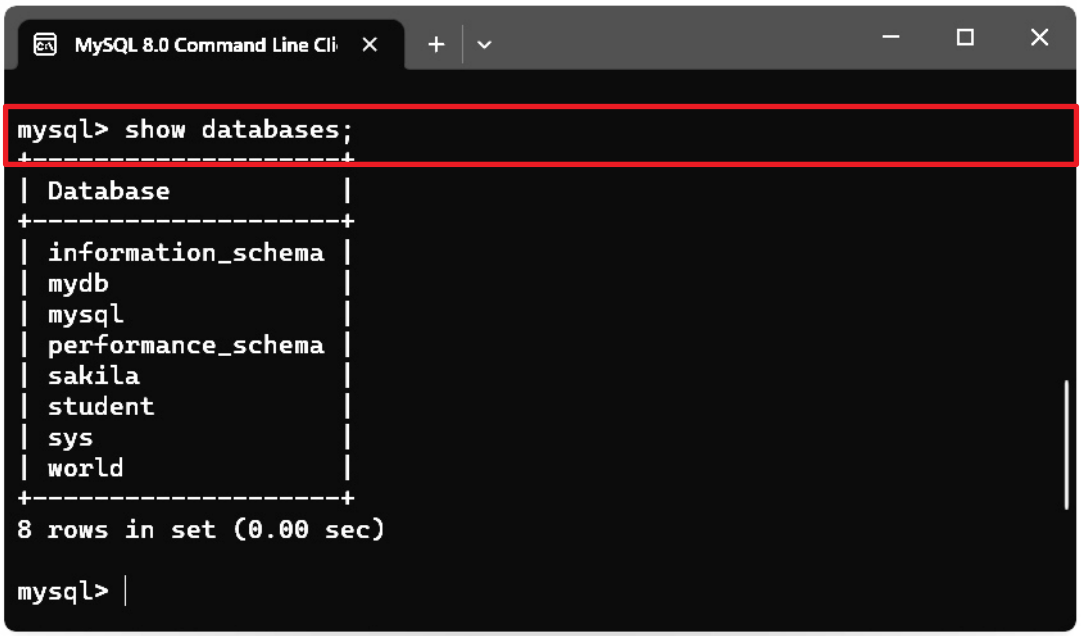
ਇੱਥੇ ਡਾਟਾਬੇਸ ਦੀ ਚੋਣ ਕਰਨ ਦੀ ਪ੍ਰਕਿਰਿਆ ਲਈ ਪਹਿਲਾਂ ਡਾਟਾਬੇਸ ਦੇ ਨਾਮ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਮੌਜੂਦਾ MySQL ਸਰਵਰ ਵਿੱਚ ਬਣਾਏ ਗਏ ਸਾਰੇ ਡਾਟਾਬੇਸਾਂ ਦੀ ਸੂਚੀ ਦੇਖਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਹੇਠ ਦਿੱਤੀ ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

ਲੈਬ 5.1.1 ਸਾਰੇ ਮੌਜੂਦਾ ਡਾਟਾਬੇਸ ਦੀ ਸੂਚੀ ਦੇਖਣਾ:

ਮੌਜੂਦਾ ਯੂਜ਼ਰ ਦੁਆਰਾ ਬਣਾਏ/ਪ੍ਰਬੰਧਿਤ ਕੀਤੇ (created/managed) ਗਏ ਸਾਰੇ ਡਾਟਾਬੇਸ ਦੀ ਸੂਚੀ ਦੇਖਣ ਲਈ ਅਸੀਂ "SHOW" ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਸਦੇ ਲਈ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

```
SHOW DATABASES;
```

ਅਸੀਂ MySQL ਕਮਾਂਡ ਲਾਈਨ ਕਲਾਇੰਟ ਵਿੰਡੋਜ਼ ਵਿੱਚ ਇਸ ਕਮਾਂਡ ਦੀ ਇੱਕ ਉਦਾਹਰਣ ਇਸ ਪ੍ਰਕਾਰ ਦੇਖ ਸਕਦੇ ਹਾਂ:



ਦਿੱਤੀ ਗਈ ਕੁਐਰੀ ਦੀ ਆਉਟਪੁੱਟ ਦੇ ਰੂਪ ਵਿੱਚ ਮੌਜੂਦਾ ਯੂਜ਼ਰ ਦੁਆਰਾ ਬਣਾਏ ਗਏ ਵੱਖ-ਵੱਖ ਡਾਟਾਬੇਸ ਦਿਖਾਈ ਦੇ ਰਹੇ ਹਨ। ਅਸੀਂ ਉਹਨਾਂ ਵਿੱਚੋਂ ਕਿਸੇ ਨੂੰ ਵੀ ਆਪਣੇ ਮਕਸਦ ਲਈ ਵਰਤ ਸਕਦੇ ਹਾਂ ਜਾਂ ਕਿਸੇ ਨਵੇਂ ਪ੍ਰੋਜੈਕਟ ਲਈ ਨਵਾਂ ਡਾਟਾਬੇਸ ਵੀ ਬਣਾ ਸਕਦੇ ਹਾਂ।

ਲੈਬ 5.1.2 ਇੱਕ ਪਹਿਲਾਂ ਬਣੇ ਡਾਟਾਬੇਸ ਦੀ ਚੋਣ:

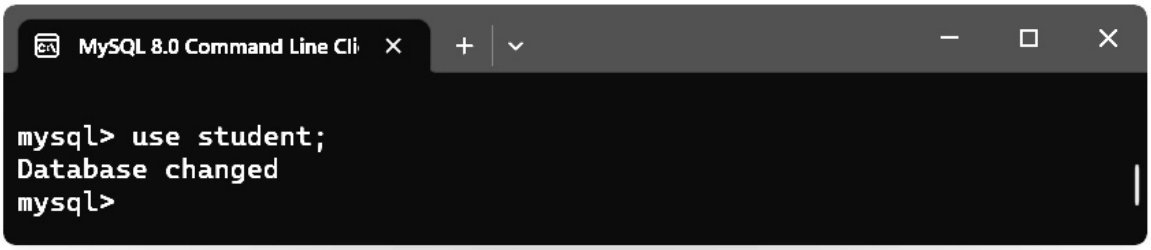
ਅਸੀਂ ਆਪਣੇ ਲੋੜੀਂਦੇ ਕੰਮ ਨਾਲ ਸਬੰਧਤ ਸਾਡੇ DBMS ਕਾਰਜਾਂ ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਇੱਕ ਡਾਟਾਬੇਸ ਦੀ ਚੋਣ ਕਰ ਸਕਦੇ ਹਾਂ। "USE" ਸਟੇਟਮੈਂਟ ਨੂੰ ਇਸ ਮਕਸਦ ਲਈ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਲਈ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

ਸਿੰਟੈਕਸ:

```
USE Database_Name;
```

ਉਦਾਹਰਣ:

```
mysql> USE student;
```



ਇਸ ਪ੍ਰਕਾਰ ਕਮਾਂਡ ਦੇ ਲਾਗੂ ਹੋਣ ਤੋਂ ਬਾਅਦ ਇੱਕ ਸੁਨੇਹਾ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ ਕਿ ਦਿੱਤਾ ਗਿਆ ਡਾਟਾਬੇਸ ਚੁਣਿਆ ਗਿਆ ਹੈ। ਜੇਕਰ ਡਾਟਾਬੇਸ ਮੌਜੂਦ ਨਹੀਂ ਹੈ ਜਾਂ ਦਿੱਤਾ ਗਿਆ ਨਾਮ ਮੇਲ ਨਹੀਂ ਖਾਂਦਾ ਤਾਂ "ERROR 1049 (42000): Unknown database 'DatabseName' (ਅਣਜਾਣ ਡਾਟਾਬੇਸ 'ਡਾਟਾਬੇਸ ਨਾਮ') ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਜਾਵੇਗਾ। ਇਸ ਸਥੀਤੀ ਵਿੱਚ ਕਮਾਂਡ ਵਿੱਚ ਲੋੜੀਂਦੀਆਂ ਤਬਦੀਲੀਆਂ ਕਰਨ ਤੋਂ ਬਾਅਦ ਦੋਬਾਰਾ ਇਸੇ ਕਮਾਂਡ ਨੂੰ ਲਾਗੂ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

ਲੈਬ 5.2 ਡਾਟਾ ਡੈਫੀਨੇਸ਼ਨ ਲੈਂਗੂਏਜ (DDL) ਕਮਾਂਡਜ਼:

DDL ਇੱਕ ਆਮ ਭਾਸ਼ਾ ਹੈ ਜੋ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰੇਜ ਗਰੁੱਪਜ਼, ਵੱਖ-ਵੱਖ ਢਾਂਚੇ ਅਤੇ ਆਬਜੈਕਟਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਕਮਾਂਡਜ਼ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। DDL ਸਟੇਟਮੈਂਟਾਂ ਮੁੱਖ ਤੌਰ 'ਤੇ ਡਾਟਾਬੇਸ ਵਸਤੂਆਂ ਜਿਵੇਂ ਕਿ ਟੇਬਲ, ਇੰਡੈਕਸ ਆਦਿ ਨੂੰ ਬਣਾਉਣ, ਤਬਦੀਲ ਕਰਨ ਅਤੇ ਡਲੀਟ ਕਰਨ ਨਾਲ ਸਬੰਧਤ ਹਨ। DDL ਵਿੱਚ ਡਾਟਾਬੇਸ (databases), ਟੇਬਲ (tables), ਸਿਕੂਐਂਸ (sequences), ਅਲਾਇਸ (aliases) ਅਤੇ ਇੰਡੈਕਸ (index) ਬਣਾਉਣ ਅਤੇ ਡਰੋਪ ਕਰਨ ਲਈ ਸਟ੍ਰਕਚਰਡ ਕਿਊਰੀ ਲੈਂਗੂਏਜ (SQL) ਸਟੇਟਮੈਂਟ ਸ਼ਾਮਲ ਹਨ। DDL ਕਮਾਂਡਜ਼ ਦੇ ਡਾਟਾ ਨਾਲ ਪ੍ਰਬੰਧਨ ਕਰਨ ਲਈ ਸਿੰਟੈਕਸ ਪਹਿਲਾਂ ਤੋਂ ਪਰਿਭਾਸ਼ਿਤ ਹੈ। ਡਾਟਾ ਡੈਫੀਨੇਸ਼ਨ ਲੈਂਗੂਏਜ ਦੀਆਂ ਕਮਾਂਡਜ਼ ਇਹ ਨਿਰਧਾਰਿਤ ਕਰਦੀਆਂ ਹਨ ਕਿ ਡਾਟਾਬੇਸ ਵਿੱਚ ਡਾਟਾ ਕਿਵੇਂ ਮੌਜੂਦ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। MySQL ਵਿੱਚ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਕੁਝ DDL ਕਮਾਂਡਾਂ CREATE, ALTER, DROP ਅਤੇ TRUNCATE ਹਨ।

ਲੈਬ 5.2.1 ਨਵਾਂ ਡਾਟਾਬੇਸ ਬਣਾਉਣਾ:

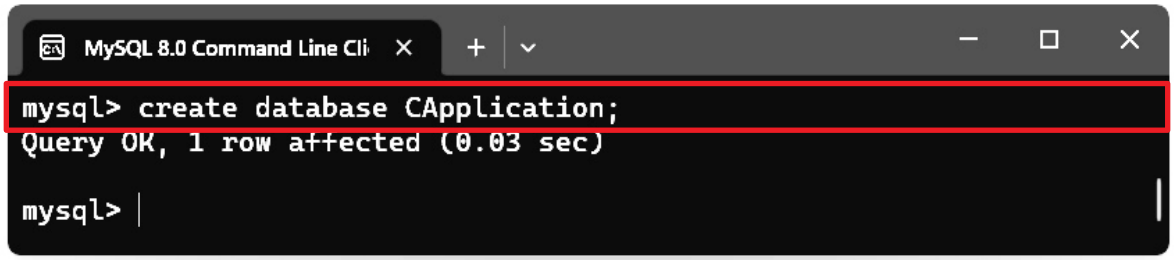
ਜੇਕਰ ਡਾਟਾਬੇਸ ਐਡਮੀਨੀਸ਼ਟ੍ਰੇਟਰ (administrator) ਦੁਆਰਾ ਸਾਡੇ ਲਈ ਡਾਟਾਬੇਸ ਬਣਾਏ ਗਏ ਹਨ ਤਾਂ ਅਸੀਂ ਡਾਟਾਬੇਸ ਦੀ ਚੋਣ ਕਰਕੇ ਇਸ ਉਪਰ ਆਪਣੇ ਲੋੜੀਂਦੇ ਕੰਮ ਕਰ ਸਕਦੇ ਹਾਂ। ਨਹੀਂ ਤਾਂ ਸਾਨੂੰ ਆਪਣੇ ਉਦੇਸ਼ਾਂ ਦੀ ਪੂਰਤੀ ਲਈ ਇੱਕ ਨਵਾਂ ਡਾਟਾਬੇਸ ਬਣਾਉਣ ਦੀ ਜ਼ਰੂਰਤ ਹੁੰਦੀ ਹੈ। ਇਸ ਲਈ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

ਸਿੰਟੈਕਸ:

```
CREATE DATABSE Databse_Name;
```

ਉਦਾਹਰਣ

```
mysql> CREATE DATABSE Capplication;
```



"Query OK" ਦੇ ਰੂਪ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਸੁਨੇਹਾ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਕਮਾਂਡ ਰਾਹੀਂ ਬਣਾਇਆ ਡਾਟਾਬੇਸ ਸਫ਼ਲਤਾਪੂਰਵਕ ਬਣ ਗਿਆ ਹੈ। ਅਸੀਂ ਇਸ ਵਿੱਚ ਸਬੰਧਾਂ ਅਤੇ ਡਾਟਾ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਲਈ "Use" ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸਨੂੰ ਚੁਣ ਸਕਦੇ ਹਾਂ।

Lab 5.2.2 MySQL ਵਿੱਚ ਡਾਟਾ ਟਾਈਪਸ:

ਟੇਬਲ ਵਿੱਚ ਇੱਕ ਫੀਲਡ ਵਿਚ ਸਟੋਰ ਹੋਣ ਵਾਲੇ ਡਾਟਾ ਮੁੱਲਾਂ ਦੀ ਡਾਟਾ ਕਿਸਮ ਨੂੰ ਸਹੀ ਢੰਗ ਨਾਲ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨਾ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਹੁੰਦਾ ਹੈ। ਅਸੀਂ ਪਹਿਲਾਂ ਤੋਂ ਪ੍ਰਭਾਸ਼ਿਤ ਕੀਅ ਵਰਡਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਆਸਾਨੀ ਨਾਲ ਡਾਟਾ ਟਾਇਪ ਨਿਰਧਾਰਿਤ ਕਰ ਸਕਦੇ ਹਾਂ। ਡਾਟਾ ਦੀ ਕਿਸਮ ਨੂੰ ਉਸ ਵਿਸ਼ੇਸ਼ ਫਿਲਡ ਵਿੱਚ ਦਾਖਲ ਕੀਤੇ ਜਾਣ ਦੀ ਇਜਾਜ਼ਤ ਵਾਲੇ ਮੁੱਲਾਂ ਦੇ ਸਮੂਹ ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਗਿਆ ਹੈ। MySQL ਕਈ ਵੱਖ-ਵੱਖ ਡਾਟਾ ਕਿਸਮਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਮੁੱਖ ਡਾਟਾ ਕਿਸਮਾਂ ਦਾ ਵਰਣਨ ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਹੈ, ਜਿਨ੍ਹਾਂ ਨੂੰ ਤਿੰਨ ਸ਼੍ਰੇਣੀਆਂ ਵਿੱਚ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ:

1. **ਨੂਮੈਰਿਕ ਡਾਟਾ ਟਾਇਪ (Numeric Data Types):** MySQL ਵਿੱਚ ਸਾਰੀਆਂ ਜ਼ਰੂਰੀ SQL ਸੰਖਿਆਤਮਕ ਡਾਟਾ ਕਿਸਮਾਂ ਹਨ ਜਿਨ੍ਹਾਂ ਵਿੱਚ ਲੋੜੀਂਦੀਆਂ ਸੰਖਿਆਤਮਕ ਡਾਟਾ ਕਿਸਮਾਂ (ਉਦਾਹਰਨ ਲਈ INT, DECIMAL, INTEGER, ਆਦਿ) ਅਤੇ ਨਾਲ ਹੀ ਦਸਮਲਬ ਪੁਾਇੰਟ ਵਾਈਆਂ ਕਿਸਮਾਂ (ਉਦਾਹਰਨ ਲਈ, FLOAT, DOUBLE ਆਦਿ), ਸ਼ਾਮਲ ਹੋ ਸਕਦੀਆਂ ਹਨ।

2. **ਸਟ੍ਰਿੰਗ ਡਾਟਾ ਟਾਇਪ (String Data Types):** ਸਟ੍ਰਿੰਗ ਡਾਟਾ ਕਿਸਮ ਦੀ ਵਰਤੋਂ ਟੈਕਸਟ ਅਤੇ ਬਾਈਨਰੀ ਡਾਟਾ (ਜਿਵੇਂ ਕਿ ਫਾਈਲਾਂ, ਚਿੱਤਰ, ਆਦਿ) ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। MySQL ਪੈਟਰਨ ਮੈਚਿੰਗ ਜਿਵੇਂ ਕਿ LIKE ਆਪਰੇਟਰ ਦੇ ਆਧਾਰ ਤੇ ਸਟ੍ਰਿੰਗ ਮੁੱਲ ਵਿਚ ਖੋਜ ਅਤੇ ਤੁਲਨਾ ਕਾਰਜ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਸਟ੍ਰਿੰਗ ਡਾਟਾ ਟਾਇਪ ਦੀਆਂ ਕੁਝ ਉਦਾਹਰਣਾਂ CHAR(size), VARCHAR(size) ਜਾਂ ਟੈਕਸਟ (size) ਹੋ ਸਕਦੀਆਂ ਹਨ।

3. **ਡੇਟ ਅਤੇ ਟਾਈਮ ਡਾਟਾ ਟਾਇਪ (Date and Time Data Type):** ਡੇਟ ਅਤੇ ਟਾਈਮ (Date and Time) ਡਾਟਾ ਕਿਸਮ ਦੀ ਵਰਤੋਂ ਅਸਥਾਈ ਮੁੱਲਾਂ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਸਾਡੇ ਕੋਲ DATE, TIME, DATETIME ਆਦਿ ਡਾਟਾ ਟਾਈਪਸ ਹੋ ਸਕਦੀਆਂ ਹਨ।

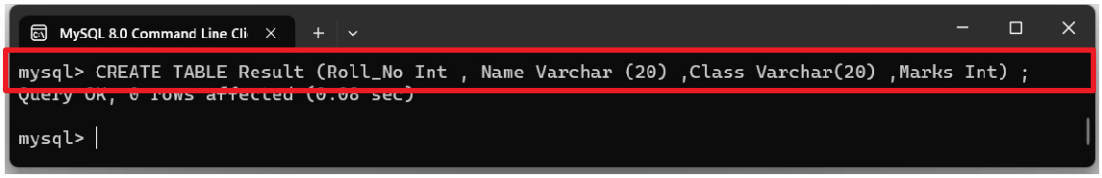
ਲੈਬ 5.2.3 CREATE ਕਮਾਂਡ (CREATE Command):

CREATE ਇੱਕ DDL ਕਮਾਂਡ ਹੈ ਜੋ ਡਾਟਾਬੇਸ, ਟੇਬਲ, ਟਰਿਗਰਸ ਅਤੇ ਹੋਰ ਡਾਟਾਬੇਸ ਆਬਜੈਕਟ ਬਣਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। CREATE ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇੱਕ ਨਵਾਂ ਟੇਬਲ ਬਣਾਉਣ ਲਈ ਸਿੰਟੈਕਸ ਅਤੇ ਉਦਾਹਰਨ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ।

ਸਿੰਟੈਕਸ:

```
CREATE TABLE table_name
(
column_Name1 data_type( size of the column ),
column_Name2 data_type( size of the column),
column_Name3 data_type( size of the column),
...
column_NameNdata_type( size of the column )
);
```

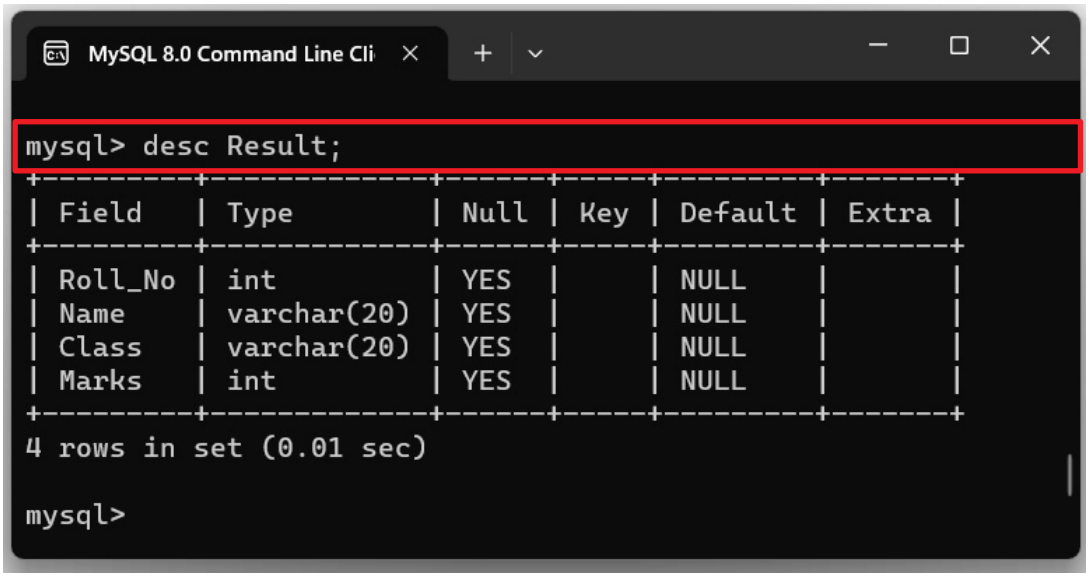
ਮੰਨ ਲਓ, ਅਸੀਂ MySQL ਵਿੱਚ ਚਾਰ ਕਾਲਮਾਂ ਦੇ ਨਾਲ ਇੱਕ Result ਟੇਬਲ ਬਣਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਸਾਨੂੰ ਹੇਠ ਦਰਸਾਏ ਅਨੁਸਾਰ ਸਟੇਟਮੈਂਟ ਲਿਖਣੀ ਪਵੇਗੀ:



ਦਿੱਤੀ ਗਈ ਕਮਾਂਡ ਸਾਡੇ ਸਲੈਕਟ ਕੀਤੇ ਡਾਟਾਬੇਸ ਵਿੱਚ ਇਕ ਨਵਾਂ ਟੇਬਲ ਬਣਾਏਗੀ।

ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ DESC ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ CREATE ਕਮਾਂਡ ਨਾਲ ਦਿੱਤੇ ਗਏ ਫੀਲਡ ਦੇ ਨਾਮ, ਡਾਟਾ ਕਿਸਮਾਂ ਅਤੇ ਹੋਰ ਜਾਣਕਾਰੀ ਬਾਰੇ ਵੇਰਵੇ ਦੇਖ ਸਕਦੇ ਹਾਂ:

ਅਸੀਂ ਹੁਣੇ ਹੀ ਪਿਛਲੀ ਕਮਾਂਡ ਵਿੱਚ ਬਣਾਇਆ ਗਿਆ Result ਨਾਂ ਦੇ ਟੇਬਲ ਦਾ ਵੇਰਵਾ ਦੇਖ ਸਕਦੇ ਹਾਂ।



ਅਸੀਂ ਹੁਣੇ ਹੀ ਪਿਛਲੀ ਕਮਾਂਡ ਵਿੱਚ ਬਣਾਇਆ ਗਿਆ Result ਨਾਂ ਦੇ ਟੇਬਲ ਦਾ ਵੇਰਵਾ ਦੇਖ ਸਕਦੇ ਹਾਂ।

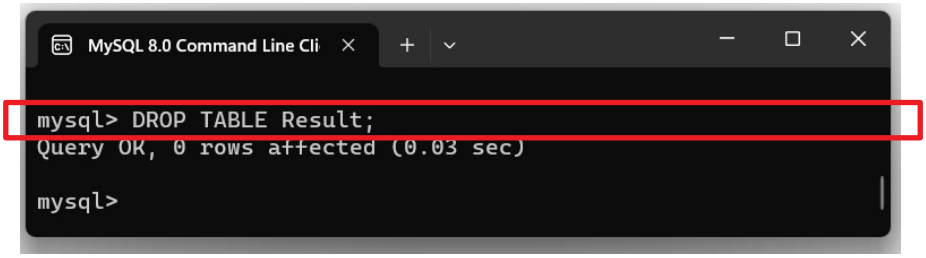
ਲੈਬ 5.2.4 DROP ਕਮਾਂਡ (DROP Command):

DROP ਇੱਕ DDL ਕਮਾਂਡ ਹੈ ਜੋ SQL ਡਾਟਾਬੇਸ ਤੋਂ ਡਾਟਾਬੇਸ ਆਬਜੈਕਟਾਂ ਨੂੰ ਮਿਟਾਉਣ/ਡਲੀਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ਇਸ DDL ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਟੇਬਲ ਦੇ ਸਟ੍ਰਕਚਰ (structure), ਵਿਊ (view) ਜਾਂ ਇੰਡੈਕਸ (index) ਨੂੰ ਆਸਾਨੀ ਨਾਲ ਡਲੀਟ ਕਰ ਸਕਦੇ ਹਾਂ। DROP ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਟੇਬਲ ਨੂੰ ਹਟਾਉਣ ਲਈ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ।

ਸਿੰਟੈਕਸ:

DROP TABLE Table_Name;

ਮੰਨ ਲਓ ਕਿ ਅਸੀਂ ਚੁਣੇ ਹੋਏ ਡਾਟਾਬੇਸ ਤੋਂ Result ਟੇਬਲ ਨੂੰ ਡਲੀਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ DROP ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:



ਦਿਖਾਈ ਗਈ ਉਦਾਹਰਨ DROP ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਅਤੇ ਡਾਟਾ ਸਮੇਤ Result ਨਾਮ ਦੇ ਟੇਬਲ ਨੂੰ ਡਲੀਟ ਕਰ ਦੇਵੇਗੀ.

ਲੈਬ 5.2.5 ALTER ਕਮਾਂਡ (ALTER Command):

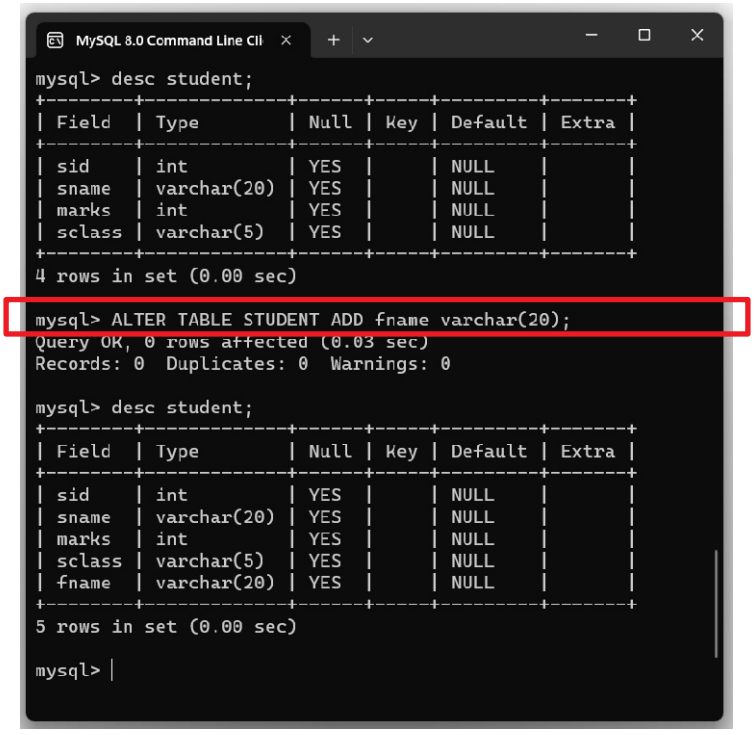
ALTER ਇੱਕ DDL ਕਮਾਂਡ ਹੈ ਜੋ ਡਾਟਾਬੇਸ ਆਬਜੈਕਟ ਜਿਵੇਂ ਕਿ ਟੇਬਲ, ਡਾਟਾਬੇਸ ਸਕੀਮਾ ਆਦਿ ਦੀ ਮੌਜੂਦਾ ਬਣਤਰ ਨੂੰ ਬਦਲਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ALTER ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਟੇਬਲ ਦੇ ਕੰਸਟ੍ਰੇਂਟਸ (constraints) ਨੂੰ ਵੀ ਸ਼ਾਮਿਲ ਜਾਂ ਡਲੀਟ ਕਰ ਸਕਦੇ ਹਾਂ। ALTER ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਨ ਦਾ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

ਮੌਜੂਦਾ ਟੇਬਲ ਵਿੱਚ ਇੱਕ ਨਵਾਂ ਫੀਲਡ ਜੋੜਨ ਲਈ ALTER ਕਮਾਂਡ:

ਸਿੰਟੈਕਸ:

ALTER TABLE table_name ADD column_name column_definition;

ਮੰਨ ਲਓ ਕਿ ਅਸੀਂ ਮੌਜੂਦਾ Student ਟੇਬਲ ਵਿੱਚ fname ਨਾਂ ਦਾ ਕਾਲਮ ਜੋੜਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੀ DDL ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:



ਅਸੀਂ ADD ਕੀਵਰਡ ਦੀ ਬਜਾਏ ਇੱਕ ਕਾਲਮ ਨੂੰ ਹਟਾਉਣ ਲਈ "DROP" ਕੀਅਵਰਡ ਅਤੇ ਇੱਕ ਕਾਲਮ ਦਾ ਆਕਾਰ, ਡਾਟਾ ਕਿਸਮ ਬਦਲਣ ਲਈ "MODIFY" ਕੀਅਵਰਡ ਜਾਂ ਇੱਕ ਕਾਲਮ ਦਾ ਨਾਮ ਬਦਲਣ ਲਈ "RENAME" ਕੀਅਵਰਡ ਦੀ ਵਰਤੋਂ ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ ਜਿਵੇਂ ਕਿ ਉੱਪਰ ਦਿੱਤੀ ਉਦਾਹਰਣ ਵਿੱਚ ਦੱਸਿਆ ਗਿਆ ਹੈ।

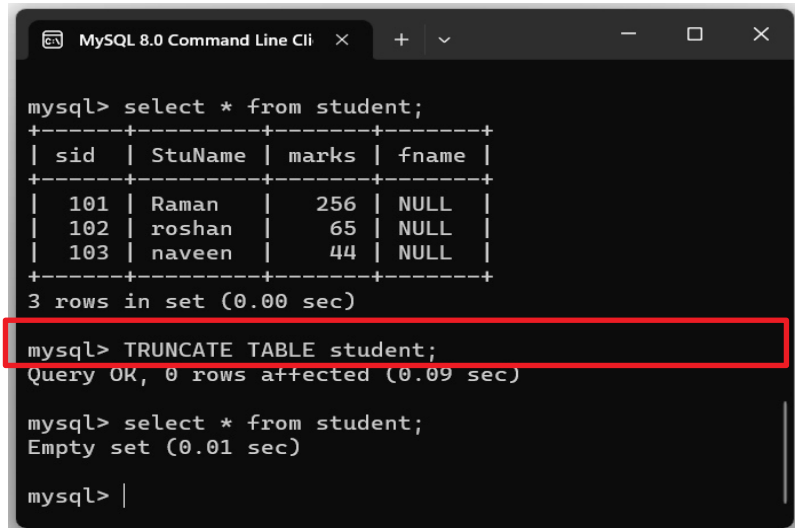
ਲੈਬ 5.2.6 TRUNCATE ਕਮਾਂਡ (TRUNCATE Command):

TRUNCATE ਇੱਕ DDL ਕਮਾਂਡ ਹੈ ਜੋ ਇੱਕ ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦਾ ਸਾਰੇ ਡਾਟਾ ਨੂੰ ਮਿਟਾ ਦਿੰਦੀ ਹੈ ਪਰ ਟੇਬਲ ਦੀ ਬਣਤਰ ਨੂੰ ਪ੍ਰਭਾਵਿਤ ਨਹੀਂ ਕਰਦੀ। TRUNCATE ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਦਾ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

ਸਿੰਟੈਕਸ:

```
TRUNCATE TABLE Table_Name;
```

ਮੰਨ ਲਓ ਕਿ ਅਸੀਂ ਮੌਜੂਦਾ Student ਟੇਬਲ ਦੇ ਸਾਰੇ ਡਾਟਾ ਨੂੰ ਡਲੀਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੀ DDL ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:



ਉਪਰੋਕਤ ਕੁਐਰੀ ਨੇ student ਟੇਬਲ ਤੋਂ ਸਾਰੇ ਰਿਕਾਰਡਾਂ ਨੂੰ ਸਫਲਤਾਪੂਰਵਕ ਮਿਟਾ ਦਿੱਤਾ ਹੈ।

ਲੈਬ 5.3 ਡਾਟਾ ਮੈਨੀਪੁਲੇਸ਼ਨ ਲੈਂਗੁਏਜ (DML) ਕਮਾਂਡਜ਼

DML ਇੱਕ ਸਟੈਂਡਰਡ ਲੈਂਗੁਐਜ਼ ਹੈ ਜੋ ਡਾਟਾਬੇਸ ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਡਾਟਾ ਤੇ ਵੱਖ-ਵੱਖ ਕੰਮਾਂ ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਕਮਾਂਡਜ਼ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। DML ਵਿੱਚ ਡਾਟਾਬੇਸ ਆਬਜੈਕਟ ਜਿਵੇਂ ਟੇਬਲ ਵਿੱਚ ਡਾਟਾ ਨੂੰ INSERT, SELECT, UPDATE ਜਾਂ DELETE ਲਈ ਸਟ੍ਰਕਚਰਡ ਕਿਊਰੀ ਲੈਂਗੁਏਜ (SQL) ਸਟੇਟਮੈਂਟਜ਼ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ। ਅਸੀਂ ਹੇਠਾਂ ਦੱਸੇ ਅਨੁਸਾਰ DML ਸਟੇਟਮੈਂਟ ਦੀ ਵਿਆਖਿਆ ਕਰ ਸਕਦੇ ਹਾਂ

1. INSERT - ਇੱਕ ਟੇਬਲ ਵਿੱਚ ਡਾਟਾ ਦਾਖਲ ਕਰਨ ਲਈ.
2. SELECT - ਇੱਕ ਟੇਬਲ ਤੋਂ ਡਾਟਾ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ।
3. UPDATE - ਇੱਕ ਟੇਬਲ ਦੇ ਅੰਦਰ ਮੌਜੂਦਾ ਡਾਟਾ ਨੂੰ ਬਦਲਣ ਲਈ।
4. DELETE - ਡਾਟਾਬੇਸ ਟੇਬਲ ਤੋਂ ਰਿਕਾਰਡਜ਼ ਨੂੰ ਮਿਟਾਓਣ/ਡਲੀਟ ਕਰ ਲਈ।

ਆਓ ਇਹਨਾਂ ਕਮਾਂਡਜ਼ ਦੀ ਵਰਤੋਂ ਨੂੰ ਵਿਸਥਾਰ ਵਿੱਚ ਸਮਝੀਏ।

ਲੈਬ 5.3.1 INSERT ਕਮਾਂਡ (INSERT Command):

INSERT ਕਮਾਂਡ ਯੂਜ਼ਰ ਨੂੰ ਡਾਟਾਬੇਸ ਟੇਬਲ ਵਿੱਚ ਡਾਟਾ ਦਾਖਲ ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦੀ ਹੈ। ਇਹ ਸਟ੍ਰਕਚਰਡ ਕਿਊਰੀ ਲੈਂਗੂਏਜ ਵਿੱਚ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਡਾਟਾ ਮੈਨੀਪੂਲੇਸ਼ਨ ਕਮਾਂਡ ਹੈ ਕਿਉਂਕਿ ਹਰੇਕ ਡਾਟਾਬੇਸ ਡਾਟਾ ਦੀ ਮੌਜੂਦਗੀ ਤੋਂ ਬਿਨਾਂ ਕੁਝ ਵੀ ਅਹਿਮੀਅਤ ਨਹੀਂ ਰੱਖਦਾ। ਅਸੀਂ MySQL ਵਿੱਚ INSERT ਕਮਾਂਡ ਦੀ ਮਦਦ ਨਾਲ ਇੱਕ ਜਾਂ ਇੱਕ ਤੋਂ ਵੱਧ ਰਿਕਾਰਡ ਸ਼ਾਮਿਲ ਕਰ ਸਕਦੇ ਹਾਂ ਜਿਵੇਂ ਕਿ ਉਦਾਹਰਣ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ।

ਸਿੰਟੈਕਸ:

ਜੇਕਰ ਅਸੀਂ ਟੇਬਲ ਦੇ ਸਾਰੇ ਫੀਲਡਜ਼ ਵਿੱਚ ਮੁੱਲਾਂ ਨੂੰ ਉਸੇ ਤਰਤੀਬ ਵਿੱਚ ਦਰਜ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਜਿਵੇਂ ਕਿ ਉਹ ਟੇਬਲ ਦੀ ਬਣਤਰ ਵਿੱਚ ਦਿਖਾਈ ਦਿੰਦੇ ਹਨ ਤਾਂ ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੇ ਸਿੰਟੈਕਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸ INSERT ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:

```
INSERT INTO table_name VALUES (value_1, value_2, value_3, .... value-N);
```

ਉਦਾਹਰਨ:

```
INSERT INTO student VALUES (101,“Ramandeep”,346,“Jasveer”);
```

ਇਸ ਤੋਂ ਇਲਾਵਾ ਅਸੀਂ INSERT ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਮੁੱਲ ਦਾਖਲ ਕਰਨ ਲਈ ਕਾਲਮ(s) ਦਾ ਨਾਮ ਵੀ ਦੇ ਸਕਦੇ ਹਾਂ। ਇਸ ਸਥਿਤੀ ਵਿੱਚ, ਕੁਐਰੀ ਵਿੱਚ ਦਿੱਤੇ ਮੁੱਲ ਦਾਖਲ ਕਰਨ ਲਈ ਜੋ ਜੋ ਕਾਲਮ ਛੱਡੇ ਗਏ (ਜੇ ਕੋਈ ਹੋਵੇ) ਤਾਂ ਉਹਨਾਂ ਕਾਲਮਾਂ ਵਿੱਚ NULL ਮੁੱਲ ਸ਼ਾਮਲ ਹੋ ਜਾਵੇਗਾ। ਦਿੱਤੇ ਫਾਰਮੈਟ ਵਿੱਚ INSERT ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਨ ਦਾ ਸਿੰਟੈਕਸ (syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਹੈ:

ਸਿੰਟੈਕਸ:

```
INSERT INTO table_name ( column_Name1, column_Name2, column_Name3,
.... column_NameN) VALUES (value_1, value_2, value_3, .... value_N ) ;
```

ਉਦਾਹਰਨ:

```
INSERT INTO student(sid,StuName,fname) VALUES(102,"Harish","Naveen");
```

ਅਸੀਂ INTO ਕਮਾਂਡ ਦੀ ਮਦਦ ਨਾਲ ਇਕੋ ਹੀ ਕਮਾਂਡ ਰਾਹੀਂ ਇੱਕ ਤੋਂ ਵੱਧ ਰਿਕਾਰਡ ਵੀ ਜੋੜ ਸਕਦੇ ਹਾਂ। ਅਜਿਹਾ ਕਰਨ ਦਾ ਸਿੰਟੈਕਸ (Syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

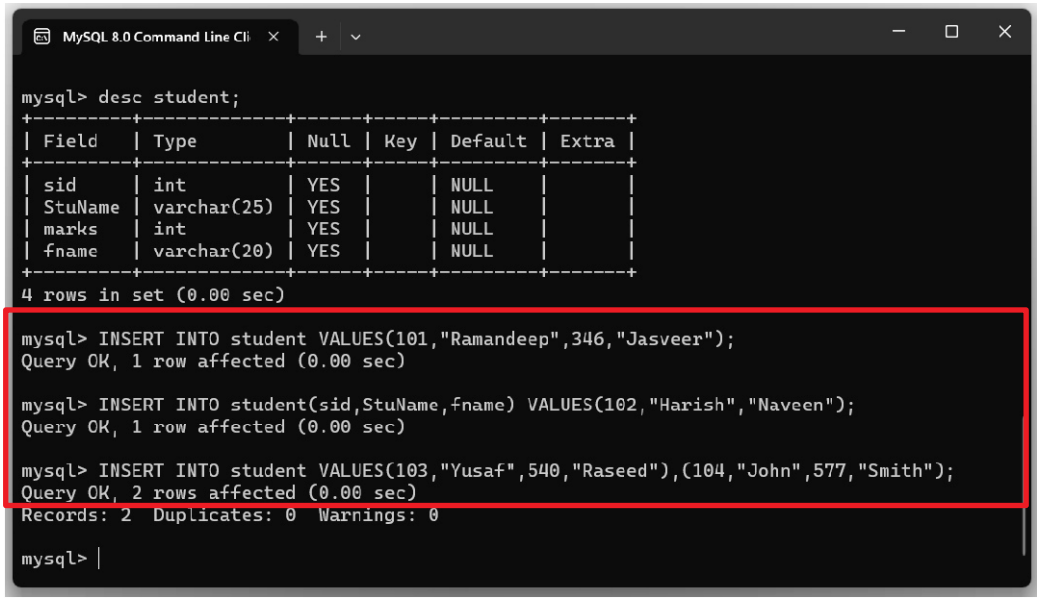
ਸਿੰਟੈਕਸ:

```
INSERT INTO table_name VALUES (value_R1_1, value_ R1_2, value_ R1_3, ....
value_ R1_N ), (value_ R2_1, value_ R2_2, value_ R2_3, .... value_ R2_N ),
(value_ R3_1, value_ R3_2, value_ R3_3, .... value_ R3_N ),..... ;
```

ਉਦਾਹਰਨ:

```
INSERT INTO student VALUES(103,"Yusaf",540,"Raseed"),(104,"John",577,"Smith");
```

ਉਕਤ ਕਮਾਂਡਜ਼ ਦੀ ਆਉਟਪੁੱਟ ਹੇਠਾਂ ਦਿਤੇ ਅਨੁਸਾਰ ਹੋਵੇਗੀ।



ਉਕਤ ਦਿੱਤੀਆਂ ਗਈਆਂ ਕਮਾਂਡਜ਼ ਨੂੰ ਲਾਗੂ ਕਰਨ ਤੋਂ ਬਾਅਦ student ਨਾਂ ਦੇ ਟੇਬਲ ਵਿੱਚ ਕੁੱਲ 4 ਰਿਕਾਰਡ ਜੋੜ ਦਿੱਤੇ ਜਾਣਗੇ। ਅਸੀਂ SELECT ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਦਾਖਲ ਕੀਤੇ ਰਿਕਾਰਡਜ਼ ਨੂੰ ਦੇਖ ਸਕਦੇ ਹਾਂ ਜਿਸ ਨੂੰ ਅਗਲੀ DML ਕਮਾਂਡ ਵਜੋਂ ਸਮਝਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।

ਲੈਬ 5.3.2 SELECT ਕਮਾਂਡ (SELECT Command):

SELECT ਕਮਾਂਡ ਕਿਸੇ ਵੀ ਦਿੱਤੇ ਗਏ ਟੇਬਲ ਦੇ ਰਿਕਾਰਡਜ਼ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਇਹ ਸਟ੍ਰਕਚਰਡ ਕੁਐਰੀ ਲੈਂਗੁਏਜ ਵਿੱਚ ਸਭ ਤੋਂ ਮਹੱਤਵਪੂਰਨ ਡਾਟਾ ਮੈਨੀਪੁਲੇਸ਼ਨ ਕਮਾਂਡ ਹੈ। ਅਸੀਂ ਡਾਟਾਬੇਸ ਤੋਂ ਰਿਕਾਰਡ ਦੇ ਲੋੜੀਂਦੇ ਸੈੱਟ ਲਈ ਹੇਠਾਂ ਦਿਤੇ ਕਲਾਜ ਦੀ ਵਰਤੋਂ SELECT ਕਮਾਂਡ ਵਿੱਚ ਕਰ ਸਕਦੇ ਹਾਂ।

- FROM - ਉਹਨਾਂ ਟੇਬਲ (ਟੇਬਲਾਂ) ਦੇ ਨਾਮ ਦੱਸਣ ਲਈ ਜਿੱਥੋਂ ਨਤੀਜਾ ਪ੍ਰਾਪਤ ਕੀਤਾ ਜਾਣਾ ਹੈ।
- WHERE – ਕੁਐਰੀ ਵਿੱਚ ਕੋਈ ਵੀ ਕੰਡੀਸ਼ਨ ਨਿਰਦਾਰਤ ਕਰਨ ਲਈ ਜਿੰਨ੍ਹਾਂ ਅਨੁਸਾਰ ਨਤੀਜਾ ਪ੍ਰਾਪਤ ਕਰਨਾ ਹੈ।
- ORDER BY - ਨਤੀਜਾ ਸੈੱਟ ਵਿੱਚ ਰਿਕਾਰਡਾਂ ਦਾ ਕ੍ਰਮ ਸੈੱਟ ਕਰਨ ਲਈ.

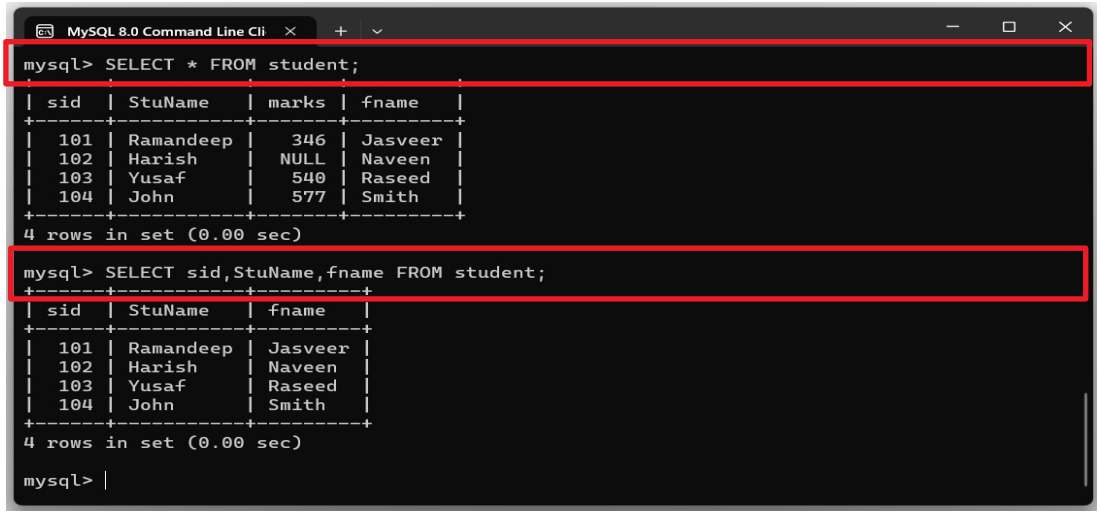
ਸਿੰਟੈਕਸ

```
SELECT column_Name_1, column_Name_2, ....., column_Name_N FROM
table_name;
```

ਇੱਥੇ column_Name_1, column_Name_2, …., column_Name_N ਨਤੀਜੇ ਸੈੱਟ ਵਿੱਚ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਕਾਲਮਾਂ ਦੇ ਨਾਮ ਹਨ। ਜੇਕਰ ਅਸੀਂ ਕਿਸੇ ਟੇਬਲ ਦੇ ਸਾਰੇ ਕਾਲਮ ਦੇਖਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਹਰੇਕ ਕਾਲਮ ਦਾ ਨਾਮ ਦੇਣ ਦੀ ਬਜਾਏ * ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ SELECT ਕਮਾਂਡ ਦੀਆਂ ਉਦਾਹਰਣਾਂ ਹੋ ਸਕਦੀਆਂ ਹਨ।

```
SELECT * FROM table_name;
```

ਹੇਠਾਂ ਦਰਸਾਏ ਅਨੁਸਾਰ ਸਾਡੇ ਕੋਲ SELECT ਕਮਾਂਡ ਦੀਆਂ ਉਦਾਹਰਣਾਂ ਹੋ ਸਕਦੀਆਂ ਹਨ:

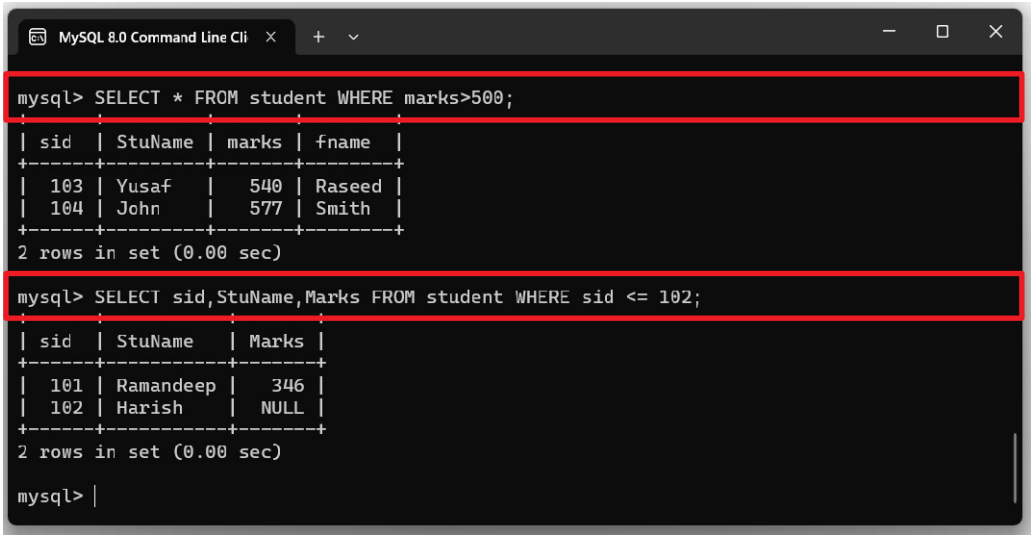


ਰਿਕਾਰਡ ਦਿਖਾਉਂਦੀ ਹੈ ਪਰੰਤੂ ਕੇਵਲ ਚੁਣੇ ਹੋਏ ਫੀਲਡ। ਜੇਕਰ ਅਸੀਂ ਕੁਝ ਚੋਣਵੇਂ ਨਤੀਜੇ ਸੈੱਟ ਕਰਨ ਲਈ ਰਿਕਾਰਡਾਂ 'ਤੇ ਫਿਲਟਰ ਲਗਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਸਿੰਟੈਕਸ ਅਤੇ ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਉਦਾਹਰਣਾਂ ਦੇ ਅਨੁਸਾਰ SELECT ਕਮਾਂਡ ਵਿੱਚ WHERE ਕਲਾਜ਼ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:

ਸਿੰਟੈਕਸ:

```
SELECT * FROM table_name WHERE condition;
```

ਅਸੀਂ ਹੇਠਾਂ ਦਰਸਾਈ ਉਦਾਹਰਣ ਦੇ ਅਨੁਸਾਰ SELECT ਕਮਾਂਡ ਵਿੱਚ WHERE ਕਲੌਜ਼ (clause) ਦੀ ਵਰਤੋਂ ਕਰਨ ਦੀਆਂ ਵੱਖ ਵੱਖ ਕਿਸਮਾਂ ਦੇਖ ਸਕਦੇ ਹਾਂ:



ਅਸੀਂ student ਟੇਬਲ ਦੇ ਸਾਰੇ ਫੀਲਡਜ਼ ਜਾਂ ਚੋਣਵੇਂ ਫੀਲਡਜ਼ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ ਉਦਾਹਰਣਾਂ ਉਕਤ ਅਨੁਸਾਰ ਦੇਖ ਸਕਦੇ ਹਾਂ।

ਲੈਬ 5.3.3 DELETE ਕਮਾਂਡ (DELETE Command):

DELETE ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਰਿਕਾਰਡਜ਼ ਨੂੰ ਡਲੀਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ਮਿਟਾਏ ਜਾਣ ਵਾਲੇ ਰਿਕਾਰਡਜ਼ ਦੀ ਚੋਣ ਕਰਦੇ ਸਮੇਂ ਕੰਡੀਸ਼ਨਾਂ ਨੂੰ ਨਿਰਧਾਰਿਤ ਕਰਨ ਲਈ ਇਸ DELETE ਕਮਾਂਡ ਦੇ ਨਾਲ WHERE ਕਲਾਜ਼ (clause) ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਸਾਰੇ ਓਪਰੇਟਰਾਂ ਨੂੰ DELETE ਕਮਾਂਡ ਨਾਲ ਵੀ ਉਸੇ ਤਰ੍ਹਾਂ

ਹੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਿਵੇਂ ਕਿ SELECT ਕਮਾਂਡ ਵਿੱਚ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਕਮਾਂਡ ਦਾ ਸਿੰਟੈਕਸ (syntax) ਅਤੇ ਉਦਾਹਰਨ (example) ਇਸ ਪ੍ਰਕਾਰ ਹੈ:

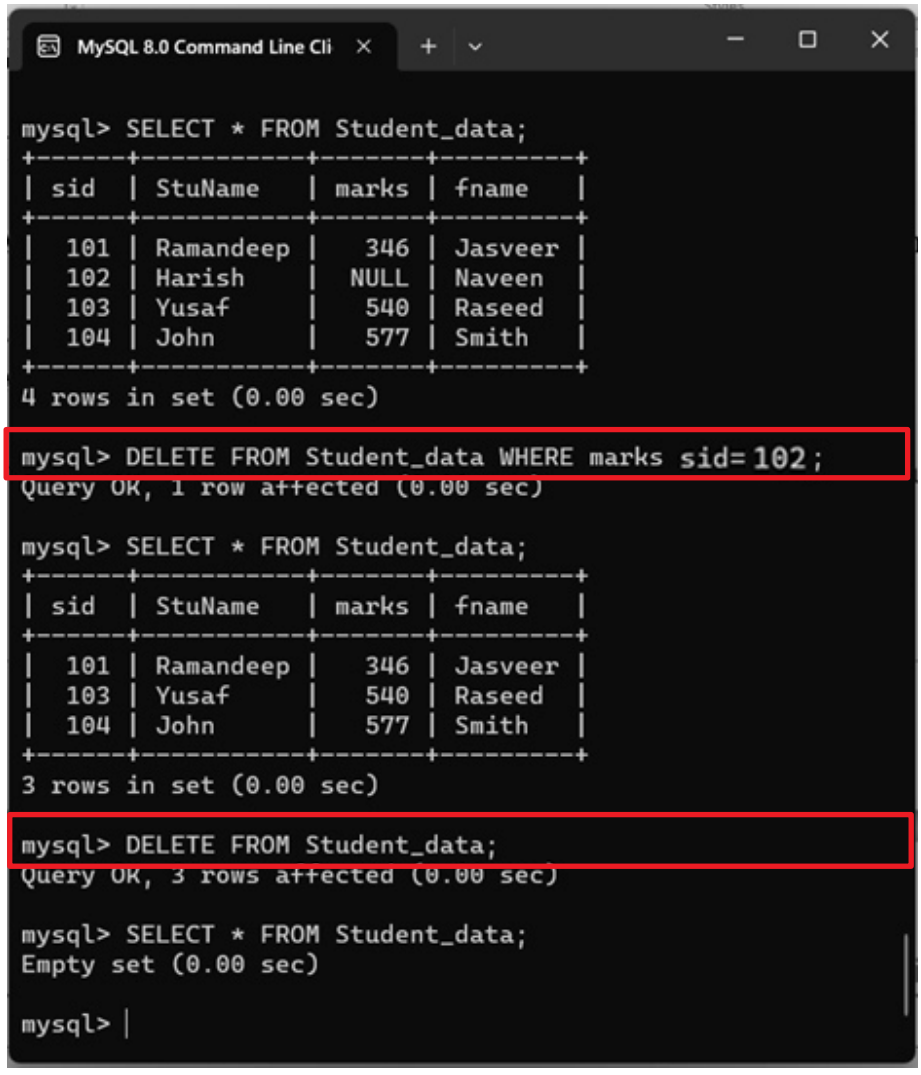
ਸਿੰਟੈਕਸ:

```
DELETE FROM table_name WHERE condition;
```

ਉਦਾਹਰਣ:

```
DELETE FROM Student_data WHERE sid=102;
```

ਜੇਕਰ ਅਸੀਂ WHERE ਕਲਾਜ਼(clause) ਦੀ ਵਰਤੋਂ ਕੀਤੇ ਬਿਨਾਂ DELETE ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ ਤਾਂ ਇਹ ਟੇਬਲ ਤੋਂ ਸਾਰੇ ਰਿਕਾਰਡਾਂ ਨੂੰ ਮਿਟਾ ਦੇਵੇਗਾ। ਹਾਲਾਂਕਿ, ਟੇਬਲ ਦਾ ਢਾਂਚਾ ਮੌਜੂਦਾ ਰਹੇਗਾ।



ਲੈਬ 5.3.4 UPDATE ਕਮਾਂਡ (UPDATE Command):

UPDATE MySQL ਵਿੱਚ ਵਰਤੀ ਜਾਣ ਵਾਲੀ ਇੱਕ ਹੋਰ ਸਭ ਤੋਂ ਮਹੱਤਵਪੂਰਨ ਡਾਟਾ ਮੈਨੂਪੁਲੇਸ਼ਨ ਕਮਾਂਡ ਹੈ ਜੋ ਸਾਨੂੰ ਡਾਟਾਬੇਸ ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦਾ ਡਾਟਾ ਨੂੰ ਸ਼ੋਧਣ/ਤਬਦੀਲ ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦੀ ਹੈ। ਕੀਵਰਡ SET ਦੀ ਵਰਤੋਂ ਟੇਬਲ ਵਿੱਚ ਕੀਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਸੋਧਾਂ ਨੂੰ ਨਿਰਦਾਰਤ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ਹੇਠ ਲਿਖੇ ਸਿੰਟੈਕਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸ ਕਮਾਂਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:

ਸਿੰਟੈਕਸ:

```
UPDATE table_name SET column_name1= value_1, ....., column_nameN =
value_N WHERE CONDITION;
```

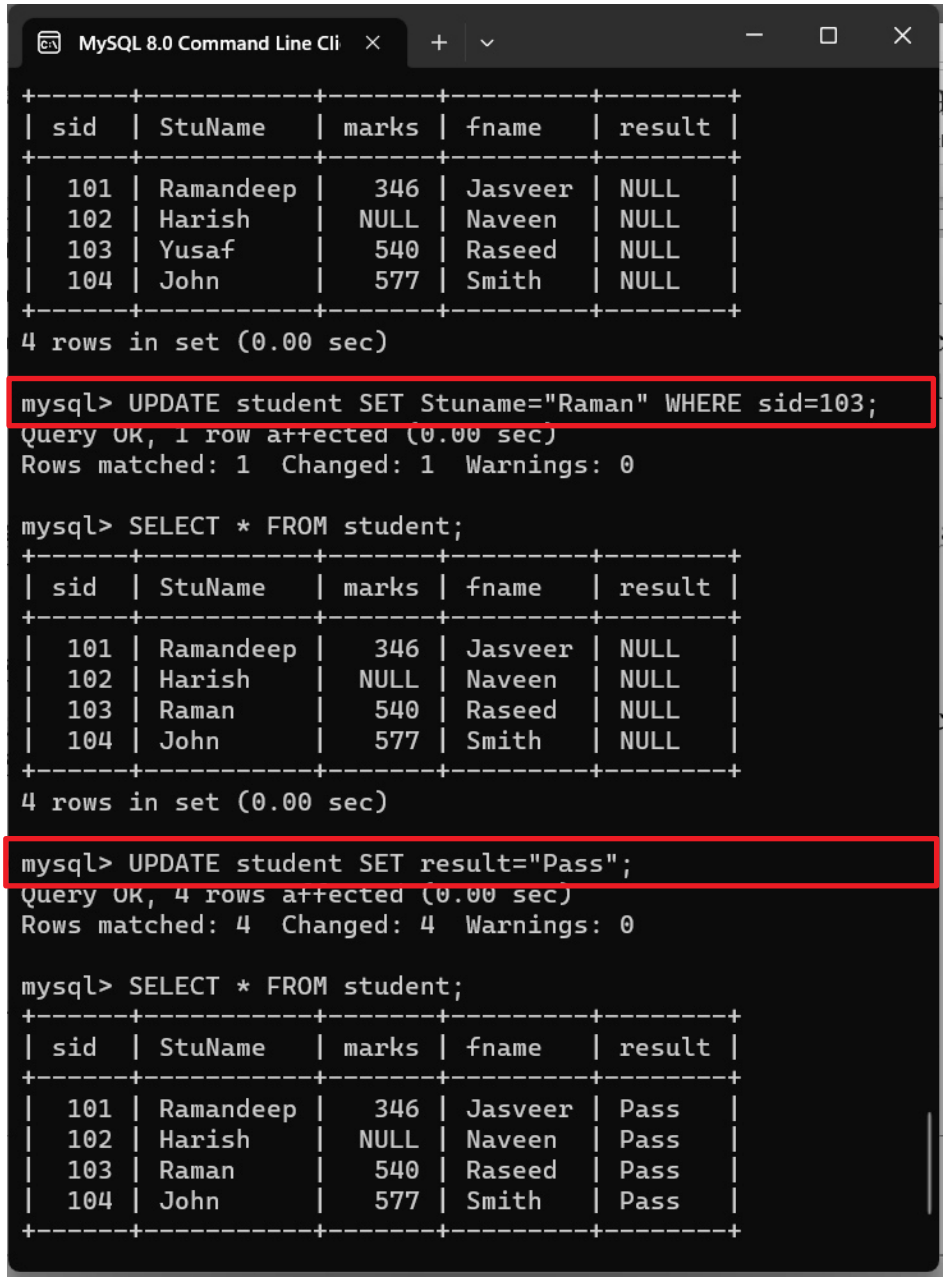
ਉਦਾਹਰਨ:

```
UPDATE student SET Stu_name="Raman" WHERE sid=103;
```

ਜੇਕਰ ਅਸੀਂ ਸਾਰੇ ਰਿਕਾਰਡਜ਼ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਬਦਲਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ WHERE ਕਲਾਜ਼ (clause) ਨੂੰ ਛੱਡ ਸਕਦੇ ਹਾਂ। ਜਿਵੇ ਕੀ:

```
UPDATE student SET result="Pass";
```

ਦਿੱਤੀਆਂ ਕਮਾਂਡਜ਼ ਦੀ ਐਗਜ਼ੀਕਿਊਸ਼ਨ ਹੇਠਾਂ ਦਿਖਾਏ ਅਨੁਸਾਰ ਹੋ ਸਕਦੀ ਹੈ:



ਲੈਬ 5.3.5 ORDER BY ਕਲੌਜ (ORDER BY clause):

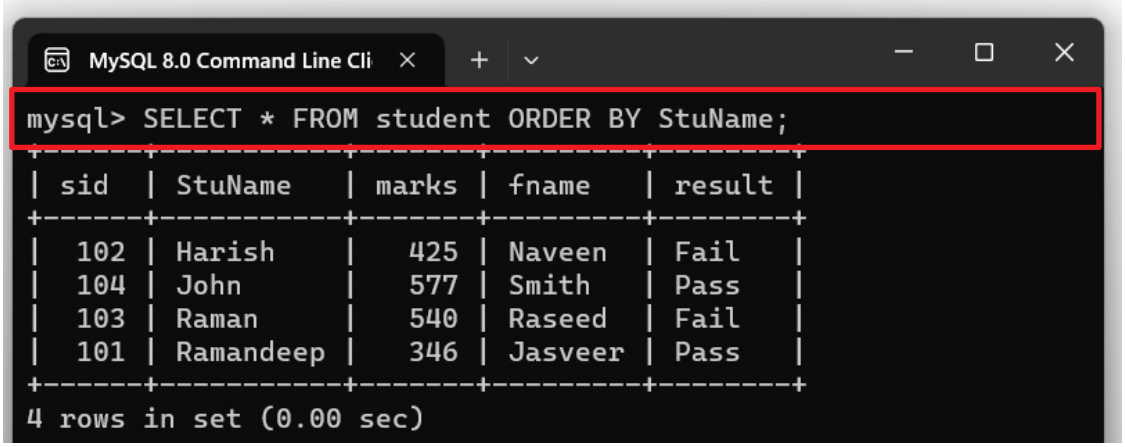
ORDER BY ਕਲਾਜ਼ (clause) ਦੀ ਵਰਤੋਂ ਨਤੀਜੇ ਨੂੰ ਵੱਧਦੇ ਜਾਂ ਘਟਦੇ ਕ੍ਰਮ ਵਿੱਚ ਤਰਤੀਵ ਵਿੱਚ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਕਮਾਂਡ ਮੂਲ ਰੂਪ ਵਿੱਚ ਨਤੀਜੇ ਨੂੰ ਵਧਦੇ ਕ੍ਰਮ ਵਿੱਚ ਕ੍ਰਮਬਧ ਕਰਦੀ ਹੈ। ਰਿਕਾਰਡਾਂ ਨੂੰ ਘਟਦੇ ਕ੍ਰਮ ਵਿੱਚ ਕ੍ਰਮਬੱਧ ਕਰਨ ਲਈ ਅਸੀਂ ਇਸ ਦੇ ਨਾਲ DESC ਕੀਵਰਡ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੇ ਸਿੰਟੈਕਸ ਰਾਹੀਂ SELECT ਕਮਾਂਡ ਵਿੱਚ ਇਸ ਕਲਾਜ਼ (clause) ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ:

ਸਿੰਟੈਕਸ

```
SELECT * FROM table_name ORDER BY column1,column2,....;
```

ਉਦਾਹਰਣ

```
SELECT * FROM student ORDER BY Stuname;
```



ਉਦਾਹਰਣ:

```
SELECT * FROM student ORDER BY Stuname DESC;
```

ਉੱਪਰ ਦਿੱਤੀ ਕਮਾਂਡ ਘਟਦੇ ਕ੍ਰਮ ਵਿੱਚ ਟੇਬਲ ਵਿਚਲੇ ਡਾਟਾ ਨੂੰ ਕ੍ਰਮਬੱਧ ਕਰੇਗੀ।

MySQL ਵਿੱਚ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਇਨ-ਬਿਲਟ ਫੰਕਸ਼ਨਾਂ ਦੀ ਇੱਕ ਵਿਸ਼ਾਲ ਲਾਇਬ੍ਰੇਰੀ ਹੈ। ਅਸੀਂ ਹੁਣ ਤੱਕ ਸਿਰਫ਼ ਆਮ ਤੌਰ ਤੇ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਫੰਕਸ਼ਨਾਂ ਬਾਰੇ ਹੀ ਚਰਚਾ ਕੀਤੀ ਹੈ। ਅਸੀਂ MySQL ਵਿੱਚ ਮੌਜੂਦ ਹੋਰ ਕੁਐਰੀਜ਼ ਅਤੇ ਫੰਕਸ਼ਨਾਂ ਦੇ ਸਬੰਧ ਵਿੱਚ ਕਿਤਾਬਾਂ ਅਤੇ ਔਨਲਾਈਨ ਸਰੋਤਾਂ ਤੋਂ ਹੋਰ ਸਮੱਗਰੀ ਪ੍ਰਾਪਤ ਕਰ ਸਕਦੇ ਹਾਂ।

ਯਾਦ ਰੱਖਣ ਯੋਗ ਨੁਕਤੇ

- MySQL ਇੱਕ ਡਾਟਾਬੇਸ ਬਣਾਉਂਦਾ ਹੈ ਜੋ ਸਾਡੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਕਰਨ ਅਤੇ ਉਸ ਵਿਚ ਲੋੜੀਂਦੇ ਬਦਲਾਵ ਕਰਨ ਅਤੇ ਵੱਖ-ਵੱਖ ਟੇਬਲਾਂ ਦੇ ਵਿਚਕਾਰ ਸਬੰਧਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦਾ ਹੈ।
- ਅਸੀਂ MySQL ਨੂੰ **Start -> Search "MySQL Command Line Client" ->** ਪ੍ਰੋਗਰਾਮ ਆਈਕਨ ਤੇ ਕਲਿੱਕ ਕਰ ਕੇ ਖੋਲ ਸਕਦੇ ਹਾਂ।
- MySQL ਦੀਆਂ ਮੁੱਖ ਡਾਟਾ ਟਾਈਪਸ ਨੂੰ ਤਿੰਨ ਸ਼੍ਰੇਣੀਆਂ ਵਿੱਚ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ: ਨੂਮੈਰਿਟਕ ਡਾਟਾ ਟਾਈਪਸ, ਸਟ੍ਰਿੰਗ ਡਾਟਾ ਟਾਈਪਸ, ਡੇਟ ਅਤੇ ਟਾਇਮ ਡਾਟਾ ਟਾਈਪਸ।
- DDL ਇੱਕ ਆਮ ਭਾਸ਼ਾ ਹੈ ਜੋ ਇੱਕ ਡਾਟਾਬੇਸ ਵਿੱਚ ਸਟੋਰੇਜ਼ ਗਰੁੱਪਸ, ਵੱਖ-ਵੱਖ ਢਾਂਚੇ ਅਤੇ ਆਬਜੈਕਟਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਕਮਾਂਡਜ਼ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।

- DML ਇੱਕ ਆਮ ਭਾਸ਼ਾ ਹੈ ਜੋ ਡਾਟਾਬੇਸ ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦ ਡਾਟਾ ਨੂੰ ਤਬਦੀਲ ਕਰਨ ਲਈ ਕਮਾਂਡਜ਼ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।
- ਅਸੀਂ ਆਪਣੀ MySQL ਕੁਐਰੀਜ਼ ਨੂੰ ਹੋਰ ਗੁੰਝਲਦਾਰ ਬਣਾਉਣ ਲਈ ਕਈ ਲਾਜ਼ੀਕਲ ਓਪਰੇਟਰਾਂ ਜਿਵੇਂ ਕਿ AND, OR, NOT, BETWEEN, IN, LIKE, IS NULL ਆਦਿ ਨਾਲ ਕੁਐਰਜ਼ ਡਿਜ਼ਾਈਨ ਕਰ ਸਕਦੇ ਹਾਂ।
- ORDER BY ਕਲਾਜ਼ ਦੀ ਵਰਤੋਂ ਨਤੀਜੇ ਨੂੰ ਵੱਧਦੇ ਜਾਂ ਘਟਦੇ ਕ੍ਰਮ ਵਿੱਚ ਸੈੱਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ. ਇਹ ਕਮਾਂਡ ਮੂਲ ਰੂਪ ਵਿੱਚ ਵਧਦੇ ਕ੍ਰਮ ਵਿੱਚ ਨਤੀਜੇ ਨੂੰ ਕ੍ਰਮਬੱਧ ਕਰਦੀ ਹੈ।

ਲੈਬ ਗਤੀਵਿਧੀਆਂ

Activity No: 5.5

ਦਿੱਤੇ ਗਏ ਆਪ੍ਰੇਸ਼ਨਾਂ ਨੂੰ ਕਰਨ ਲਈ ਹੇਠਾਂ ਲਿਖਿਆ DDL Commands ਨੂੰ ਪੂਰਾ ਕਰੋ।

1. ਇੱਕ ਨਵਾਂ ਟੇਬਲ ਬਨਾਉਣ ਲਈ :

   CREATE _____ table_name

   (

   column_Name1 data_type (size of the column) ,

   ...

   column_NameNdata_type (size of the column)

) ;

2. ਮੌਜੂਦਾ ਟੇਬਲ ਦੀ ਬਣਤਰ ਮਿਟਾਉਣ ਲਈ :

   _____ TABLE Table_Name;

3. ਮੌਜੂਦਾ ਟੇਬਲ ਵਿੱਚ ਇੱਕ ਨਵਾਂ ਕਾਲਮ ਦਾਖਲ ਕਰਨ ਲਈ :

   ALTER TABLE table_name _____ column_namecolumn_definition;

4. ਟੇਬਲ ਵਿੱਚ ਮੌਜੂਦਾ ਕਾਲਮ ਦਾ ਨਾਂ ਬਦਲਣ ਲਈ :

   ALTER TABLE table_name _____ COLUMN Old_column_name TO New_column_name;

5. ਮੌਜੂਦਾ ਟੇਬਲ ਵਿੱਚੋਂ ਸਾਰੇ ਰਿਕਾਰਡ ਮਿਟਾਉਣ ਲਈ :

   _____ TABLE Table_Name;

Activity No: 5.6

MySQL ਵਿੱਚ ਦਿੱਤਿਆਂ ਗਈਆਂ ਕਮਾਂਡਾਂ ਦਾ ਮੰਤਵ ਦੱਸੋ।

1. USE Database_Name;

   _____

2. ALTER TABLE table_name DROP Column_Name_1;

   _____

3. INSERT INTO student VALUES(101,"Ramandeep",346,"Jasveer");

   _____

4. SELECT column_Name_1, column_Name_2 FROM table_name;

   _____

5. SELECT * FROM student where sid>=103 AND marks>550;

   _____

Activity No: 5.7

ਹੇਠਾਂ ਦਿੱਤਿਆਂ ਕਮਾਂਡਾਂ ਨੂੰ MySQL ਵਿੱਚ ਲਾਗੂ ਕਰੋ ਅਤੇ ਪ੍ਰਾਪਤ ਹੋਏ ਨਤੀਜੇ ਨੂੰ ਕਾਪੀ ਤੇ ਨੋਟ ਕਰੋ।

1. CREATE DATABASE LABACTIVITY;
2. SHOW DATABASES;
3. USE LABACTIVITY;
4. CREATE TABLE PRACTICAL(SID INT, SUBJECT VARCHAR(25), PERIOD INT, TOPIC VARCHAR(30), MINUTE INT);
5. INSERT INTO PRACTICAL VALUES(1001, "COMPUTER SCIENCE", 7, "MSWORD", 35);
6. SELECT * FROM PRACTICAL;
7. INSERT INTO PRACTICAL VALUES(1002, "IT/ITES", 4, "PAINT", 20),(1003,"COMPUTER APPLICATION",2,"PYTHON",38);
8. SELECT * FROM PRACTICAL;
9. INSERT INTO PRACTICAL(SID, SUBJECT) VALUES(1004,"DIGITAL ELECTRONICS");
10. SELECT * FROM PRACTICAL;
11. ALTER TABLE PRACTICAL ADD COLUMN CLASS VARCHAR(10);
12. SELECT * FROM PRACTICAL;
13. UPDATE PRACTICAL SET CLASS="11TH" WHERE CLASS IS NULL;
14. SELECT * FROM PRACTICAL;
15. UPDATE PRACTICAL SET PERIOD=1, TOPIC="EXCEL",MINUTE=30 WHERE SID=1004;
16. SELECT * FROM PRACTICAL;
17. SELECT SID,SUBJECT, MINUTE FROM PRACTICAL;
18. SELECT SID,SUBJECT, TOPIC FROM PRACTICAL WHERE MINUTE<=30;
19. SELECT * FROM PRACTICAL ORDER BY MINUTE;
20. SELECT * FROM PRACTICAL ORDER BY MINUTE DESC;

ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਮੈਨਟੇਨੈਂਸ

ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ ਉਹਨਾਂ ਕੰਮਾਂ ਜਾਂ ਪ੍ਰਕਿਰਿਆਵਾਂ ਦੇ ਸਮੂਹ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ, ਜਿਹਨਾਂ ਦਾ ਉਦੇਸ਼ ਕੰਪਿਊਟਰ ਉਪਕਰਣਾਂ ਦੇ ਸਹੀ ਕੰਮ ਕਾਜ ਨੂੰ ਯਕੀਨੀ ਬਣਾਉਣਾ ਹੈ। ਬਰੇਕਡਾਊਨ ਅਤੇ ਹੋਰ ਕਈ ਤਰ੍ਹਾਂ ਦੀਆਂ ਸਮੱਸਿਆਵਾਂ ਤੋਂ ਬਚਣ ਲਈ ਕੰਪਿਊਟਰ ਦੀ ਨਿਯਮਿਤ ਦੇਖਭਾਲ ਕਰਨਾ ਜ਼ਰੂਰੀ ਹੁੰਦਾ ਹੈ।

ਪਾਠ ਦੇ ਉਦੇਸ਼

- **ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ:** ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ, ਸਾਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ, ਕੰਪਿਊਟਰ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਬਿਹਤਰ ਬਣਾਉਣ ਲਈ ਟੂਲਜ਼ ਅਤੇ ਤਕਨੀਕਾਂ
- ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ, ਥਿਨ ਕਲਾਇੰਟ ਟੈਕਨੋਲੋਜੀ, ਬੂਟਿੰਗ ਅਤੇ ਸੇਫ-ਮੋਜ, ਕੰਟਰੋਲ ਪੈਨਲ, ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮਜ਼, ਸਾਫਟਵੇਅਰ ਅਪਡੇਟ ਅਤੇ ਅਪਗਰੇਡ
- ਪੋਰਟਸ ਅਤੇ ਉਹਨਾਂ ਦੀਆਂ ਕਿਸਮਾਂ, ਡਿਵਾਈਸ ਡਰਾਈਵਰਜ਼
- PC ਸਿਕਿਉਰਟੀ ਟੂਲਜ਼

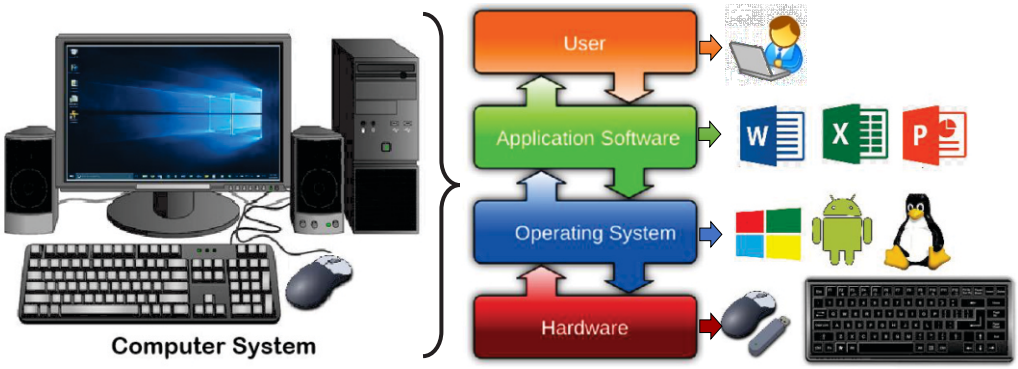
ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ਕੰਪਿਊਟਰ ਮੈਨਟੇਨਸ ਦੇ ਸੰਕਲਪ ਅਤੇ ਮਹੱਤਤਾ ਨੂੰ ਸਮਝਣਾ।
- ਵਿਦਿਆਰਥੀ ਸਿੱਖਣਗੇ ਕਿ ਆਪਣੇ PC ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਕਿਵੇਂ ਵਧਾਉਣਾ ਹੈ।
- ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਅਤੇ ਥਿਨ ਕਲਾਇੰਟ ਤਕਨਾਲੋਜੀ ਦੇ ਸੰਕਲਪ ਨੂੰ ਸਮਝਣਾ।
- ਵਿਦਿਆਰਥੀ ਇਹ ਸਿੱਖਣ ਯੋਗ ਹੋਣਗੇ ਕਿ ਸੇਫ-ਮੋਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ PC ਨਾਲ ਸਬੰਧਤ ਸਮੱਸਿਆਵਾਂ ਦਾ ਨਿਪਟਾਰਾ ਕਿਵੇਂ ਕਰਨਾ ਹੈ।
- ਵਿਦਿਆਰਥੀ ਕੰਟਰੋਲ ਪੈਨਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਆਪਣੇ PC ਨੂੰ ਕੌਨਫਿਗਰ ਕਰਨ ਯੋਗ ਹੋਣਗੇ।
- ਵਿਦਿਆਰਥੀ PC ਲਈ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਵੱਖ-ਵੱਖ ਯੂਟੀਲੀਟੀਜ਼ ਬਾਰੇ ਜਾਣ ਸਕਣਗੇ।
- ਪੋਰਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਡਿਵਾਈਸਾਂ ਨੂੰ ਕਨੈਕਟ ਕਰਨ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਣਗੇ।
- ਡਿਵਾਈਸ ਡਰਾਈਵਰਾਂ ਅਤੇ PnP (ਪਲੱਗ ਐਂਡ ਪਲੇਅ) ਡਿਵਾਈਸਾਂ ਦੇ ਸੰਕਲਪ ਨੂੰ ਸਮਝਣਗੇ।
- ਵਿਦਿਆਰਥੀ ਵੱਖ-ਵੱਖ PC ਸਿਕਿਉਰਟੀ ਟੂਲਜ਼ ਅਤੇ ਉਹਨਾਂ ਦੀ ਮਹੱਤਤਾ ਬਾਰੇ ਸਮਝਣਗੇ।

06
ਪਾਠ

ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਮੈਨਟੇਨੈਂਸ

ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਦੋ ਮੁੱਖ ਭਾਗ ਹੁੰਦੇ ਹਨ: ਹਾਰਡਵੇਅਰ ਅਤੇ ਸਾਫਟਵੇਅਰ। ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੁਆਰਾ ਸਾਡੇ ਕੰਮ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਹਾਰਡਵੇਅਰ ਅਤੇ ਸਾਫਟਵੇਅਰ ਦੋਨਾਂ ਦੀ ਜ਼ਰੂਰਤ ਹੁੰਦੀ ਹੈ। ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਬਿਨਾਂ ਕਿਸੇ ਖਰਾਬੀ ਦੇ ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਢੰਗ ਨਾਲ ਕੰਮ ਕਰਦਾ ਰਹੇ, ਇਸ ਮੰਤਵ ਲਈ ਉਸ ਦੀ ਮੈਨਟੇਨੈਂਸ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ।



ਚਿੱਤਰ 6.1: ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਅਤੇ ਇਸਦੇ ਭਾਗ

ਇਸ ਪਾਠ ਵਿੱਚ ਅਸੀਂ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀ ਮੈਨਟੇਨੈਂਸ, ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ, ਥਿਨ-ਕਲਾਇੰਟ ਤਕਨਾਲੋਜੀ, ਡਿਵਾਈਸ ਡਰਾਈਵਰਜ਼, ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀਆਂ ਪੋਰਟਸ, ਸਿਸਟਮ ਦੀ ਸਕਿਊਰਟੀ ਆਦਿ ਨਾਲ ਸਬੰਧਤ ਵੱਖ-ਵੱਖ ਸੰਕਲਪਾਂ ਬਾਰੇ ਚਰਚਾ ਕਰਾਂਗੇ।

6.1 ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਮੈਨਟੇਨੈਂਸ

ਰੱਖ-ਰਖਾਵ ਦੀਆਂ ਉਹ ਵੱਖ-ਵੱਖ ਗਤੀਵਿਧੀਆਂ ਜਿਹਨਾਂ ਨਾਲ ਅਸੀਂ ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਜਾਂ ਕੰਪਿਊਟਰ ਦੇ ਭਾਗਾਂ ਨੂੰ ਉਹਨਾਂ ਦੇ ਕੰਮ ਕਾਜ ਦੀ ਚੰਗੀ ਸਥਿਤੀ ਵਿੱਚ ਰੱਖਦੇ ਹਾਂ, ਨੂੰ ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਮੈਨਟੇਨੈਂਸ ਜਾਂ ਰੱਖ-ਰਖਾਅ ਲਈ, ਸਾਨੂੰ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਹਾਰਡਵੇਅਰ ਅਤੇ ਸਾਫਟਵੇਅਰ ਦੋਵਾਂ ਭਾਗਾਂ ਦਾ ਧਿਆਨ ਰੱਖਣ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ:

- **ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ:** ਇਸ ਵਿੱਚ ਕੰਪਿਊਟਰ ਦੇ ਭੌਤਿਕ ਭਾਗਾਂ ਦੀ ਦੇਖਭਾਲ ਕਰਨਾ ਸ਼ਾਮਲ ਹੈ, ਜਿਵੇਂ ਕਿ ਇਸਦਾ ਕੀਬੋਰਡ, ਮਾਊਸ, ਹਾਰਡ ਡਰਾਈਵ ਆਦਿ। ਉਦਾਹਰਨ ਲਈ: ਕੰਪਿਊਟਰ ਨੂੰ ਸਾਫ਼ ਕਰਨਾ, ਇਸਦੇ ਪੱਖਿਆਂ (ਫੈਨ) ਨੂੰ ਧੂੜ ਤੋਂ ਮੁਕਤ ਰੱਖਣਾ, ਅਤੇ ਇਸ ਦੀ ਹਾਰਡ ਡਰਾਈਵ ਨੂੰ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਡੀਫ੍ਰੈਗਮੈਂਟ ਕਰਨਾ ਆਦਿ।
- **ਸਾਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ:** ਜ਼ਿਆਦਾਤਰ ਕੰਪਿਊਟਰ ਕੁਝ ਸਮੇਂ ਤੋਂ ਬਾਅਦ ਹੌਲੀ ਕੰਮ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰ ਦਿੰਦੇ ਹਨ। ਜੇਕਰ ਸਾਡੇ ਕੰਪਿਊਟਰ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਹੌਲੀ ਹੁੰਦੀ ਜਾਪਦੀ ਹੈ, ਤਾਂ ਇਹ ਕੰਪਿਊਟਰ ਵਿੱਚ ਇੰਸਟਾਲਡ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਸਾਫਟਵੇਅਰਾਂ ਦੀ ਮੈਨਟੇਨੈਂਸ ਕਰਨ ਦਾ ਸਮਾਂ ਹੋ ਸਕਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ: ਸਿਸਟਮ ਅੱਪਡੇਟ ਇੰਸਟਾਲ ਕਰਨਾ, ਇੰਟਰਨੈਟ ਬ੍ਰਾਊਜ਼ਰ ਦਾ ਕੈਸ਼ ਸਾਫ਼ ਕਰਨਾ, ਐਂਟੀ-ਵਾਇਰਸ ਸਾਫਟਵੇਅਰ ਨੂੰ ਅਪਡੇਟ ਕਰਨਾ ਆਦਿ।

ਸਾਡੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨੂੰ ਹਮੇਸ਼ਾ ਚੱਲਦੀ ਹਾਲਤ ਵਿੱਚ ਰੱਖਣ ਲਈ, ਸਾਨੂੰ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਇਸਦੀ ਦੇਖਭਾਲ ਜਾਂ ਮੈਨਟੇਨੈਂਸ ਕਰਨ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ। ਨਿਯਮਤ ਰੱਖ-ਰਖਾਅ ਡਿਵਾਈਸਾਂ ਨੂੰ ਸੁਚਾਰੂ ਢੰਗ ਨਾਲ ਚੱਲਣ ਅਤੇ ਮਹਿੰਗੀ ਮੁਰੰਮਤ ਤੋਂ ਬਚਣ ਵਿੱਚ ਮਦਦ ਕਰਦਾ ਹੈ।

ਜੇਕਰ ਸਾਡਾ PC ਹਮੇਸ਼ਾ ਸਾਫ਼ ਸੁਥਰੇ ਵਾਤਾਵਰਣ ਵਿੱਚ ਰਹਿੰਦਾ ਹੈ, ਤਾਂ ਉਸਦੀ ਸਾਲਾਨਾ ਇੱਕ ਵਾਰ ਸਫਾਈ ਕਰਨਾ ਕਾਫ਼ੀ ਹੋਵੇਗਾ। ਪਰ ਜ਼ਿਆਦਾਤਰ ਜਗ੍ਹਾਵਾਂ ਉੱਤੇ, ਜਿਵੇਂ ਕਿ ਧੂੜ ਭਰੇ ਦਫਤਰਾਂ ਜਾਂ ਦੁਕਾਨਾਂ ਵਿਚ, ਸਾਡੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮਾਂ ਨੂੰ ਕੁਝ ਦਿਨਾਂ ਜਾਂ ਮਹੀਨਿਆ ਵਿੱਚ ਸਫਾਈ ਦੀ ਲੋੜ ਪੈ ਸਕਦੀ ਹੈ।

ਆਉ ਹੁਣ ਹਾਰਡਵੇਅਰ ਅਤੇ ਸਾਫਟਵੇਅਰ ਦੇ ਮੈਨਟੇਨੈਂਸ ਲਈ ਵੱਖ-ਵੱਖ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼ਾਂ ਤੇ ਚਰਚਾ ਕਰੀਏ:

6.1.1 ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ:

ਅਸੀਂ ਜਾਣਦੇ ਹਾਂ ਕਿ ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਵਿੱਚ ਕੰਪਿਊਟਰ ਦੇ ਭੌਤਿਕ ਭਾਗਾਂ, ਜਿਵੇਂ ਕਿ ਕੀਬੋਰਡ, ਮਾਊਸ, ਹਾਰਡ ਡਰਾਈਵ ਅਤੇ ਹੋਰ ਪੈਰੀਫਿਰਲਾਂ (peripherals) ਦੀ ਦੇਖਭਾਲ ਕਰਨਾ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ। ਇਹ ਯਕੀਨੀ ਬਣਾਉਣ ਲਈ ਕਿ ਕੰਪਿਊਟਰ ਦੇ ਭੌਤਿਕ ਭਾਗ ਵਧੀਆ ਢੰਗ ਨਾਲ ਕੰਮ ਕਰ ਰਹੇ ਹਨ, ਨਿਯਮਤ ਅਧਾਰ ਤੇ ਹਾਰਡਵੇਅਰ ਦੀ ਜਾਂਚ ਕਰਦੇ ਰਹੋ। ਜੇਕਰ ਅਸੀਂ ਆਪਣੇ PC ਦੀ ਚੰਗੀ ਦੇਖਬਾਲ ਕਰ ਰਹੇ ਹਾਂ, ਤਾਂ ਇਹ ਕ੍ਰੈਸ਼ ਨਹੀਂ ਹੋਵੇਗਾ ਅਤੇ ਇਸ ਤਰ੍ਹਾਂ ਸਾਡਾ ਡਾਟਾ ਦਾ ਨੁਕਸਾਨ ਹੋਣ ਤੋਂ ਬਚੇਗਾ।

ਸਿਸਟਮ ਯੂਨਿਟ ਦੇ ਅੰਦਰੂਨੀ ਭਾਗਾਂ ਦੇ ਮੈਨਟੇਨੈਂਸ ਲਈ, ਸਿਸਟਮ ਯੂਨਿਟ ਦੇ ਕੇਸ ਨੂੰ ਸਮੇਂ-ਸਮੇਂ ਤੇ ਖੋਲ੍ਹੋ ਅਤੇ ਸਿਸਟਮ ਯੂਨਿਟ ਦੇ ਅੰਦਰੂਨੀ ਭਾਗਾਂ ਅਤੇ CPU ਪੱਖਿਆਂ ਦੀ ਕੰਡੀਸ਼ਨ ਦੀ ਜਾਂਚ ਕਰੋ। ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਪੂਰੀ ਤਰ੍ਹਾਂ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਕੰਮ ਕਰਨ ਵਾਲੇ ਵਾਤਾਵਰਣ ਤੇ ਨਿਰਭਰ ਕਰਦਾ ਹੈ। ਉਦਾਹਰਨ ਲਈ: ਜੇਕਰ ਵਾਤਾਵਰਨ ਬਹੁਤ ਜਿਆਦਾ ਧੂੜ ਭਰਿਆ ਹੈ, ਤਾਂ ਇੱਕ ਮਹੀਨੇ ਵਿੱਚ ਇੱਕ ਵਾਰ ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਕਰੋ, ਪਰ ਜੇਕਰ ਵਾਤਾਵਰਨ ਘੱਟ ਧੂੜ ਵਾਲਾ ਹੈ, ਤਾਂ ਸਾਲ ਵਿੱਚ ਇੱਕ ਵਾਰ ਮੈਨਟੇਨੈਂਸ ਕਰਨਾ ਕਾਫ਼ੀ ਹੋਵੇਗਾ। ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਜਾਂ ਰੱਖ-ਰਖਾਅ ਲਈ ਕੁਝ ਆਮ ਗਤੀਵਿਧੀਆਂ ਹਨ: ਸਫਾਈ, ਡਸਟਿੰਗ ਐਂਡ ਬਲੋਇੰਗ (blowing), ਅਤੇ ਵੱਖ-ਵੱਖ ਹਾਰਡਵੇਅਰ ਭਾਗਾਂ ਦੇ ਪੇਚ ਦੁਬਾਰਾ ਕਸਨਾ (re-screwing)।

ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਲਈ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼:

1. ਆਮ ਤੌਰ ਤੇ ਓਵਰਹੀਟਿੰਗ (overheating) ਸਮੱਸਿਆਵਾਂ ਦਾ ਇੱਕ ਵੱਡਾ ਕਾਰਨ ਧੂੜ ਹੁੰਦੀ ਹੈ। ਇਸ ਲਈ, ਇੱਕ ਬਲੋਅਰ, ਇੱਕ ਨਰਮ ਬੁਰਸ਼ ਜਾਂ ਇੱਕ ਵੈਕਿਊਮ ਕਲੀਨਰ ਦੀ ਵਰਤੋਂ ਹਾਰਡਵੇਅਰ ਦੇ ਹਿੱਸਿਆਂ ਤੋਂ ਧੂੜ ਨੂੰ ਹਟਾਉਣ ਲਈ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ। ਇਸ ਵਿਧੀ ਨੂੰ ਬਲੋਇੰਗ ਐਂਡ ਡਸਟਿੰਗ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
2. ਕੀਬੋਰਡ ਦੀ ਵਰਤੋਂ ਕਰਨ ਤੋਂ ਬਾਅਦ ਉਸਨੂੰ ਢੱਕ ਦੇਣਾ ਚਾਹੀਦਾ ਹੈ। ਕੀਬੋਰਡ ਨੂੰ ਉਸ ਅੰਦਰ ਜਾਣ ਵਾਲੇ ਧੂੜ ਕਣਾਂ ਤੋਂ ਬਚਾਉਣਾ ਚਾਹੀਂਦਾ ਹੈ।
3. ਮਾਊਸ ਦੀ ਸਤਹਿ ਤੋਂ ਗੰਦਗੀ ਨੂੰ ਸਾਫ ਕਰੋ। ਹਮੇਸ਼ਾ ਮਾਊਸ ਪੈਡ ਦੀ ਵਰਤੋਂ ਕਰੋ।
4. ਤਰਲ ਪਦਾਰਥਾਂ ਨੂੰ ਵੱਖ-ਵੱਖ ਹਾਰਡਵੇਅਰ ਭਾਗਾਂ ਤੋਂ, ਖਾਸ ਕਰਕੇ ਕੀਬੋਰਡ ਤੋਂ ਦੂਰ ਰੱਖਣਾ ਚਾਹੀਦਾ ਹੈ।
5. ਕੰਪਿਊਟਰ ਅਤੇ ਇਸਦੇ ਭਾਗਾਂ ਨੂੰ ਸਿੱਧੀ ਧੁੱਪ, ਗਰਮੀ ਦੇ ਸਰੋਤਾਂ, ਇਲੈਕਟ੍ਰੋਸਟੈਟਿਕ ਅਤੇ ਇਲੈਕਟ੍ਰੋ-ਮੈਗਨੈਟਿਕ ਮੀਡੀਆ ਤੋਂ ਦੂਰ ਰੱਖਣਾ ਚਾਹੀਦਾ ਹੈ।
6. ਕੰਪਿਊਟਰ ਨੂੰ ਭਰੋਸੇਮੰਦ ਸਟੈਬੀਲਾਈਜ਼ਰ ਨਾਲ ਜੋੜਿਆ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।
7. ਕੰਪਿਊਟਰ ਸਿਸਟਮਾਂ ਤੇ ਭਾਰੀ ਵਸਤੂਆਂ ਨਹੀਂ ਰੱਖਣੀਆਂ ਚਾਹੀਦੀਆਂ। ਖਾਸ ਕਰਕੇ ਲੈਪਟਾਪ ਉਪਰ ਭਾਰੀ ਵਸਤੂਆਂ ਰੱਖਣ ਨਾਲ ਇਸਦੀ ਸਕਰੀਨ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚ ਸਕਦਾ ਹੈ।
8. ਧੂੜ ਹਟਾਉਣ ਲਈ ਕੰਪਿਊਟਰ ਸਕ੍ਰੀਨ ਨੂੰ ਸੁੱਕੇ ਤੌਲੀਏ ਨਾਲ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਸਾਫ਼ ਕਰੋ। ਅਜਿਹਾ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਯਕੀਨੀ ਬਣਾਓ ਕਿ ਕੰਪਿਊਟਰ ਪੂਰੀ ਤਰ੍ਹਾਂ ਬੰਦ ਹੋਵੇ।
9. ਜਿੰਨ੍ਹਾ ਸੰਭਵ ਹੋ ਸਕੇ ਆਪਣੇ ਹੱਥਾਂ/ਉਂਗਲਾਂ ਨਾਲ ਕੰਪਿਊਟਰ ਸਕ੍ਰੀਨ ਨੂੰ ਛੂਹਣ ਤੋਂ ਬਚੋ।
10. ਪਾਵਰ ਕੇਬਲ ਚੰਗੀ ਕੁਆਲਿਟੀ ਦੀ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। ਸਾਨੂੰ ਸਮੇਂ-ਸਮੇਂ ਤੇ ਇਸ ਦੇ ਦੋਵੇਂ ਸਿਰੇ ਦੀ ਜਾਂਚ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ। ਕੇਬਲ ਕੁਨੈਕਸ਼ਨ ਢਿੱਲੇ ਨਹੀਂ ਹੋਣੇ ਚਾਹੀਦੇ।

ਹਮੇਸ਼ਾ ਸਿਸਟਮ ਨੂੰ ਸਾਫ਼ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਸਿਸਟਮ ਨੂੰ ਬੰਦ ਅਤੇ ਅਨਪਲੱਗ ਕਰੋ। ਕਿਸੇ ਵੀ ਹਾਰਡਵੇਅਰ ਤੇ ਕਦੇ ਵੀ ਕੋਈ ਤਰਲ ਪਦਾਰਥ ਜਿਵੇਂ ਪਾਣੀ ਜਾਂ ਕਲੀਨਰ ਨਾ ਲਗਾਓ। ਪਹਿਲਾਂ ਨਰਮ ਕੱਪੜੇ ਤੇ ਕਲੀਨਰ ਦਾ ਛਿੜਕਾਓ ਕਰੋ ਅਤੇ ਫਿਰ ਕੱਪੜੇ ਨਾਲ PC ਨੂੰ ਸਾਫ਼ ਕਰੋ।

6.1.2 ਸਾਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ:

ਸਾਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਵਿੱਚ ਉਹ ਗਤੀਵਿਧੀਆਂ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ, ਜੋ ਕੰਪਿਊਟਰ ਨੂੰ ਚੰਗੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨਾਲ ਚਲਾਉਂਦੀਆਂ ਹਨ ਅਤੇ ਸਿਸਟਮ ਨੂੰ ਸੁਚਾਰੂ ਢੰਗ ਨਾਲ ਚਲਾਉਣ ਵਿਚ ਮਦਦ ਕਰਦੀਆਂ ਹਨ। ਇਸ ਕਿਸਮ ਦਾ ਰੱਖ-ਰਖਾਅ ਕਈ ਕਾਰਨਾਂ ਕਰਕੇ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਜਿਸ ਵਿੱਚ ਕੰਪਿਊਟਰ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਵਧਾਉਣਾ, ਮੁੱਦਿਆਂ (issues) ਨੂੰ ਠੀਕ ਕਰਨਾ ਅਤੇ ਹੋਰ ਬਹੁਤ ਕੁਝ ਸ਼ਾਮਲ ਹੈ। ਸਾਨੂੰ ਸਾਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਕੁੱਝ ਬੁਨਿਆਦੀ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼ਾਂ ਦੀ ਪਾਲਣਾ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ:

ਸਾਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਲਈ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼:

1. ਆਪਣੇ ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ ਨੂੰ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਸਕੈਨ ਕਰਕੇ ਵਾਇਰਸਾਂ ਅਤੇ ਵਾਰਮਸ (worms) ਤੋਂ ਮੁਕਤ ਰੱਖੋ।
2. ਗੈਰ-ਭਰੋਸੇਯੋਗ ਅਤੇ ਅਣਜਾਣ, ਨਿੱਜੀ ਜਾਂ ਜਨਤਕ ਨੈੱਟਵਰਕਾਂ ਨਾਲ ਜੁੜਨ ਤੋਂ ਪਰਹੇਜ਼ ਕਰੋ।
3. ਗੈਰ-ਭਰੋਸੇਯੋਗ ਅਤੇ ਅਣਜਾਣ ਸਰੋਤਾਂ ਤੋਂ ਪ੍ਰੋਗਰਾਮਾਂ/ਐਪਲੀਕੇਸ਼ਨਾਂ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਤੋਂ ਬਚੋ।
4. ਡਿਵਾਈਸ ਡਰਾਈਵਰਾਂ ਅਤੇ ਸਾਫਟਵੇਅਰਾਂ ਨੂੰ ਅੱਪ-ਟੂ-ਡੇਟ ਰੱਖੋ।
5. ਸਾਡੇ ਕੰਪਿਊਟਰਾਂ ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਮਹੱਤਵਪੂਰਨ ਡਾਟਾ ਦਾ ਬੈਕਅੱਪ ਹਮੇਸ਼ਾ ਬਾਹਰੀ ਡਿਸਕਾਂ ਤੇ ਰੱਖੋ।
6. ਡਾਟੇ ਦੇ ਨੁਕਸਾਨ ਨੂੰ ਰੋਕਣ ਲਈ, ਕੰਪਿਊਟਰ ਤੋਂ USB ਡਿਸਕਾਂ ਨੂੰ ਅਚਾਨਕ ਅਨ-ਪਲੱਗ ਕਰਨ ਤੋਂ ਬਚੋ। ਸਾਨੂੰ USB ਡਿਵਾਈਸਾਂ ਨੂੰ ਅਨਪਲੱਗ ਕਰਨ ਲਈ Safely Remove ਆਪਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਚਾਹਦੀ ਹੈ।
7. ਇਹ ਯਕੀਨੀ ਬਣਾਓ ਕਿ ਸਾਡੇ ਕੰਪਿਊਟਰ ਦੀ ਸਿਕਿਓਰਟੀ ਨਿਯਮਿਤ ਰੂਪ ਵਿਚ ਅਪਡੇਟ ਹੁੰਦੀ ਰਹੇ।
8. ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਕੰਮ ਦੀ ਰਫਤਾਰ ਮੈਨਟੇਨ ਰੱਖਣ ਲਈ ਟੈਂਪਰੇਰੀ ਇੰਟਰਨੈਟ ਫਾਈਲਾਂ ਨੂੰ ਹਟਾਉਂਦੇ ਰਹੋ।
9. ਅਸੀਂ ਸਿਸਟਮ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਬਿਹਤਰ ਬਣਾਉਣ ਲਈ, ਹਾਰਡ ਡਰਾਈਵ ਉਪਰ ਵਾਧੂ ਖਾਲੀ ਜਗ੍ਹਾ ਬਣਾਈ ਰੱਖਣ ਲਈ ਡਿਸਕ ਕਲੀਨ-ਅੱਪ ਆਪਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।
10. ਅਸੀਂ ਆਪਣੀ ਡਿਵਾਈਸ ਤੋਂ ਬੇਲੋੜੀਆਂ ਜਾਂ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨੂੰ ਮਿਟਾਉਣ ਲਈ ਸਟੋਰੇਜ ਸੈਂਸ (Storage Sense) ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ, ਜਿਸ ਨਾਸ ਸਟੋਰੇਜ ਸਪੇਸ ਖਾਲੀ ਹੋ ਸਕਦੀ ਹੈ।

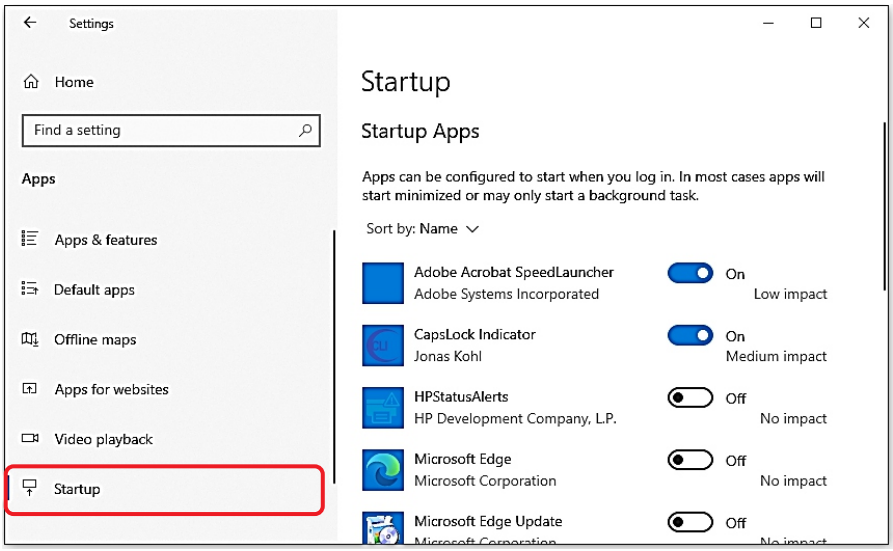
6.2 ਕੰਪਿਊਟਰ ਦੇ ਪ੍ਰਦਰਸ਼ਨ ਵਿੱਚ ਸੁਧਾਰ ਲਈ ਟੂਲਜ਼ ਅਤੇ ਤਕਨੀਕਾਂ

ਕੰਪਿਊਟਰ ਤੋਂ ਸਰਵੋਤਮ ਢੰਗ ਨਾਲ ਕੰਮ ਕਰਵਾਉਣ ਲਈ ਸਾਨੂੰ ਨਿਮਨਲਿਖਤ ਟੂਲਜ਼ ਅਤੇ ਤਕਨੀਕਾਂ ਦੀ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਵਰਤੋਂ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ:

6.2.1 ਅਨਨਸੈਸਰੀ ਸਟਾਰਟ-ਅੱਪ ਪ੍ਰੋਗਰਾਮਜ਼ ਨੂੰ ਡਿਸੇਬਲ ਕਰੋ (Disabe Unnecessary Start-up of Programs):

ਜਦੋਂ ਅਸੀਂ ਆਪਣੇ PC ਨੂੰ ਸਟਾਰਟ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਕੁਝ ਪ੍ਰੋਗਰਾਮ ਆਪਣੇ ਆਪ ਸੁਰੂ ਹੋ ਜਾਂਦੇ ਹਨ ਅਤੇ ਉਹ ਬੈਕਗ੍ਰਾਊਂਡ ਵਿੱਚ ਚੱਲਦੇ ਹਨ। ਜਦੋਂ ਅਸੀਂ ਉਹਨਾਂ ਨੂੰ ਵਰਤਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਜਿਹੇ ਪ੍ਰੋਗਰਾਮ ਜਲਦੀ ਖੁੱਲ੍ਹਣਗੇ। ਇਹ ਉਹਨਾਂ ਪ੍ਰੋਗਰਾਮਾਂ ਲਈ ਮਦਦਗਾਰ ਹੈ ਜੋ ਅਸੀਂ ਬਹੁਤ ਵਰਤਦੇ ਹਾਂ, ਪਰ ਉਹਨਾਂ ਪ੍ਰੋਗਰਾਮਾਂ ਲਈ ਨਹੀਂ ਜੋ ਅਸੀਂ ਅਕਸਰ ਨਹੀਂ ਵਰਤਦੇ। ਅਜਿਹੇ ਪ੍ਰੋਗਰਾਮਾਂ ਦਾ ਆਟੋਮੈਟਿਕ ਸਟਾਰਟ-ਅੱਪ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੇ ਬੂਟਿੰਗ ਸਮੇਂ ਨੂੰ ਵਧਾਉਂਦਾ ਹੈ। ਅਸੀਂ ਗੈਰ-ਜ਼ਰੂਰੀ ਪ੍ਰੋਗਰਾਮਾਂ ਦੇ ਆਟੋਮੈਟਿਕ ਸਟਾਰਟ-ਅੱਪ ਨੂੰ ਡਿਸੇਬਲ ਕਰ ਸਕਦੇ ਹਾਂ ਤਾਂ ਕਿ ਜਦੋਂ ਸਾਡਾ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਸਟਾਰਟ ਹੋਵੇ, ਤਾਂ ਉਹ ਪ੍ਰੋਗਰਾਮ ਆਪਣੇ ਆਪ ਨਾ ਚੱਲਣ। ਕਿਸੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਆਪਣੇ ਆਪ ਸ਼ੁਰੂ ਹੋਣ ਤੋਂ ਰੋਕਣ ਲਈ (ਵਿੰਡੋਜ਼ 10 ਜਾਂ ਬਾਅਦ ਦੇ ਵਰਜਨ ਵਿੱਚ), ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਵਰਤੋਂ ਕਰੋ:

1. **Start** ਬਟਨ ਉੱਤੇ ਕਲਿੱਕ ਕਰੋ ਫਿਰ **Setting –> Apps –> Startup** ਸਿਲੈਕਟ ਕਰੋ।
2. **Startup Apps** ਖੇਤਰ ਵਿੱਚ ਉਹ ਪ੍ਰੋਗਰਾਮ ਲੱਭੋ ਜਿਸ ਨੂੰ ਅਸੀਂ ਆਪਣੇ ਆਪ ਸ਼ੁਰੂ ਹੋਣ ਤੋਂ ਰੋਕਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਅਤੇ ਇਸਨੂੰ Off ਤੇ ਸੈੱਟ ਕਰੋ।



ਚਿੱਤਰ 6.2: ਇਨੇਬਲ/ਡਿਸੇਬਲ ਸਟਾਰਟਅਪ ਐਪਲੀਕੇਸ਼ਨ

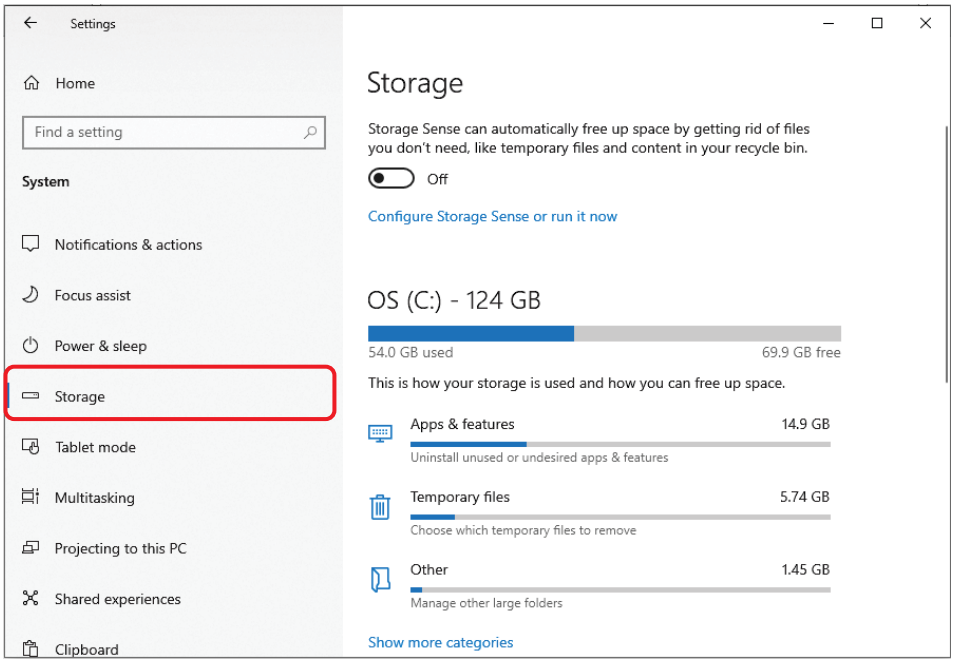
ਜੇਕਰ ਤੁਸੀਂ ਕਿਸੇ ਪ੍ਰੋਗਰਾਮ ਦੇ ਸਟਾਰਟ-ਅੱਪ ਨੂੰ ਬੰਦ ਕਰ ਦਿੰਦੇ ਹੋ ਅਤੇ ਵਿੰਡੋਜ਼ ਸ਼ੁਰੂ ਹੋਣ ਉਪਰੰਤ ਉਹ ਪ੍ਰੋਗਰਾਮ ਫਿਰ ਵੀ ਆਪਣੇ ਆਪ ਚਾਲੂ ਹੋ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਤੁਹਾਨੂੰ ਆਪਣੇ ਸਿਸਟਮ ਨੂੰ ਵਾਇਰਸ ਅਤੇ ਮਾਲਵੇਅਰ ਸਾਫਟਵੇਅਰਾਂ ਨਾਲ ਸਕੈਨ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ।

6.2.2 ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨੂੰ ਡਿਲੀਟ ਕਰਨਾ (Delete Temporary Files):

ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਉਹ ਫਾਈਲਾਂ ਹੁੰਦੀਆਂ ਹਨ ਜੋ ਟੈਂਪਰੇਰੀ ਤੌਰ ਤੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਬਣਾਈਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਅਜਿਹੀਆਂ ਫਾਈਲਾਂ ਪ੍ਰੋਗਰਾਮਾਂ ਦੁਆਰਾ ਬਣਾਈਆਂ ਜਾਂਦੀਆਂ ਹਨ, ਜਿਵੇਂ ਕਿ ਵਰਡ-ਪ੍ਰੋਸੈਸਰ, ਸਪ੍ਰੈਡਸ਼ੀਟ ਆਦਿ ਦੀ ਐਗਜੀਕਿਊਸ਼ਨ ਦੌਰਾਨ ਇਹ ਫਾਈਲਾਂ ਬਣਦੀਆਂ ਹਨ। ਇੱਥੋਂ ਤੱਕ ਕਿ ਜਦੋਂ ਅਸੀਂ ਵੈਬਸਾਈਟਾਂ ਨੂੰ ਐਕਸੈਸ ਕਰਨ ਲਈ ਵੈਬ ਬ੍ਰਾਊਜ਼ਰ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਵੈਬਸਾਈਟਾਂ ਦੁਆਰਾ ਕਈ ਤਰ੍ਹਾਂ ਦੀਆਂ ਜਾਣਕਾਰੀਆਂ ਨੂੰ ਕੈਸ਼ ਕਰਨ ਲਈ ਵੀ ਕਈ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਬਣਾਈਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਪ੍ਰੋਗਰਾਮ ਦੇ ਬੰਦ ਹੋਣ ਤੋਂ ਬਾਅਦ ਇਹਨਾਂ ਫਾਈਲਾਂ ਦੀ ਲੋੜ ਨਹੀਂ ਪੈਂਦੀ। ਅਜਿਹੀਆਂ ਫਾਈਲਾਂ ਸਮੇਂ ਦੇ ਨਾਲ ਨਾਲ ਇਕੱਠੀਆਂ ਹੁੰਦੀਆਂ ਰਹਿੰਦੀਆਂ ਹਨ। ਕੰਪਿਊਟਰ ਤੇ ਬਹੁਤ ਸਾਰੀਆਂ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਇਕੱਠੀਆਂ ਹੋਣ ਨਾਲ ਸਿਸਟਮ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਹੌਲੀ ਹੋ ਸਕਦੀ ਹੈ ਅਤੇ ਇਹ ਕੀਮਤੀ ਸਟੋਰੇਜ ਸਪੇਸ ਵੀ ਘੇਰਦੀਆਂ ਹਨ। ਅਸੀਂ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਬਿਹਤਰ ਬਣਾਉਣ ਲਈ ਅਜਿਹੀਆਂ ਫਾਈਲਾਂ ਨੂੰ ਡਿਲੀਟ ਕਰ ਸਕਦੇ ਹਾਂ। ਕੰਪਿਊਟਰ ਸਿਸਟਮ (Windows 10 ਜਾਂ ਬਾਅਦ ਦੇ ਵਰਜਨ) ਤੋਂ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨੂੰ ਮਿਟਾਉਣ ਲਈ, ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਵਰਤੋਂ ਕਰੋ:

1. **Settings** ਓਪਨ ਕਰੋ, ਫਿਰ **System –> Storage** ਸਿਲੈਕਟ ਕਰੋ।
2. ਸਟੋਰੇਜ ਬ੍ਰੇਕਡਾਊਨ ਵਿੱਚ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨੂੰ ਸਲੈਕਟ ਕਰੋ।

ਨੋਟ: ਜੇਕਰ ਲਿਸਟ ਵਿੱਚ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨਹੀਂ ਦਿਖਾਈ ਦੇ ਰਹੀਆਂ ਤਾਂ show more categories ਤੇ ਕਲਿੱਕ ਕਰੋ।



ਚਿੱਤਰ 6.3: ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨੂੰ ਡਿਲੀਟ ਕਰਨਾ

3. ਵਿੰਡੋ ਇਹ ਨਿਰਧਾਰਤ ਕਰਨ ਵਿੱਚ ਕੁਝ ਸਮਾਂ ਲਵੇਗੀ ਕਿ ਕਿਹੜੀਆਂ ਫਾਈਲਾਂ ਅਤੇ ਐਪਸ ਸਾਡੇ PC ਤੇ ਸਭ ਤੋਂ ਵੱਧ ਜਗ੍ਹਾ ਲੈ ਰਹੇ ਹਨ।

4. ਉਹ ਆਈਟਮਾਂ ਸਲੈਕਟ ਕਰੋ ਜੋ ਅਸੀਂ ਮਿਟਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਅਤੇ ਫਿਰ **Remove files** ਸਿਲੈਕਟ ਕਰੋ।

6.2.3 ਉਹਨਾਂ ਐਪਸ ਨੂੰ ਅਣਇੰਸਟਾਲ ਕਰੋ ਜੋ ਅਸੀਂ ਹੁਣ ਨਹੀਂ ਵਰਤਦੇ (Uninstall Apps we don't use anymore):

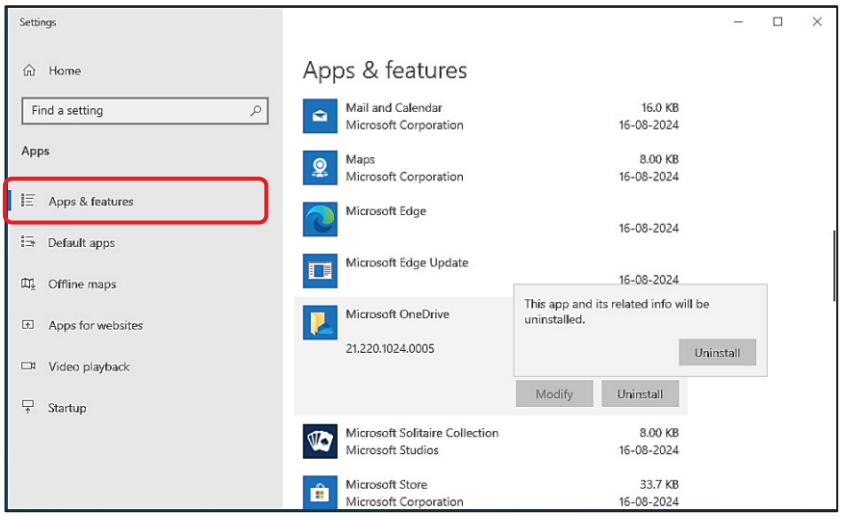
ਸਾਨੂੰ ਉਹਨਾਂ ਐਪਸ ਨੂੰ ਅਣਇੰਸਟਾਲ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ ਜੋ ਅਸੀਂ ਹੁਣ ਨਹੀਂ ਵਰਤਦੇ, ਕਿਉਂਕਿ ਇਹ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਪ੍ਰਭਾਵਿਤ ਕਰਦੇ ਹਨ। ਅਜਿਹੀਆਂ ਐਪਸ ਨੂੰ ਅਣਇੰਸਟਾਲ ਕਰਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਵਰਤੋਂ ਕਰੋ:

1. **Start –> Settings –> Apps –> Apps and Features** ਸਿਲੈਕਟ ਕਰੋ। ਇਹ ਕੰਪਿਊਟਰ ਵਿੱਚ ਇੰਸਟਾਲਡ ਸਾਰੇ ਐਪਸ ਦੀ ਸੂਚੀ ਦਿਖਾਏਗਾ।

2. ਜਿਸ ਖਾਸ ਐਪ ਨੂੰ ਅਸੀਂ ਅਨਇੰਸਟਾਲ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਉਸਨੂੰ ਐਪਸ ਦੀ ਲਿਸਟ ਵਿਚੋਂ ਲੱਭੋ। ਅਸੀਂ ਵੱਧ ਸਪੇਸ ਦੀ ਵਰਤੋਂ ਕਰਨ ਵਾਲੀਆਂ ਐਪਸ ਨੂੰ ਕ੍ਰਮਬੱਧ ਕਰਕੇ ਵੀ ਦੇਖ ਸਕਦੇ ਹਾਂ।

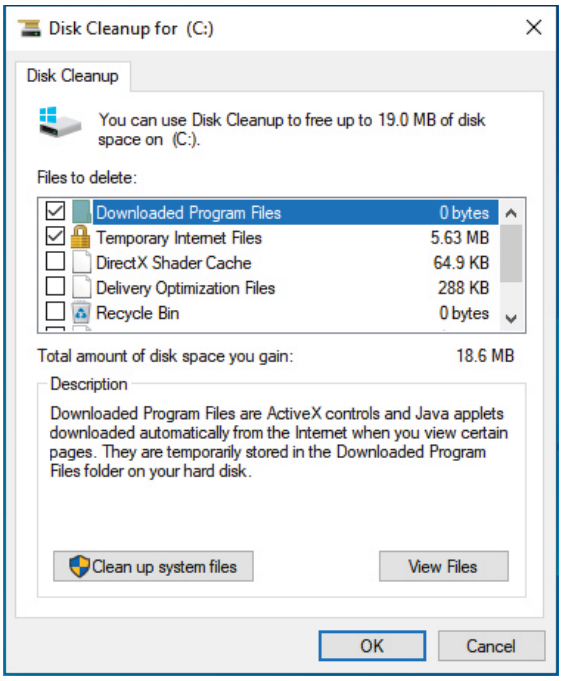
3. ਜਦੋਂ ਸਾਨੂੰ ਲਿਸਟ ਵਿਚੋਂ ਉਹ ਐਪ ਮਿਲ ਜਾਂਦੀ ਹੈ ਜਿਸ ਨੂੰ ਅਸੀਂ ਅਨਇੰਸਟਾਲ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਉਸ ਤੇ ਕਲਿੱਕ ਕਰੋ ਅਤੇ ਫਿਰ **Uninstall** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।

4. ਹੁਣ ਇਹ ਇਕ ਛੋਟੇ ਬਾਕਸ ਵਿਚ ਇਕ ਸੰਦੇਸ਼ ਦਿਖਾਵੇਗਾ ਕਿ "ਇਹ ਐਪ ਅਤੇ ਇਸ ਨਾਲ ਸਬੰਧਤ ਜਾਣਕਾਰੀ ਨੂੰ ਅਣਇੰਸਟਾਲ ਕਰ ਦਿੱਤਾ ਜਾਵੇਗਾ"। ਹੁਣ ਕੰਪਿਊਟਰ ਵਿਚੋਂ ਐਪ ਨੂੰ ਹਟਾਉਣ ਲਈ **Uninstall** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।



ਚਿੱਤਰ 6.4: ਐਪਸ ਨੂੰ ਅਣਇੰਸਟਾਲ ਕਰਨਾ

6.2.4 ਡਿਸਕ ਕਲੀਨ-ਅੱਪ ਨੂੰ ਰਨ ਕਰੋ (Run Disk Cleanup): ਡਿਸਕ ਕਲੀਨ-ਅੱਪ ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ ਯੂਟੀਲਿਟੀ ਹੈ ਜੋ ਮਾਈਕ੍ਰੋਸਾਫਟ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿਚ ਪਹਿਲਾਂ ਤੋਂ ਹੀ ਮੌਜੂਦ ਹੁੰਦੀ ਹੈ। ਇਹ ਯੂਟੀਲਿਟੀ ਹਾਰਡ ਡਰਾਈਵ ਤੋਂ ਜਗ੍ਹਾ ਖਾਲੀ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਕਲੀਨ-ਅੱਪ ਪ੍ਰਕਿਰਿਆ ਵਿੱਚ ਹਾਰਡ ਡਰਾਈਵ ਉਪਰ ਉਹਨਾਂ ਫਈਲਾਂ ਨੂੰ ਸਰਚ ਅਤੇ ਐਨਾਲਾਈਜ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਨ੍ਹਾਂ ਦੀ ਹੁਣ ਲੋੜ ਨਹੀਂ ਹੈ। ਫਿਰ ਇਹ ਉਹਨਾਂ ਫਾਈਲਾਂ ਨੂੰ ਹਟਾਉਣ ਲਈ ਅੱਗੇ ਵਧਦਾ ਹੈ ਅਤੇ ਇਸ ਤਰ੍ਹਾਂ ਹਾਰਡ ਡਰਾਈਵ ਤੇ ਡਿਸਕ ਸਪੇਸ ਖਾਲੀ ਕਰਦਾ ਹੈ। ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਡਿਸਕ ਕਲੀਨ-ਅੱਪ ਯੂਟੀਲਿਟੀ ਨੂੰ ਚਲਾਉਣ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਕਦਮਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ:



ਚਿੱਤਰ 6.5: ਡਿਸਕ ਕਲੀਨ-ਅੱਪ

1. Start ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ ਅਤੇ Disk Cleanup ਲਈ ਸਰਚ ਕਰੋ, ਫਿਰ ਸੂਚੀ ਵਿੱਚੋਂ **Disk Cleanup** ਸਿਲੈਕਟ ਕਰੋ।
2. ਉਸ ਡਰਾਈਵ ਦੀ ਚੋਣ ਕਰੋ ਜਿਸ ਨੂੰ ਅਸੀਂ ਕਲੀਨ-ਅੱਪ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਅਤੇ ਫਿਰ OK ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।
3. ਡਿਸਕ ਕਲੀਨਅਪ ਡਾਇਲਾਗ ਬਾਕਸ ਵਿੱਚ ਉਹਨਾਂ ਫਾਈਲਾਂ ਦੀਆਂ ਕਿਸਮਾਂ ਵਾਲੇ ਚੈਕਬਾਕਸ ਸਿਲੈਕਟ ਕਰੋ ਜਿਹਨਾਂ ਨੂੰ ਅਸੀਂ ਡਿਲੀਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।
4. **OK** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ। ਹੁਣ cleanup ਸ਼ੁਰੂ ਕਰਨ ਲਈ ਕਨਫਰਮੇਸ਼ਨ ਵਿੰਡੋ ਵਿੱਚ **Delete files** ਦੀ ਚੋਣ ਕਰੋ।
5. ਹੋਰ ਸਪੇਸ ਖਾਲੀ ਕਰਨ ਲਈ, "**Clean up system files**" ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ। ਜੇਕਰ ਹੋਰ ਸਪੇਸ ਖਾਲੀ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ, ਤਾਂ ਸਪੇਸ ਦੀ ਮਾਤਰਾ ਨੂੰ ਕੈਲਕੂਲੇਟ ਕਰਨ ਲਈ ਸਿਸਟਮ ਕੁੱਝ ਸਮਾਂ ਲਵੇਗਾ, ਡਿਲੀਟ ਕਰਨ ਲਈ ਫਾਈਲਾਂ ਦੀ ਕਿਸਮ ਚੁਣੋ ਅਤੇ OK ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।

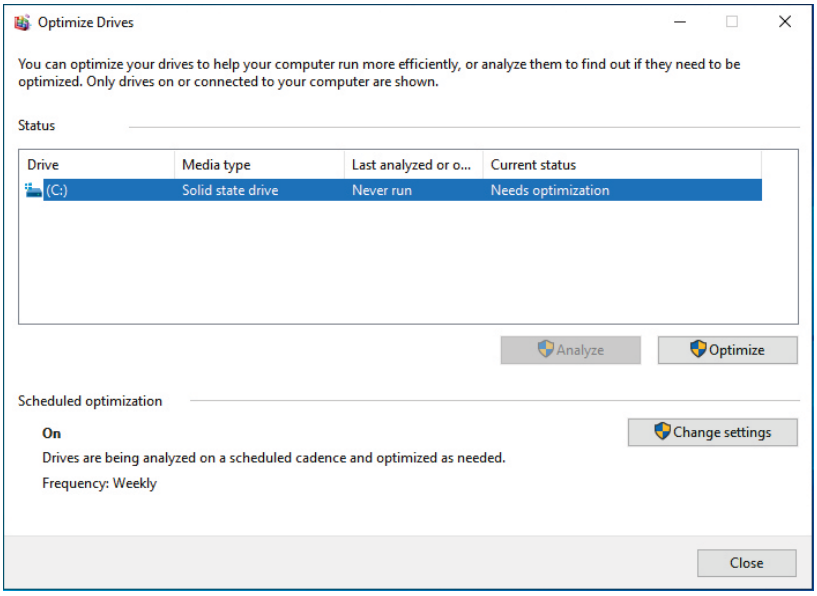
ਉਹ ਸਾਫਟਵੇਅਰ ਜੋ ਯੂਜਰ ਅਤੇ ਡਿਵੈਲਪਰਾਂ ਲਈ ਸਹਾਇਕ ਭੂਮਿਕਾ ਨਿਭਾਉਂਦੇ ਹਨ, ਉਹਨਾਂ ਨੂੰ **ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਬੁਨਿਆਦੀ ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮਾਂ ਵਿੱਚ ਫਾਈਲ/ਫੋਲਡਰ ਮੈਨੇਜਮੈਂਟ (ਕਾਪੀ, ਮੂਵ, ਆਦਿ), ਡਿਸਕ ਫਾਰਮੈਟ ਅਤੇ ਪਾਰਟੀਸ਼ਨ ਮੈਨੇਜ਼ਰ, ਡਿਸਕ ਕਲੀਨਅੱਪ, ਡਿਸਕ ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ, ਬੈਕ-ਅੱਪ ਅਤੇ ਰੀਸਟੋਰ, ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ ਆਦਿ ਟੂਲਜ਼ ਸ਼ਾਮਲ ਹਨ।

6.2.5 ਡਿਸਕ ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ (Disk Defragmentation):

ਇਹ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੀ ਇੱਕ ਹੋਰ ਮਹੱਤਵਪੂਰਨ ਇਨਬਿਲਟ ਯੂਟੀਲਿਟੀ ਹੈ। ਕਿਉਂਕਿ ਕੰਪਿਊਟਰ ਉਪਰ ਫਾਈਲਾਂ ਨੂੰ ਯੂਜ਼ਰ ਦੁਆਰਾ ਲਗਾਤਾਰ ਲਿਖਿਆ (written), ਮਿਟਾਇਆ (deleted) ਅਤੇ ਮੁੜ-ਆਕਾਰ (resized) ਦਿੱਤਾ ਜਾਂਦਾ ਰਹਿੰਦਾ ਹੈ, ਇਸਲਈ ਡਿਸਕ ਉਪਰ ਫਾਈਲਾਂ ਦੀ ਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ (ਵਿਭਾਜਨ) ਇੱਕ ਕੁਦਰਤੀ ਪ੍ਰੋਸੈਸ ਹੈ। ਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਕਾਰਨ ਇੱਕ ਫਾਈਲ, ਮੈਮੋਰੀ ਦੇ ਕਈ ਵੱਖੋ-ਵੱਖਰੇ ਖੇਤਰਾਂ ਵਿੱਚ ਸਟੋਰ ਹੁੰਦੀ ਰਹਿੰਦੀ ਹੈ ਜੋ ਕਿ ਹਾਰਡ ਡਿਸਕ ਵਿੱਚ ਖਿੰਡੇ ਰਹਿੰਦੇ ਹਨ। ਜਦੋਂ ਕੋਈ ਫਾਈਲ ਕਈ ਥਾਵਾਂ 'ਤੇ ਖਿੰਡ ਜਾਂਦੀ ਹੈ, ਤਾਂ ਕੰਪਿਊਟਰ ਨੂੰ ਉਸ ਫਾਈਲ ਨੂੰ ਪੜ੍ਹਨ ਅਤੇ ਲਿਖਣ ਲਈ ਜ਼ਿਆਦਾ ਸਮਾਂ ਲੱਗਦਾ ਹੈ। ਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਦੇ ਪ੍ਰਭਾਵ ਬਹੁਤ ਜ਼ਿਆਦਾ ਵਿਆਪਕ ਹੁੰਦੇ ਹਨ: ਇਹ ਕੰਪਿਊਟਰ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਹੌਲੀ ਕਰ ਦਿੰਦਾ ਹੈ, ਬੂਟਿੰਗ-ਟਾਈਮ ਲੰਬਾ ਹੋ ਜਾਂਦਾ ਹੈ, ਅਤੇ ਕੰਪਿਊਟਰ ਰੈਂਡਮਲੀ ਕਰੈਸ਼ ਹੋਣ ਲੱਗ ਜਾਂਦਾ ਹੈ।

ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਸਾਡੀ ਹਾਰਡ ਡਰਾਈਵ ਵਿੱਚ ਡਾਟਾ ਦੇ ਸਾਰੇ ਖਿੰਡੇ ਹੋਏ ਟੁਕੜਿਆਂ ਨੂੰ ਲੱਭ ਕੇ ਉਹਨਾਂ ਨੂੰ ਦੁਬਾਰਾ ਇਕੱਠੇ ਕਰਕੇ ਰੱਖਦਾ ਹੈ, ਅਤੇ ਇਸ ਤਰ੍ਹਾਂ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਵਿੱਚ ਸੁਧਾਰ ਕਰਦਾ ਹੈ। ਵਿੰਡੋਜ਼ OS ਵਿੱਚ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਦੀ ਵਰਤੋਂ ਕਰੋ:

- **Start** ਮੀਨੂੰ ਖੋਲ੍ਹੋ।
- '**Defragment**' ਟਾਈਪ ਕਰੋ ਅਤੇ '**Defragment and Optimize Drives**' ਨਾਮਕ ਨਤੀਜਾ ਦਿਖਾਈ ਦੇਵੇਗਾ।
- **Optimize Drives** ਵਿੰਡੋ ਨੂੰ ਖੋਲ੍ਹਣ ਲਈ ਇਸ ਤੇ ਕਲਿੱਕ ਕਰੋ।

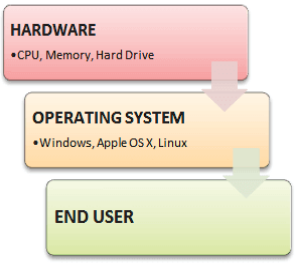


ਚਿੱਤਰ 6.6: ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਅਤੇ ਆਪਟੀਮਾਈਜ ਡਰਾਈਵਸ

- ਇਸ ਵਿੰਡੋ ਵਿੱਚ ਉਹ ਡਿਸਕ ਚੁਣੋ ਜਿਸਨੂੰ ਅਸੀਂ ਡੀਫ੍ਰੈਗਮੈਂਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।
- ਹੁਣ, ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ ਸ਼ੁਰੂ ਕਰਨ ਲਈ **Optimize** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।

6.3 ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਅਤੇ ਸੇਫ ਮੋਡ ਵਿੱਚ ਬੂਟਿੰਗ

ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ (OS) ਇੱਕ ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ ਹੁੰਦਾ ਹੈ ਜੋ ਯੂਜਰ ਅਤੇ ਕੰਪਿਊਟਰ ਹਾਰਡਵੇਅਰ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਵਜੋਂ ਕੰਮ ਕਰਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਚਲਾਉਣ ਲਈ ਹਰੇਕ ਕੰਪਿਊਟਰ ਵਿੱਚ ਘੱਟੋ-ਘੱਟ ਇੱਕ OS ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਕ੍ਰੋਮ (Chrome), ਐਮਐਸ ਵਰਡ (MS Word), ਗੇਮਜ਼ (Games), ਆਦਿ ਵਰਗੀਆਂ ਐਪਲੀਕੇਸ਼ਨਾਂ ਨੂੰ ਅਜਿਹੇ ਮਾਹੌਲ (environment) ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ ਜਿਸ ਵਿਚ ਉਹ ਚੱਲ ਸਕਣ ਅਤੇ ਆਪਣੇ ਕੰਮ ਕਰ ਸਕਣ। ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਤੋਂ ਬਿਨਾਂ ਯੂਜਰ ਲਈ ਕਿਸੇ ਵੀ ਕੰਪਿਊਟਰ ਜਾਂ ਮੋਬਾਈਲ ਡਿਵਾਈਸ ਨੂੰ ਵਰਤਣਾ ਸੰਭਵ ਨਹੀਂ ਹੈ।



ਚਿੱਤਰ 6.7 ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ

ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੁਨੀਆ ਜਾ ਸਭ ਤੋਂ ਪ੍ਰਸਿੱਧ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਹੈ। ਇਹ ਮਾਈਕ੍ਰੋਸਾਫਟ ਕਾਰਪੋਰੇਸ਼ਨ ਦੁਆਰਾ 1985 ਵਿੱਚ ਵਿਕਸਤ ਕੀਤਾ ਗਿਆ ਸੀ। ਇਸਲਈ, ਇਸਨੂੰ ਮਾਈਕ੍ਰੋਸੌਫਟ ਵਿੰਡੋਜ਼ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਮਲਟੀਟਾਸਕਿੰਗ ਅਤੇ ਹੋਰ ਬਹੁਤ ਸਾਰੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਦੇ ਨਾਲ ਇੱਕ ਬਹੁਤ ਵਧੀਆ ਅਤੇ ਇੰਟਰਐਕਟਿਵ ਗ੍ਰਾਫਿਕਲ ਯੂਜ਼ਰ ਇੰਟਰਫੇਸ (Graphical User Interface (GUI)) ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਮਾਈਕਰੋਸਾਫਟ ਵਿੰਡੋਜ਼ ਨੇ 1985 ਵਿੱਚ ਆਪਣੀ ਪਹਿਲੀ ਰੀਲੀਜ਼ ਤੋਂ ਬਾਅਦ ਬਹੁਤ ਸਾਰੇ ਪ੍ਰਮੁੱਖ ਵਰਜਨਾਂ ਨੂੰ ਲਾਂਚ ਕੀਤਾ। ਹੇਠਾਂ ਦਿੱਤੀ ਤਸਵੀਰ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੇ ਮੁੱਖ ਵਰਜਨਾਂ (ਇਸਦੀ ਪਹਿਲੀ ਰਿਲੀਜ਼ ਤੋਂ ਹੁਣ ਤੱਕ) ਉਹਨਾਂ ਦੇ ਨਾਮ ਅਤੇ ਜਾਰੀ ਕਰਨ ਦੇ ਸਾਲ ਦੇ ਨਾਲ ਦਿਖਾਉਂਦੀ ਹੈ।

ਚਿੱਤਰ 6.8: ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੇ ਮੁੱਖ ਵਰਜਨ

ਵਿੰਡੋਜ ਦਾ ਮੌਜੂਦਾ ਵਰਜ਼ਨ ਇਸ ਦੇ ਪੁਰਾਣੇ ਵਰਜ਼ਨ ਨਾਲੋਂ ਬਹੁਤ ਵੱਖਰਾ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ। ਵਿੰਡੋਜ਼ 10 ਅਤੇ ਵਿੰਡੋ 11 ਵਰਗੇ ਨਵੇਂ ਵਿੰਡੋ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਪੁਰਾਣੇ ਵਰਜ਼ਨ ਨਾਲੋਂ ਵਧੇਰੀ ਕੰਪਿਊਟਿੰਗ ਪਾਵਰ ਹੈ। ਇਹ ਵਿੰਡੋਜ OS ਟੱਚ ਅਤੇ ਮਾਊਸ ਆਧਾਰਿਤ ਡਿਵਾਈਸਾਂ ਦੇ ਅਨੁਕੂਲ ਵੀ ਹਨ।

6.3.1 ਬੂਟਿੰਗ ਅਤੇ ਸੇਫ ਮੋਡ (Booting and Safe Mode):

ਜਦੋਂ ਅਸੀਂ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦਾ ਪਾਵਰ ਬਟਨ ਦਬਾਉਂਦੇ ਹਾਂ, ਤਾਂ ਇਸਦਾ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਮੇਨ ਮੈਮੋਰੀ ਵਿੱਚ ਲੋਡ ਹੋਣਾ ਸ਼ੁਰੂ ਹੋ ਜਾਂਦਾ ਹੈ, ਇਸ ਪ੍ਰਕਿਰਿਆ ਨੂੰ **ਬੂਟਿੰਗ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਜੇਕਰ ਸਾਡਾ ਕੰਪਿਊਟਰ ਸਹੀ ਤਰੀਕੇ ਨਾਲ ਬੂਟ ਨਹੀਂ ਹੋ ਰਿਹਾ, ਤਾਂ ਇਸ ਦਾ ਕਾਰਨ ਹਾਰਡਵੇਅਰ, ਸਾਫਟਵੇਅਰ ਜਾਂ ਫਰਮਵੇਅਰ ਵਿਚ ਕੋਈ ਐਰਰ (error) ਹੋ ਸਕਦਾ ਹੈ।

ਜੇਕਰ ਵਿੰਡੋ ਸਹੀ ਢੰਗ ਨਾਲ ਸ਼ੁਰੂ ਨਹੀਂ ਹੋ ਰਹੀ ਹੈ, ਤਾਂ ਇਸਨੂੰ ਠੀਕ ਕਰਨ ਲਈ ਅਸੀਂ "ਸਟਾਰਟ-ਅੱਪ ਰਿਪੇਅਰ" ਟੂਲ (Startup Repaire Tool) ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਹ ਰਿਕਵਰੀ ਟੂਲ ਸਾਡੇ PC ਨੂੰ ਮਿਸਿੰਗ ਜਾਂ ਖਰਾਬ ਸਿਸਟਮ ਫਾਈਲਾਂ ਵਰਗੀਆਂ ਸਮੱਸਿਆਵਾਂ ਲਈ ਸਕੈਨ ਕਰੇਗਾ। ਇਹ ਹਾਰਡਵੇਅਰ ਸਮੱਸਿਆਵਾਂ ਜਾਂ ਵਿੰਡੋਜ਼ ਇੰਸਟਾਲੇਸ਼ਨ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਹੱਲ ਨਹੀਂ ਕਰ ਸਕਦਾ ਹੈ। ਪਰ ਜੇਕਰ ਸਾਨੂੰ ਵਿੰਡੋਜ਼ ਵਿੱਚ ਬੂਟ ਕਰਨ ਵਿੱਚ ਮੁਸ਼ਕਲ ਆ ਰਹੀ ਹੈ ਤਾਂ ਇਹ ਆਪਸ਼ਨ ਸ਼ੁਰੂ ਕਰਨ ਲਈ ਬਹੁਤ ਉਪਯੋਗੀ ਹੈ। ਅਸੀਂ ਇਸਨੂੰ ਬਿਲਟ-ਇਨ ਵਿੰਡੋਜ਼ ਰਿਕਵਰੀ ਟੂਲਸ, ਰਿਕਵਰੀ ਮੀਡੀਆ, ਜਾਂ ਵਿੰਡੋਜ਼ ਇੰਸਟਾਲੇਸ਼ਨ ਡਿਸਕ ਤੋਂ ਐਕਸੈਸ ਕਰ ਸਕਦੇ ਹਾਂ।

ਕਈ ਹੋਰ ਕਿਸਮ ਦੀਆਂ PC ਸਮੱਸਿਆਵਾਂ ਲਈ, ਅਸੀਂ ਵਿੰਡੋਜ਼ ਨੂੰ ਸੇਫ ਮੋਡ (Safe Mode) ਵਿੱਚ ਵੀ ਬੂਟ ਕਰ ਸਕਦੇ ਹਾਂ। ਸੇਫ ਮੋਡ ਕੰਪਿਊਟਰ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦਾ ਇੱਕ ਡਾਇਗਨੌਸਟਿਕ ਮੋਡ ਹੈ। ਜਦੋਂ ਵਿੰਡੋ OS ਨੂੰ ਸਟਾਰਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ ਆਮ ਤੌਰ ਤੇ ਸਟਾਰਟ-ਅੱਪ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਲਾਂਚ ਕਰਦੀ ਹੈ, ਕੌਨਫਿਗਰ ਕੀਤੀਆਂ ਸਾਰੀਆਂ ਸੇਵਾਵਾਂ

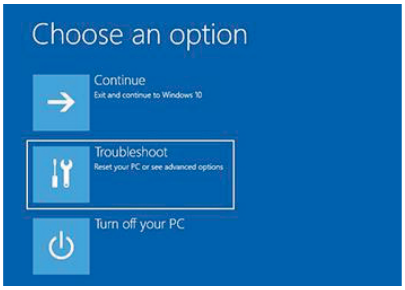
(configured services) ਨੂੰ ਚਾਲੂ ਕਰਦੀ ਹੈ, ਅਤੇ ਸਾਡੇ ਦੁਆਰਾ ਇੰਸਟਾਲ ਕੀਤੇ ਹਾਰਡਵੇਅਰ ਡਰਾਈਵਰਾਂ ਨੂੰ ਲੋਡ ਕਰਦੀ ਹੈ। ਵਿੰਡੋਜ਼ ਦੀ ਸੇਫ ਮੋਡ ਵਿਚ ਬੂਟਿੰਗ ਹੋਣ ਉਪਰੰਤ ਸਿਰਫ ਜ਼ਰੂਰੀ ਸਿਸਟਮ ਪ੍ਰੋਗਰਾਮਾਂ ਅਤੇ ਸੇਵਾਵਾਂ ਨੂੰ ਸ਼ੁਰੂ ਹੋਣ ਦੀ ਇਜਾਜ਼ਤ ਹੁੰਦੀ ਹੈ। ਸੇਫ ਮੋਡ ਸਾਡੇ PC ਨੂੰ ਡਰਾਈਵਰਾਂ ਦੇ ਘੱਟੋ-ਘੱਟ ਸੈੱਟ ਨਾਲ ਸ਼ੁਰੂ ਕਰਦਾ ਹੈ। ਇਸ ਮੋਡ ਵਿਚ ਵਿੰਡੋਜ਼ ਆਮ ਵੀਡੀਓ ਡਰਾਈਵਰਾਂ ਦੇ ਨਾਲ ਬਹੁਤ ਘੱਟ ਸਕਰੀਨ ਰੈਜ਼ੋਲਿਊਸ਼ਨ (resolution) ਦੀ ਵਰਤੋਂ ਕਰਦੀ ਹੈ ਅਤੇ ਇਸ ਮੋਡ ਵਿੱਚ ਹਾਰਡਵੇਅਰ ਬਹੁਤ ਜ਼ਿਆਦਾ ਸਪੋਰਟ ਨਹੀਂ ਕਰਦਾ।

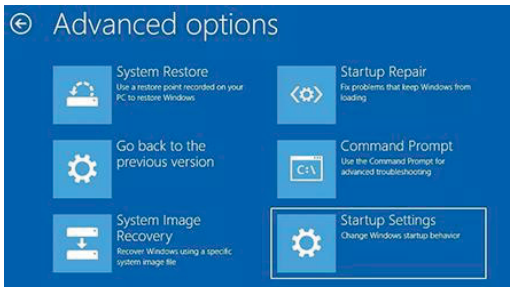
ਸੇਫ ਮੋਡ ਦਾ ਉਦੇਸ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਜ਼ਿਆਦਾਤਰ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਹੱਲ ਕਰਨ ਵਿੱਚ ਮਦਦ ਕਰਨਾ ਹੈ। ਸੇਫ ਮੋਡ ਸਮੱਸਿਆ ਪੈਦਾ ਕਰਨ ਵਾਲੇ ਸਾਫਟਵੇਅਰ ਨੂੰ ਹਟਾਉਣ ਦਾ ਇੱਕ ਵਧੀਆ ਤਰੀਕਾ ਹੈ - ਜਿਵੇਂ ਕਿ ਮਾਲਵੇਅਰ, ਅਸਥਿਰ (unstable) ਹਾਰਡਵੇਅਰ ਡ੍ਰਾਈਵਰ ਆਦਿ। ਇਹ ਇੱਕ ਅਜਿਹਾ ਵਾਤਾਵਰਣ ਵੀ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ ਜਿੱਥੇ ਅਸੀਂ ਡਰਾਈਵਰਾਂ ਨੂੰ ਆਸਾਨੀ ਨਾਲ ਰੋਲ ਬੈਕ (roll back) ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ, ਅਤੇ ਕੁਝ ਸਮੱਸਿਆਵਾਂ ਦੇ ਹੱਲ ਲਈ ਟ੍ਰਬਲਸ਼ੂਟਿੰਗ ਟੂਲਜ਼ (troubleshooting tools) ਵੀ ਵਰਤ ਸਕਦੇ ਹਾਂ।

6.3.1.1 ਸੇਫ ਮੋਡ ਵਿੱਚ ਵਿੰਡੋਜ ਨੂੰ ਕਿਵੇਂ ਸ਼ੁਰੂ ਕਰੀਏ (How to start Windows in Safe Mode):

ਜੇਕਰ ਸਾਡਾ PC ਆਮ ਤੌਰ ਤੇ ਸੁਰੂ ਹੋਣ ਸਮੇਂ ਇੱਕ ਤੋਂ ਵੱਧ ਵਾਰ ਕਰੈਸ਼ ਹੋ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ ਆਪਣੇ ਆਪ ਸੇਫ ਮੋਡ ਵਿੱਚ ਸ਼ੁਰੂ ਹੋ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਅਸੀਂ ਮੈਨੂਅਲੀ ਵੀ ਆਪਣੇ PC ਨੂੰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਬੂਟ ਕਰ ਸਕਦੇ ਹਾਂ:

- **ਵਿੰਡੋਜ਼ 7 ਅਤੇ ਪੁਰਾਣੇ ਵਰਜਨ ਵਿੱਚ:** ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਨਾਲ ਅੱਗੇ ਵਧੋ:
  1. ਜਦੋਂ ਕੰਪਿਊਟਰ ਬੂਟ ਹੋ ਰਿਹਾ ਹੋਵੇ ਤਾਂ F8 ਕੀਅ ਦਬਾਓ (ਸ਼ੁਰੂਆਤੀ BIOS ਸਕ੍ਰੀਨ ਤੋਂ ਬਾਅਦ, ਪਰ ਵਿੰਡੋਜ਼ ਲੋਡਿੰਗ ਸਕ੍ਰੀਨ ਤੋਂ ਪਹਿਲਾਂ)
  2. ਦਿਖਾਈ ਦੇ ਰਹੇ ਮੈਨੂੰ ਵਿੱਚੋਂ ਸੇਫ ਮੋਡ ਸਿਲੈਕਟ ਕਰੋ।
- **ਵਿੰਡੋਜ਼ 8 ਵਿੱਚ:** ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਨਾਲ ਅੱਗੇ ਵਧੋ:
  1. ਵਿੰਡੋਜ਼ ਲੋਗਇਨ ਸਕ੍ਰੀਨ 'ਤੇ ਪਾਵਰ ਬਟਨ ਦਬਾਓ। ਫਿਰ, ਕੀਬੋਰਡ ਤੋਂ ਸਿਫ਼ਟ ਬਟਨ ਨੂੰ ਦਬਾ ਕੇ ਰੱਖੋ ਅਤੇ ਰੀਸਟਾਰਟ ਤੇ ਕਲਿੱਕ ਕਰੋ।
  2. PC ਦੇ ਰੀਸਟਾਰਟ ਹੋਣ ਤੋਂ ਬਾਅਦ, Troubeshoot –> Advanced options –> Startup Settings –> Restart ਚੁਣੋ।
  3. PC ਦੇ ਮੁੜ ਚਾਲੂ ਹੋਣ ਤੋਂ ਬਾਅਦ, ਆਪਸ਼ਨਾਂ ਦੀ ਇੱਕ ਸੂਚੀ ਦਿਖਾਈ ਦੇਵੇਗੀ। PC ਨੂੰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਚਾਲੂ ਕਰਨ ਲਈ 4 ਜਾਂ F4 ਸਲੈਕਟ ਕਰੋ। (ਆਨ-ਸਕ੍ਰੀਨ ਨਿਰਦੇਸ਼ਾਂ ਦੀ ਪਾਲਣਾ ਕਰੋ)
- **ਵਿੰਡੋਜ਼ 10 ਅਤੇ ਵਿੰਡੋਜ਼ 11 ਵਿੱਚ:** ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਨਾਲ ਅੱਗੇ ਵਧੋ:
  1. ਡੈਸਕਟਾਪ ਸਕਰੀਨ ਤੇ, ਕੀਬੋਰਡ ਤੋਂ Windows key + X ਨੂੰ ਦਬਾਓ।
  2. **"Shut down or sign out"** ਮੀਨੂ ਤੋਂ Restart ਤੇ ਕਲਿੱਕ ਕਰਦੇ ਸਮੇਂ Shift key ਨੂੰ ਦਬਾ ਕੇ ਰੱਖੋ।
  3. PC ਰੀਸਟਾਰਟ ਹੋਣ ਤੋਂ ਬਾਅਦ, Troubleshoot –> Advanced options –> Startup Settings –> Restart ਚੁਣੋ।





4. PC ਰੀਸਟਾਰਟ ਹੋਣ ਤੋਂ ਬਾਅਦ, ਆਪਸ਼ਨਾਂ ਦੀ ਇੱਕ ਸੂਚੀ ਦਿਖਾਈ ਦੇਵੇਗੀ। PC ਨੂੰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਚਾਲੂ ਕਰਨ ਲਈ 4 ਜਾਂ F4 ਜਾਂ Fn+F4 (ਆਨ-ਸਕ੍ਰੀਨ ਨਿਰਦੇਸ਼ਾਂ ਦੀ ਪਾਲਣਾ ਕਰਦੇ ਹੋਏ) ਦੀ ਚੋਣ ਕਰੋ।

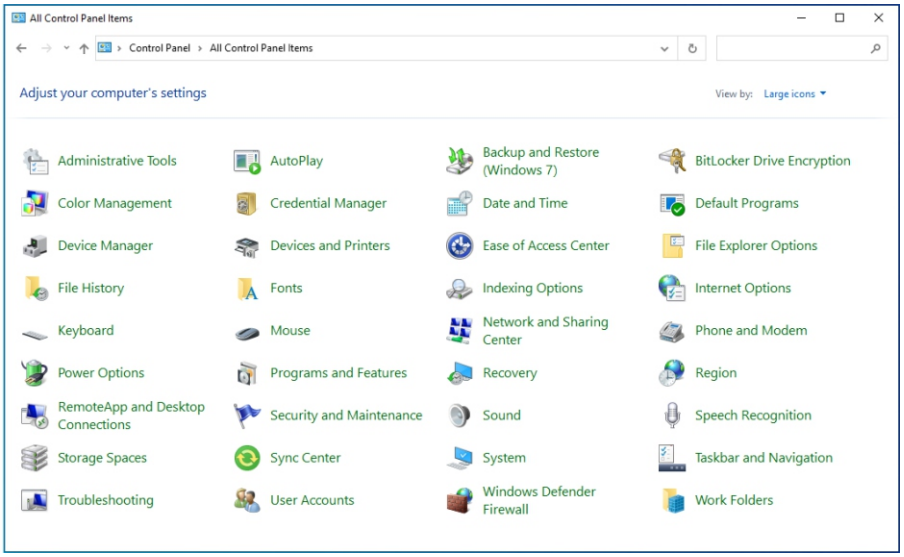
6.3.1.2 ਸੇਫ ਮੋਡ ਵਿੱਚ ਆਪਣੇ PC ਨੂੰ ਕਿਵੇਂ ਠੀਕ ਕਰਨਾ ਹੈ (How to Fix Your PC in Safe Mode):

ਵਿੰਡੋਜ਼ ਨੂੰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਸ਼ੁਰੂ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਅਸੀਂ ਆਪਣੇ ਕੰਪਿਊਟਰ ਨੂੰ ਠੀਕ ਕਰਨ ਲਈ ਜ਼ਿਆਦਾਤਰ ਸਿਸਟਮ ਮੈਨਟੇਨੈਂਸ ਅਤੇ ਟ੍ਰਬਲਸ਼ੂਟ (troubleshoot) ਕਰਨ ਦੇ ਕੰਮ ਕਰਦੇ ਹਾਂ:

- **ਮਾਲਵੇਅਰ ਲਈ ਸਕੈਨ ਕਰਨਾ (Scan for Malware):** ਐਂਟੀਵਾਇਰਸ ਐਪਲੀਕੇਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸਿਸਟਮ ਨੂੰ ਮਾਲਵੇਅਰ ਲਈ ਸਕੈਨ ਕਰੋ ਅਤੇ ਇਹਨਾਂ ਨੂੰ ਸੇਫ ਮੋਡ ਵਿਚ ਰਹਿ ਕੇ ਹਟਾਓ। ਮਾਲਵੇਅਰ ਨੂੰ ਨੋਰਮਲ ਮੋਡ ਤੋਂ ਹਟਾਉਣਾ ਅਸੰਭਵ ਹੋ ਸਕਦਾ ਹੈ ਕਿਉਂਕਿ ਇਹ ਬੇਕਗ੍ਰਾਊਂਡ ਵਿੱਚ ਚੱਲ ਰਹੇ ਹੁੰਦੇ ਹਨ ਅਤੇ ਇਹ ਐਂਟੀਵਾਇਰਸ ਵਿੱਚ ਦਖਲ ਦਿੰਦੇ ਹਨ। ਇਸ ਲਈ ਮਾਲਵੇਅਰਜ਼ ਸੇਫ ਮੋਡ ਵਿੱਚ ਆਸਾਨੀ ਨਾਲ ਹਟਾਉਣਯੋਗ ਹੁੰਦੇ ਹਨ। ਜੇਕਰ ਅਸੀਂ Windows 10 ਦੀ ਵਰਤੋਂ ਕਰ ਰਹੇ ਹਾਂ, ਤਾਂ ਅਸੀਂ ਸਿਸਟਮ ਵਿੱਚ ਮਾਲਵੇਅਰਾਂ ਨੂੰ ਸਕੈਨ ਕਰਨ ਲਈ ਆਫਲਾਈਨ ਸਕੈਨਰ Windows Defender ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਮੌਜੂਦਾ ਇੰਸਟਾਲਡ ਸਾਫਟਵੇਅਰ ਨੂੰ ਅਨਇੰਸਟਾਲ ਕਰਨਾ (Uninstall Recently Installed Software):** ਜੇਕਰ ਅਸੀਂ ਹਾਲ ਹੀ ਵਿੱਚ ਇੱਕ ਸਾਫਟਵੇਅਰ (ਜਿਵੇਂ ਕਿ ਇੱਕ ਹਾਰਡਵੇਅਰ ਡਰਾਈਵਰ ਜਾਂ ਇੱਕ ਪ੍ਰੋਗਰਾਮ ਜਿਸ ਵਿੱਚ ਇੱਕ ਡਰਾਈਵਰ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ) ਨੂੰ ਇੰਸਟਾਲ ਕੀਤਾ ਹੈ ਅਤੇ ਇਸ ਨਾਲ ਕੋਈ ਸਮੱਸਿਆ ਆ ਰਹੀ ਹੈ, ਤਾਂ ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਇਸਨੂੰ ਅਨਇੰਸਟਾਲ ਕਰ ਸਕਦੇ ਹਾਂ। ਸਾਡੇ ਸਿਸਟਮ ਵਿਚ ਦਖਲ ਦੇਣ ਵਾਲੇ ਸਾਫਟਵੇਅਰ ਨੂੰ ਅਣਇੰਸਟਾਲ ਕਰਨ ਤੋਂ ਬਾਅਦ ਕੰਪਿਊਟਰ ਨਾਰਮਲ ਸ਼ੁਰੂ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।
- **ਹਾਰਡਵੇਅਰ ਡ੍ਰਾਈਵਰਾਂ ਨੂੰ ਅੱਪਡੇਟ ਕਰਨਾ (Update Hardware Drivers):** ਜੇਕਰ ਹਾਰਡਵੇਅਰ ਡ੍ਰਾਈਵਰ ਸਾਡੇ ਸਿਸਟਮ ਦੀ ਅਸਥਿਰਤਾ (instability) ਦਾ ਕਾਰਨ ਬਣ ਰਹੇ ਹਨ, ਤਾਂ ਅਸੀਂ ਆਪਣੇ ਸਿਸਟਮ ਦੇ ਨਿਰਮਾਤਾ (manufacturer's ਦੀ ਵੈੱਬਸਾਈਟ ਤੋਂ ਅੱਪਡੇਟਿਡ ਡ੍ਰਾਈਵਰਾਂ ਨੂੰ ਡਾਊਨਲੋਡ ਕਰਕੇ ਉਨ੍ਹਾਂ ਨੂੰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਇੰਸਟਾਲ ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਸਿਸਟਮ ਕਰੈਸ਼ਾਂ ਦੀ ਜਾਂਚ ਕਰਨਾ (To check system crashes):** ਜੇਕਰ ਸਾਡਾ ਕੰਪਿਊਟਰ ਨਾਰਮਲ ਚਲਾਉਣ ਉਪਰੰਤ ਅਸਥਿਰ ਹੋ ਰਿਹਾ ਹੈ, ਪਰ ਇਹ ਸੇਫ ਮੋਡ ਵਿੱਚ ਵਧੀਆ ਕੰਮ ਕਰ ਰਿਹਾ ਹੈ, ਤਾਂ ਇਸ ਗੱਲ ਦੀ ਸੰਭਾਵਨਾ ਵੱਧ ਜਾਂਦੀ ਹੈ ਕਿ ਸਾਡੇ ਕੰਪਿਊਟਰ ਦੇ ਕਰੈਸ਼ ਹੋਣ ਦਾ ਕਾਰਨ ਕਿਸੇ ਸਾਫਟਵੇਅਰ ਵਿਚ ਸਮੱਸਿਆ ਹੈ। ਹਾਲਾਂਕਿ, ਜੇਕਰ ਕੰਪਿਊਟਰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਵੀ ਕ੍ਰੈਸ਼ ਹੁੰਦਾ ਰਹਿੰਦਾ ਹੈ, ਤਾਂ ਇਹ ਅਕਸਰ ਇਸ ਗੱਲ ਦਾ ਸੰਕੇਤ ਹੁੰਦਾ ਹੈ ਕਿ ਸਾਡੇ ਕੰਪਿਊਟਰ ਵਿੱਚ ਕੋਈ ਹਾਰਡਵੇਅਰ ਸਮੱਸਿਆ ਹੈ।
- **ਸਿਸਟਮ ਰੀਸਟੋਰ ਚਲਾਉਣਾ (Run System Restore):** ਸਿਸਟਮ ਰੀਸਟੋਰ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਨੂੰ ਪਿਛਲੀ ਬਿਹਤਰ ਸਥਿਤੀ ਵਿੱਚ ਵਾਪਸ ਲਿਜਾਉਣ ਲਈ ਇੱਕ ਮਾਈਕ੍ਰੋਸਾਫਟ ਵਿੰਡੋਜ਼ ਟੂਲ ਹੈ। ਇਸ ਅਵਸਥਾ ਨੂੰ **ਰੀਸਟੋਰ ਪੁਆਇੰਟ** ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਟੂਲ ਦੀ ਵਰਤੋਂ ਉਸ ਸਮੇਂ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ ਜਦੋਂ ਸਾਨੂੰ ਸਾਡੇ ਸਿਸਟਮ ਦੀਆਂ ਉਹਨਾਂ ਤਬਦੀਲੀਆਂ ਨੂੰ ਅਨਡੂ ਕਰਨ ਦੀ ਲੋੜ ਹੋਵੇ ਜੋ ਅਸੀਂ ਸਿਸਟਮ ਵਿਚ ਰੀਸਟੋਰ ਪੁਆਇੰਟ ਬਣਾਉਣ ਤੋਂ ਬਾਅਦ ਕੀਤੀਆਂ ਹੋਣ। ਜੇਕਰ ਸਾਡਾ ਕੰਪਿਊਟਰ ਪਹਿਲਾਂ ਠੀਕ ਕੰਮ ਕਰ ਰਿਹਾ ਸੀ, ਪਰ ਹੁਣ ਇਹ ਅਸਥਿਰ (unstable) ਹੈ, ਤਾਂ ਅਸੀਂ ਆਪਣੀ ਸਿਸਟਮ ਸਥਿਤੀ ਨੂੰ ਪਹਿਲਾਂ ਵਰਗੀ ਕੰਡੀਸ਼ਨ ਵਿਚ ਲਿਆਉਣ ਲਈ ਸਿਸਟਮ ਰੀਸਟੋਰ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

6.4 ਕੰਟਰੋਲ ਪੈਨਲ (CONTROL PANEL)

ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦਾ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਹਿੱਸਾ ਹੈ। ਇਹ ਯੂਜ਼ਰ ਨੂੰ ਸਿਸਟਮ ਸੈਟਿੰਗਾਂ ਨੂੰ ਵੇਖਣ ਅਤੇ ਬਦਲਣ ਦੀ ਸਹੂਲਤ ਦਿੰਦਾ ਹੈ। ਇਹ ਸਾਨੂੰ ਹਾਰਡਵੇਅਰ ਜਾਂ ਸਾਫਟਵੇਅਰ ਸੈਟਿੰਗਾਂ ਨੂੰ ਦੇਖਣ ਅਤੇ ਬਦਲਣ ਲਈ ਕਈ ਤਰ੍ਹਾਂ ਦੇ ਟੂਲ ਅਤੇ ਪੈਨਲ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਵਿੰਡੋਜ਼ ਕੰਟਰੋਲ ਪੈਨਲ ਨੂੰ ਵਿੰਡੋ ਦੇ ਸਟਾਰਟ ਮੀਨੂ ਦੇ ਸਰਚ ਬਾਕਸ **(seach box)** ਵਿੱਚ **"control panel"** ਟਾਈਪ ਕਰਕੇ ਖੋਲ੍ਹਿਆ ਜਾ ਸਕਦਾ ਹੈ।



ਚਿੱਤਰ 6.9: ਕੰਟਰੋਲ ਪੈਨਲ - ਵਿੰਡੋਜ਼ ਦਾ ਕਲਾਸਿਕ ਵਿਊ

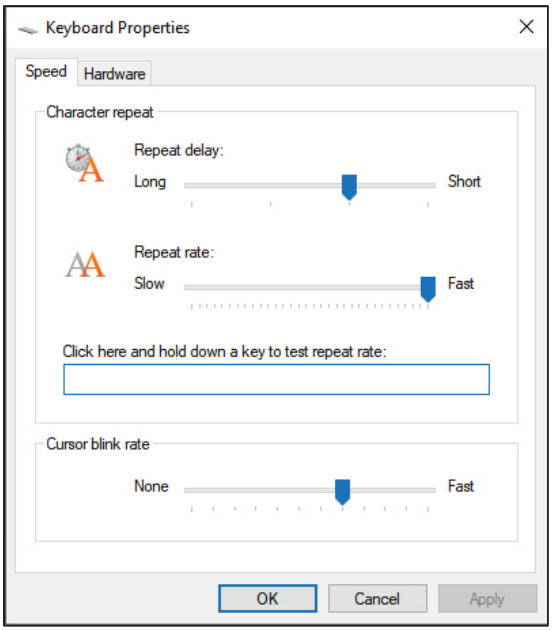
ਕੰਟੋਰਲ ਪੈਨਲ ਨੂੰ **ਕੈਟਾਗਰੀ ਵਿਊ** ਜਾਂ **ਕਲਾਸਿਕ ਵਿਊ** ਵਿੱਚ ਦੇਖਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਕੈਟਾਗਰੀ ਵਿਊ ਕੰਟਰੋਲ ਪੈਨਲਾਂ ਨੂੰ ਸੈਕਸ਼ਨਾਂ ਵਿੱਚ ਵਿਵਸਥਿਤ ਕਰਦਾ ਹੈ, ਜਦੋਂ ਕਿ ਕਲਾਸਿਕ ਵਿਊ ਉਹਨਾਂ ਸਾਰਿਆਂ ਨੂੰ ਇੱਕ ਵਾਰ ਵਿੱਚ ਦਿਖਾਉਂਦਾ ਹੈ।

ਕੰਟੋਰਲ ਪੈਨਲ ਨੂੰ **ਕੈਟਾਗਰੀ ਵਿਊ** ਜਾਂ **ਕਲਾਸਿਕ ਵਿਊ** ਵਿੱਚ ਦੇਖਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਕੈਟਾਗਰੀ ਵਿਊ ਕੰਟਰੋਲ ਪੈਨਲਾਂ ਨੂੰ ਸੈਕਸ਼ਨਾਂ ਵਿੱਚ ਵਿਵਸਥਿਤ ਕਰਦਾ ਹੈ, ਜਦੋਂ ਕਿ ਕਲਾਸਿਕ ਵਿਊ ਉਹਨਾਂ ਸਾਰਿਆਂ ਨੂੰ ਇੱਕ ਵਾਰ ਵਿੱਚ ਦਿਖਾਉਂਦਾ ਹੈ।

ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਉਪਲਬਧ ਕੁਝ ਮਹੱਤਵਪੂਰਨ ਟੂਲਸ ਜਾਂ ਪੈਨਲਾਂ ਦੀ ਵਿਆਖਿਆ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

6.4.1 ਕੀਬੋਰਡ (Keyboard):

ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੀਬੋਰਡ ਲਈ ਕੁਝ ਸੈਟਿੰਗਾਂ ਬਦਲ ਸਕਦੇ ਹਾਂ। ਕੀਬੋਰਡ ਸੈਟਿੰਗਾਂ ਨੂੰ ਦੇਖਣ ਅਤੇ ਬਦਲਣ ਲਈ, ਸਾਨੂੰ ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਕੀਬੋਰਡ ਆਈਕਨ ਤੇ ਕਲਿੱਕ ਕਰਨਾ ਹੋਵੇਗਾ, ਦਿੱਤੇ ਹੋਏ ਚਿੱਤਰ ਅਨੁਸਾਰ ਇੱਕ ਪ੍ਰੋਪਰਟੀਜ ਵਿੰਡੋ ਖੁਲੇਗੀ। ਕੀਬੋਰਡ ਪ੍ਰੋਪਰਟੀਜ ਵਿੰਡੋ ਵਿੱਚ ਦੋ ਟੈਬ ਹਨ: ਸਪੀਡ ਅਤੇ ਹਾਰਡਵੇਅਰ:



ਚਿੱਤਰ 6.10: ਕੀਬੋਰਡ ਪ੍ਰੋਪਰਟੀਜ਼ ਵਿੰਡੋਂ

- **ਸਪੀਡ ਟੈਬ (Speed tab):** ਇਸ ਟੈਬ ਦੀ ਵਰਤੋਂ ਕੀ-ਬੋਰਡ ਦੀਆਂ ਕੁਝ ਬੁਨਿਆਦੀ ਸੈਟਿੰਗਾਂ ਨੂੰ ਬਦਲਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਜਿਵੇਂ ਕਿ: **ਕਰੈਕਟਰ ਰੀਪੀਟ** ਅਤੇ **ਕਰਸਰ ਬਲਿੰਕ ਰੇਟ**।
  - ਅਸੀਂ ਕਰੈਕਟਰ ਰੀਪੀਟ ਵਾਲੇ ਭਾਗ ਵਿੱਚ ਦਿੱਤੇ ਸਲਾਈਡਰਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਰੀਪੀਟ ਡਿਲੇਅ ਅਤੇ ਰੀਪੀਟ ਰੇਟ ਨੂੰ ਸੈੱਟ ਕਰ ਸਕਦੇ ਹਾਂ।

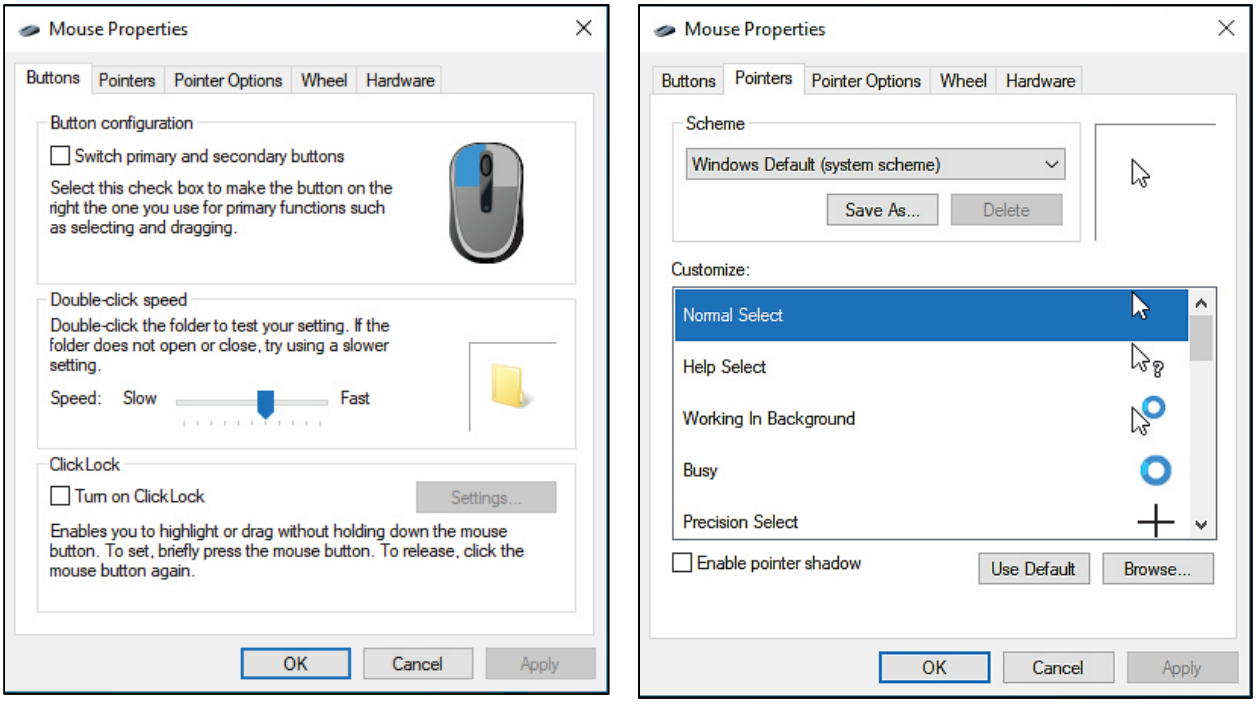
- ਅਸੀਂ ਕਰਸਰ ਬਲਿੰਕ ਰੇਟ ਸੈਕਸ਼ਨ ਵਿੱਚ ਦਿੱਤੇ ਗਏ ਸਲਾਈਡਰ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕਰਸਰ ਦੀ ਬਲਿੰਕਿੰਗ ਰੇਟ ਸੈੱਟ ਕਰ ਸਕਦੇ ਹਾਂ।

- **ਹਾਰਡਵੇਅਰ ਟੈਬ (Hardware Tab):** ਇਹ ਟੈਬ ਕੀਬੋਰਡ ਦੇ ਵਰਤਮਾਨ ਇੰਸਟਾਲੇਸ਼ਨ ਅਤੇ ਇਸਦੇ ਸਟੇਟਸ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ।

6.4.2 ਮਾਊਸ (Mouse):

ਕੰਪਿਊਟਰਾਂ ਉਪਰ ਮਾਊਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੱਖ-ਵੱਖ ਯੂਜਰਾਂ ਦੀਆਂ ਤਰਜੀਹਾਂ (preferences) ਵੱਖਰੀਆਂ ਹੁੰਦੀਆਂ ਹਨ। ਜੇਕਰ ਯੂਜ਼ਰ ਖੱਬੇ ਹੱਥ ਵਾਲਾ (left-handed) ਹੈ, ਤਾਂ ਉਹ ਪ੍ਰਾਇਮਰੀ ਮਾਊਸ ਬਟਨ ਦੇ ਬਦਲ ਕੇ ਕੰਮ ਕਰਨਾ ਪਸੰਦ ਕਰੇਗਾ, ਭਾਵ ਖੱਬੇ ਹੱਥ ਵਾਲਾ ਯੂਜ਼ਰ ਮਾਊਸ ਦੇ ਸੱਜੇ ਬਟਨ ਨੂੰ ਪ੍ਰਾਇਮਰੀ ਬਟਨ ਦੇ ਰੂਪ ਵਿੱਚ ਵਰਤਣਾ ਪਸੰਦ ਕਰੇਗਾ।

ਯੂਜ਼ਰ ਆਪਣੀ ਸਹੂਲਤ ਅਨੁਸਾਰ ਇਹ ਵੀ ਸੈਟਿੰਗ ਬਦਲ ਸਕਦੇ ਹਨ ਕਿ ਮਾਊਸ ਪੁਆਇੰਟਰ ਕਿੰਨੀ ਤੇਜ਼ੀ ਨਾਲ ਚੱਲੇਗਾ ਅਤੇ ਕਿੰਨ੍ਹੀ ਤੇਜ਼ ਗਤੀ ਨਾਲ ਉਹ ਡਬਲ-ਕਲਿੱਕ ਕਰੇਗਾ ਅਤੇ ਹੋਰ ਵੀ ਬਹੁਤ ਕੁਝ ਮਾਊਸ ਸੈਟਿੰਗਾਂ ਨੂੰ ਦੇਖਣ ਅਤੇ ਬਦਲਣ ਲਈ, ਯੂਜ਼ਰ ਨੂੰ ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਮਾਊਸ ਆਈਕਨ ਤੇ ਕਲਿੱਕ ਕਰਨਾ ਹੋਵੇਗਾ। ਇਹ ਦਿੱਤੇ ਚਿੱਤਰ ਵਿੱਚ ਦਰਸਾਏ ਅਨੁਸਾਰ ਪ੍ਰੋਪਰਟੀਜ ਵਿੰਡੋ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇਗਾ:



ਚਿੱਤਰ 6.11 : ਮਾਊਸ ਪ੍ਰੋਪਰਟੀਜ ਵਿੰਡ

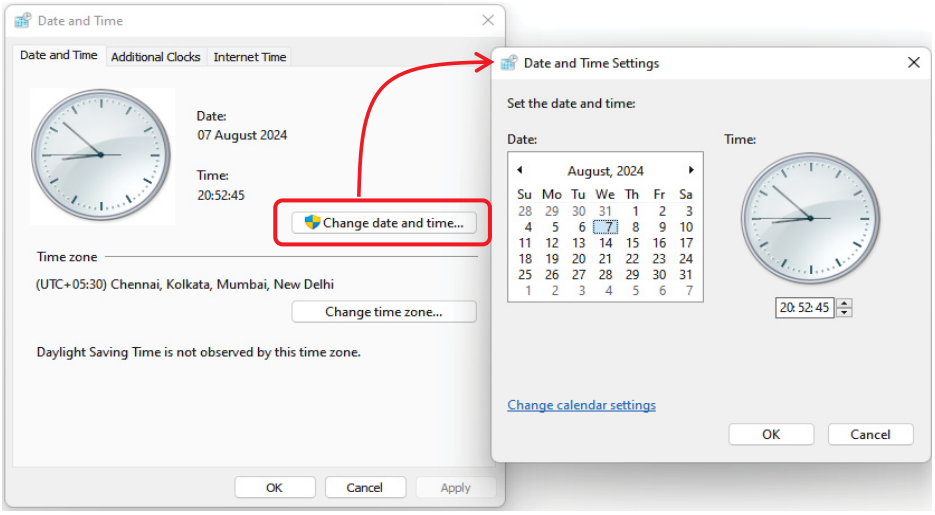
ਮਾਊਸ ਪ੍ਰੋਪਰਟੀਜ ਡਾਇਲਾਗ ਬਾਕਸ ਵਿੱਚ 5 ਟੈਬਜ਼ ਹੁੰਦੀਆਂ ਹਨ: ਬਟਨਜ਼, ਪੁਆਇੰਟਰ, ਪੁਆਇੰਟਰ ਆਪਸ਼ਨ, ਵ੍ਹੀਲ ਅਤੇ ਹਾਰਡਵੇਅਰ। ਕੁਝ ਆਮ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਮਾਊਸ ਸੈਟਿੰਗਾਂ ਨੂੰ ਹੇਠਾਂ ਸਮਝਾਇਆ ਗਿਆ ਹੈ:

- **ਬਟਨਜ਼ ਟੈਬ (Buttons tab):** ਇਹ ਟੈਬ ਸਾਨੂੰ ਮਾਊਸ ਦੇ ਫਿਜੀਕਲ ਬਟਨਾਂ ਦੀ ਸੈਟਿੰਗ ਨੂੰ ਐਡਜਸਟ ਕਰਨ ਦੀ ਸਹੁਲਤ ਦਿੰਦੀ ਹੈ:
  - ਇਸ ਟੈਬ ਦੀ ਵਰਤੋਂ ਕਰਕੇ, ਅਸੀਂ ਪ੍ਰਾਇਮਰੀ ਮਾਊਸ ਬਟਨ ਨੂੰ ਖੱਬੇ ਤੋਂ ਸੱਜੇ ਸਵਿੱਚ ਕਰ ਸਕਦੇ ਹਾਂ।
  - ਅਸੀਂ ਸਲਾਈਡਰ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਡਬਲ ਕਲਿੱਕ ਸਪੀਡ ਨੂੰ ਵੀ ਐਡਜਸਟ ਕਰ ਸਕਦੇ ਹਾਂ।
  - ਅਸੀਂ ਕਲਿੱਕ ਲੱਾਕ ਆਨ ਨੂੰ ਟੌਗਲ (toggle) ਕਰ ਸਕਦੇ ਹਾਂ, ਤਾਂ ਜੋ ਅਸੀਂ ਮਾਊਸ ਬਟਨ ਨੂੰ ਦਬਾਏ ਬਿਨਾਂ ਕਲਿੱਕ-ਅਤੇ-ਹੋਲਡ ਐਕਸ਼ਨ ਕਰਨ ਯੋਗ ਬਣ ਸਕੀਏ।
- **ਪੁਆਇੰਟਰ ਟੈਬ (Pointers tab):** ਇਸ ਟੈਬ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਅਸੀਂ ਕਰਸਰ ਆਈਕਨਾਂ ਨੂੰ ਬਦਲ ਸਕਦੇ ਹਾਂ। ਅਸੀਂ ਕਰਸਰਾਂ ਦੇ ਪ੍ਰੀ-ਇੰਸਟਾਲ ਸੰਗ੍ਰਹਿ ਵਿੱਚੋਂ ਕਰਸਰ ਆਈਕਨ ਸਿਲੈਕਟ ਕਰਨ ਲਈ "ਸਕੀਮ" ਡ੍ਰਾਪ-ਡਾਉਨ ਮੀਨੂ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਪੁਆਇੰਟਰ ਆਪਸ਼ਨ ਟੈਬ (Pointers Options tab):** ਇਸ ਟੈਬ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਅਸੀਂ ਮਾਊਸ ਦੇ ਕਰਸਰ ਨੂੰ ਸਕਰੀਨ ਤੇ ਘੁੰਮਾਉਣ ਦੀ ਸੈਟਿੰਗ ਨੂੰ ਬਦਲ ਸਕਦੇ ਹਾਂ।
- **ਵ੍ਹੀਲ ਟੈਬ (Wheel tab):** ਇਸ ਟੈਬ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਅਸੀਂ ਮਾਊਸ ਵ੍ਹੀਲ ਦੇ ਸਕਰੋਲ ਕਰਨ ਦੀ ਸਪੀਡ ਨੂੰ ਬਦਲ ਸਕਦੇ ਹਾਂ। ਵ੍ਹੀਲ ਟੈਬ ਦੀਆਂ ਸੈਟਿੰਗਾਂ ਇਸ ਗੱਲ ਤੇ ਅਸਰ ਪਾਉਂਦੀਆਂ ਹਨ ਕਿ ਅਸੀਂ ਡਾਕੂਮੈਂਟਾਂ ਅਤੇ ਵੈਬ ਪੇਜਾਂ ਨੂੰ ਕਿੰਨੀ ਤੇਜ਼ੀ ਨਾਲ ਸਕ੍ਰੋਲ ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਹਾਰਡਵੇਅਰ ਟੈਬ (Hardware Tab):** ਹਾਰਡਵੇਅਰ ਟੈਬ ਇੰਸਟਾਲਡ ਮਾਊਸ ਅਤੇ ਉਸ ਦੇ ਸਟੇਟਸ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ।

6.4.3 ਡੇਟ ਅਤੇ ਟਾਈਮ (Date and Time):

ਮੌਜੂਦਾ ਮਿਤੀ ਅਤੇ ਸਮਾਂ ਹਮੇਸ਼ਾ ਟਾਸਕਬਾਰ ਦੇ ਸੂਚਨਾ ਖੇਤਰ (notification area) ਵਿੱਚ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ। ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਰਾਹੀਂ ਮਿਤੀ ਅਤੇ ਸਮੇਂ ਨਾਲ ਸੰਬੰਧਿਤ ਕਈ ਸੈਟਿੰਗਾਂ ਨੂੰ ਸੈੱਟ ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਬਦਲ ਸਕਦੇ ਹਾਂ। ਡੇਟ ਐਂਡ ਟਾਈਮ ਪ੍ਰੋਪਰਟੀਜ ਵਿੰਡੋ ਨੂੰ ਖੋਲ੍ਹਣ ਲਈ ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਡੇਟ ਐਂਡ ਟਾਈਮ ਆਈਕਨ 'ਤੇ ਕਲਿੱਕ ਕਰੋ:

- **ਮੌਜੂਦਾ ਮਿਤੀ ਅਤੇ ਸਮਾਂ ਬਦਲਣਾ (Change Current Date and Time):** ਡੇਟ ਐਂਡ ਟਾਈਮ ਪ੍ਰਾਪਰਟੀ ਵਿੰਡੋ ਖੁੱਲ੍ਹਣ ਤੋਂ ਬਾਅਦ, ਅਸੀਂ ਆਪਣੇ ਕੰਪਿਊਟਰ ਲਈ ਕੁਝ ਬੁਨਿਆਦੀ ਸੈਟਿੰਗਾਂ ਨੂੰ ਬਦਲ ਸਕਦੇ ਹਾਂ। ਡੇਟ ਐਂਡ ਟਾਈਮ ਵਿੰਡੋ ਵਿੱਚ, ਡੇਟ ਅਤੇ ਟਾਈਮ (Date and Time) ਟੈਬ ਦੇ ਹੇਠਾਂ, ਚੇਂਜ ਡੇਟ ਐਂਡ ਟਾਈਮ **(Change date and time...)** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।



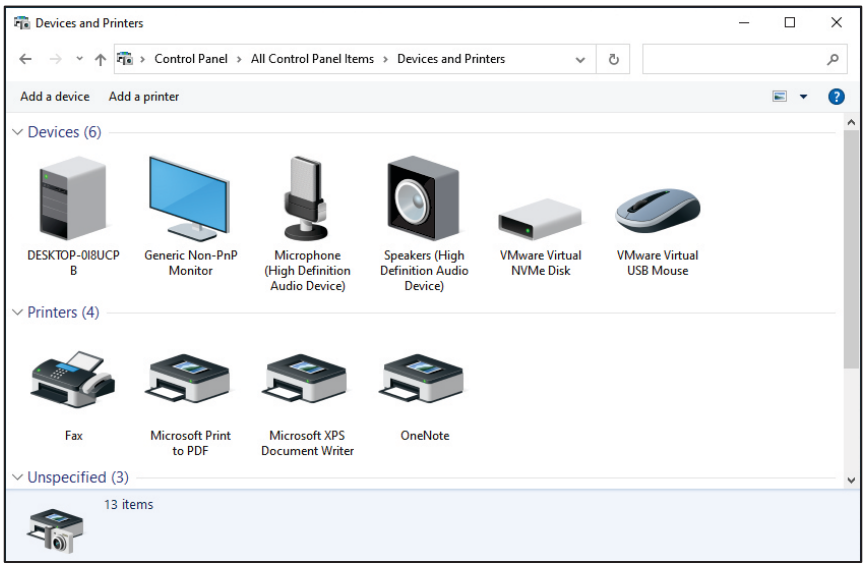
ਚਿੱਤਰ 6.12: ਡੇਟ ਐਂਡ ਟਾਈਮ ਪ੍ਰੋਪਰਟੀਜ਼ ਵਿੰਡੋ

- ਜ਼ਰੂਰਤ ਅਨੁਸਾਰ ਮਿਤੀ ਅਤੇ ਸਮੇਂ ਵਿੱਚ ਬਦਲਾਅ ਕਰੋ ਅਤੇ ਫਿਰ OK ਕੇ ਕਲਿੱਕ ਕਰੋ।
- ਹੁਣ, ਤਬਦੀਲੀਆਂ ਨੂੰ ਸੇਵ ਕਰਨ ਲਈ ਡੇਟ ਐਂਡ ਟਾਈਮ ਵਿੰਡੋ ਵਿਚ OK ਤੇ ਕਲਿੱਕ ਕਰੋ।

ਟਾਈਮ ਜੋਨ ਨੂੰ ਐਡਜਸਟ ਕਰਨਾ (Adjusting the time zone): ਅਸੀਂ ਇਸ ਪ੍ਰਾਪਰਟੀ ਵਿੰਡੋ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ, ਜਿਸ ਦੇਸ਼ ਵਿੱਚ ਰਹਿੰਦੇ ਹਾਂ ਉਸ ਦੇ ਅਨੁਸਾਰ ਟਾਈਮ ਜੋਨ ਨੂੰ ਵੀ ਬਦਲ ਸਕਦੇ ਹਾਂ:

- ਚੇਂਜ ਟਾਈਮ ਜੋਨ ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।
- ਟਾਈਮ ਜ਼ੋਨ ਡ੍ਰਾਪ-ਡਾਉਨ ਫੀਲਡ ਵਿੱਚ ਨਵਾਂ ਟਾਈਮ ਜ਼ੋਨ ਚੁਣੋ ਅਤੇ OK ਤੇ ਕਲਿੱਕ ਕਰੋ।
- ਟਾਈਮ ਜ਼ੋਨ ਤਬਦੀਲੀਆਂ ਨੂੰ ਸੇਵ ਕਰਨ ਲਈ ਮੇਨ ਡੇਟ ਐਂਡ ਟਾਈਮ ਵਿੰਡੋ ਵਿੱਚ OK ਤੇ ਕਲਿੱਕ ਕਰੋ।

6.4.4 ਡਿਵਾਈਸਿਸ ਅਤੇ ਪ੍ਰਿੰਟਰਜ਼ (Devices and Printers): ਡਿਵਾਈਸਿਸ ਅਤੇ ਪ੍ਰਿੰਟਰਜ਼ ਪੈਨਲ ਪਹਿਲੀ ਵਾਰ ਵਿੰਡੋਜ਼ 7 ਵਿੱਚ ਸਾਡੇ ਕੰਪਿਊਟਰ ਨਾਲ ਜੁੜੇ ਬਾਹਰੀ ਡਿਵਾਈਸਾਂ ਨਾਲ ਇੰਟਰਐਕਟ ਕਰਨ ਦੇ ਉਦੇਸ਼ ਨਾਲ ਪੇਸ਼ ਕੀਤਾ ਗਿਆ ਸੀ। ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਡਿਵਾਈਸਿਸ ਅਤੇ ਪ੍ਰਿੰਟਰਜ਼ ਵਿੰਡੋ ਨੂੰ ਖੋਲ੍ਹ ਸਕਦੇ ਹਾਂ। ਡਿਵਾਈਸਿਸ ਅਤੇ ਪ੍ਰਿੰਟਰਜ਼ ਵਿੰਡੋ ਵਿੱਚ ਅਸੀਂ ਕੰਪਿਊਟਰ ਨਾਲ ਜੁੜੇ ਬਾਹਰੀ ਡਿਵਾਈਸਾਂ ਦੇ ਨਾਲ ਨਾਲ ਆਪਣੇ ਕੰਪਿਊਟਰ ਨੂੰ ਵੀ ਦੇਖ

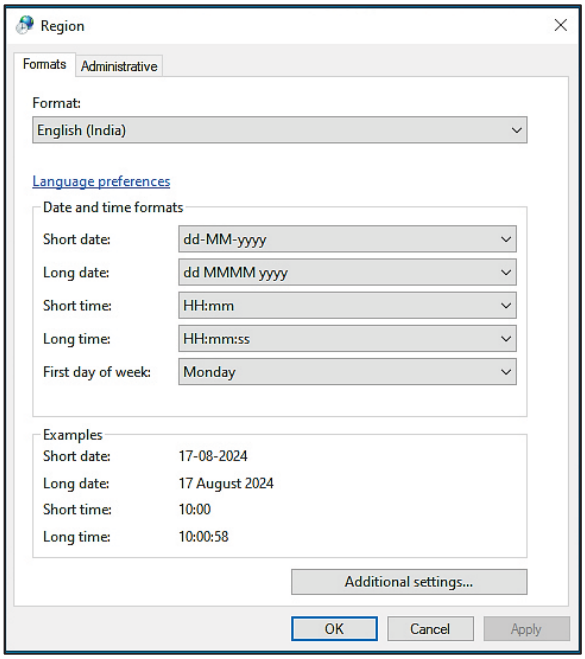


ਚਿੱਤਰ 6.13 ਡਿਵਾਈਸਿਸ ਅਤੇ ਪ੍ਰਿੰਟਰਜ਼ ਪੈਨਲ

ਸਕਦੇ ਹਾਂ। ਇਸ ਵਿੱਚ ਸ਼ਾਮਲ ਕੀਤੇ ਗਏ ਯੰਤਰਾਂ ਦੀ ਸੂਚੀ ਇਸ ਪ੍ਰਕਾਰ ਹੈ: ਸਮਾਰਟਫ਼ੋਨ, ਪੋਰਟੇਬਲ ਮਿਊਜਿਕ ਪਲੇਅਰ, ਡਿਜੀਟਲ ਕੈਮਰੇ, ਵੈਬਕੈਮ, ਮਾਨੀਟਰ, ਕੀਬੋਰਡ, ਮਾਊਸ, ਪ੍ਰਿੰਟਰ, ਸਕੈਨਰ, ਬਲੂਟੁੱਥ ਅਡਾਪਟਰ, ਐਕਸਟਰਨਲ ਹਾਰਡ ਡਰਾਈਵਾਂ, ਮੀਡੀਆ ਐਕਸਟੈਂਡਰ ਅਤੇ ਸਾਡੇ ਕੰਪਿਊਟਰ ਨਾਲ ਜੁੜੇ ਨੈੱਟਵਰਕ ਡਿਵਾਈਸ।

6.4.5 ਰੀਜ਼ਨਲ ਸੈਟਿੰਗਜ਼ (Regional Settings):

ਰੀਜ਼ਨਲ ਸੈਟਿੰਗਜ਼ ਯੂਜ਼ਰ ਦੀ ਲੋਕੇਸ਼ਨ ਨਾਲ ਸਬੰਧਤ ਹੁੰਦੀਆਂ ਹਨ, ਜਿਵੇਂ ਕਿ ਭਾਸ਼ਾ, ਕਰੰਸੀ, ਅਤੇ ਟਾਈਮ ਜ਼ੋਨ। ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੰਡੋ ਵਿੱਚ "ਰੀਜ਼ਨ ਐਂਡ ਲੈਂਗੁਏਜ" (ਵਿੰਡੋਜ਼ 7 ਵਿੱਚ) ਜਾਂ "ਰੀਜ਼ਨ" (ਵਿੰਡੋਜ਼ 10 ਅਤੇ 11) ਆਈਕਨ ਤੇ ਕਲਿੱਕ ਕਰਕੇ ਰੀਜ਼ਨਲ ਸੈਟਿੰਗਜ਼ ਨੂੰ ਦੇਖ ਅਤੇ ਬਦਲ ਸਕਦੇ ਹਾਂ, ਜਿਵੇਂ ਕਿ ਚਿੱਤਰ ਵਿੱਚ ਇਸ ਪ੍ਰੋਪਰਟੀ ਵਿੰਡੋ ਨੂੰ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਇਸ ਵਿੰਡੋ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਅਸੀਂ ਮਿਤੀ, ਸਮਾਂ, ਸੰਖਿਆਵਾਂ ਅਤੇ ਮੁਦਰਾ (ਕਰੰਸੀ) (date, time, numbers, and currency) ਦੇ ਫਾਰਮੈਟਾਂ ਨੂੰ ਬਦਲ ਸਕਦੇ ਹਾਂ:

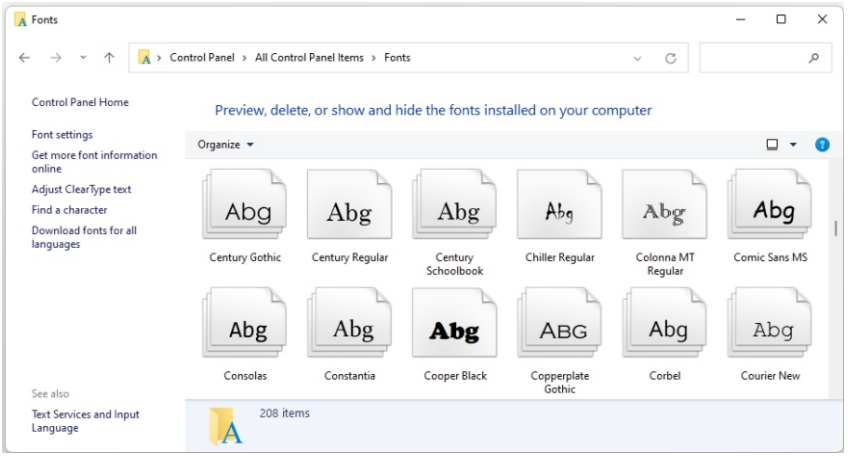


ਚਿੱਤਰ 6.14: ਰੀਜ਼ਨ ਪ੍ਰੋਪਰਟੀਜ ਵਿੰਡੋ

- ਫਾਰਮੈਟ **(Formats)** ਡਰਾਪ ਡਾਊਨ ਸੂਚੀ ਵਿੱਚ ਉਹ ਦੇਸ਼ ਸਿਲੈਕਟ ਕਰੋ ਜਿਸ ਅਨੁਸਾਰ ਅਸੀਂ ਮਿਤੀ ਅਤੇ ਸਮਾਂ, ਨੰਬਰ ਅਤੇ ਕਰੰਸੀ ਦੇ ਫਾਰਮੈਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।
- ਚੁਣੇ ਹੋਏ ਰੀਜ਼ਨ ਲਈ ਫਾਰਮੈਟਾਂ ਨੂੰ ਕਸਟਮਾਈਜ਼ ਕਰਨ ਲਈ, "ਐਡੀਸ਼ਨਲ ਸੈਟਿੰਗਜ਼" ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ। ਹੁਣ ਕਸਟਮਾਈਜ਼ ਫਾਰਮੈਟਸ ਡਾਇਲਾਗ ਬਾਕਸ ਦਿਖਾਈ ਦੇਵੇਗਾ।
- ਲੋੜ ਅਨੁਸਾਰ ਇਸ ਡਾਇਲਾਗ ਬਾਕਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਮਿਤੀ ਅਤੇ ਸਮਾਂ, ਨੰਬਰ ਅਤੇ ਕਰੰਸੀ ਫਾਰਮੈਟਾਂ ਨੂੰ ਕਸਟਮਾਈਜ਼ ਕਰੋ। ਫਾਰਮੈਟਾਂ ਨੂੰ ਕਸਟਮਾਈਜ਼ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਤਬਦੀਲੀਆਂ ਨੂੰ ਸੇਵ ਕਰਨ ਲਈ **Apply** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ ਅਤੇ ਫਿਰ **OK** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।
- ਹੁਣ "ਰੀਜ਼ਨਲ" ਵਿੰਡੋ ਨੂੰ ਬੰਦ ਕਰਨ ਲਈ ਦੁਬਾਰਾ **"OK"** ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।

6.4.6 ਫੌਂਟਸ (Fonts):

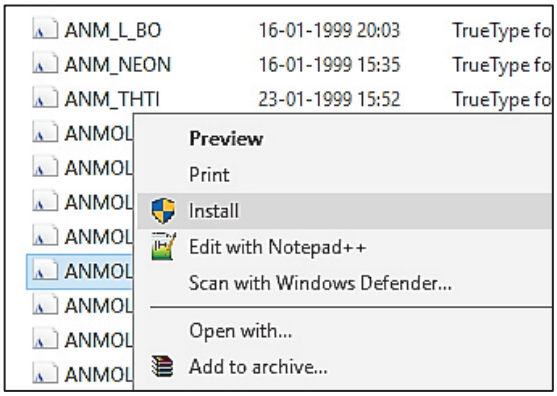
ਇੱਕ ਫੌਂਟ ਟਾਈਪਫੇਸ ਅਤੇ ਹੋਰ ਗੁਣਾਂ ਦਾ ਸੁਮੇਲ ਹੁੰਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਆਕਾਰ, ਪਿੱਚ, ਅਤੇ ਸਪੇਸਿੰਗ। ਉਦਾਹਰਨ ਲਈ: ਟਾਈਮਜ਼ ਨਿਊ ਰੋਮਨ ਇੱਕ ਟਾਈਪਫੇਸ ਹੈ ਜੋ ਹਰੇਕ ਅੱਖਰ ਦੀ ਸ਼ਕਲ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦਾ ਹੈ। ਹਾਲਾਂਕਿ, ਟਾਈਮਜ਼ ਨਿਊ ਰੋਮਨ ਦੇ ਅੰਦਰ, ਚੁਣਨ ਲਈ ਬਹੁਤ ਸਾਰੇ ਫੌਂਟਸ ਹਨ – ਵੱਖ-ਵੱਖ ਆਕਾਰ, ਇਟਾਲਿਕ, ਬੋਲਡ, ਅਤੇ ਹੋਰ। ਕੈਲੀਬਰੀ, ਟਾਈਮਜ਼ ਨਿਊ ਰੋਮਨ, ਏਰੀਅਲ, ਅਨਮੋਲ ਲਿਪੀ, ਜੁਆਏ, ਅਸੀਸ, ਰਾਵੀ, ਗੁਰਬਾਣੀ ਹਿੰਦੀ ਆਦਿ ਆਮ ਵਰਤੇ ਜਾਂਦੇ ਫੌਂਟਸ ਦੀਆਂ ਉਦਾਹਰਣਾਂ ਹਨ। ਜਦੋਂ ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਫੌਂਟਸ ਆਈਕਨ ਤੇ ਕਲਿੱਕ ਕਰਕੇ ਹਾਂ, ਤਾਂ ਇਹ ਸਾਡੇ ਸਿਸਟਮ ਵਿੱਚ ਇੰਸਟਾਲ ਹੋਏ ਸਾਰੇ ਫੌਂਟਸ ਨੂੰ ਡਿਸਪਲੇ ਕਰੇਗਾ, ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਦਿੱਤੀ ਵਿੰਡੋ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ:



ਚਿੱਤਰ 6.15: ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਫੌਂਟਸ

ਅਸੀਂ ਕੰਟਰੋਲ ਪੈਨਲ ਵਿੱਚ ਫੌਂਟ ਵਿੰਡੋ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਆਪਣੇ ਕੰਪਿਊਟਰ ਵਿੱਚ ਨਵੇਂ ਫੌਂਟ ਜੋੜ ਸਕਦੇ ਹਾਂ ਜਾਂ ਮੌਜੂਦਾ ਫੌਂਟਸ ਨੂੰ ਹਟਾਅ ਵੀ ਸਕਦੇ ਹਾਂ।

ਇੱਕ ਨਵਾਂ ਫੌਂਟ ਇੰਸਟਾਲ ਕਰਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਉਸਨੂੰ ਆਪਣੇ ਦਸਤਾਵੇਜ਼ਾਂ ਵਿੱਚ ਵਰਤ ਸਕਦੇ ਹਾਂ। ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਆਪਣੇ ਸਿਸਟਮ ਵਿੱਚ ਫੌਂਟਸ ਨੂੰ ਆਸਾਨੀ ਨਾਲ ਇੰਸਟਾਲ ਕਰ ਸਕਦੇ ਹਾਂ:



ਚਿੱਤਰ 6.16: ਫੌਂਟਸ ਇੰਸਟਾਲੇਸ਼ਨ

- ਇੰਟਰਨੈੱਟ ਤੋਂ ਨਵਾਂ ਫੌਂਟ ਡਾਊਨਲੋਡ ਕਰੋ।
- ਉਸ ਫੋਲਡਰ ਨੂੰ ਓਪਨ ਕਰੋ ਜਿਸ ਵਿੱਚ ਅਸੀਂ ਫੌਂਟ ਡਾਊਨਲੋਡ ਕੀਤਾ ਹੈ।
- ਫੌਂਟ ਫਾਈਲ ਤੇ ਰਾਈਟ ਕਲਿਕ ਕਰੋ ਅਤੇ ਮੀਨੂੰ ਵਿੱਚ ਦਿਖਾਈ ਦੇਣ ਵਾਲੇ ਇੰਸਟਾਲ ਆਪਸ਼ਨ ਤੇ ਕਲਿੱਕ ਕਰੋ, ਜਿਵੇਂ ਕਿ ਦਿੱਤੇ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ।

ਨੋਟ: ਜੇਕਰ ਡਾਊਨਲੋਡ ਕੀਤੇ ਫੌਂਟ ਜ਼ਿਪਡ ਫਾਰਮੈਟ ਵਿੱਚ ਹੋਣ, ਤਾਂ ਸਾਨੂੰ ਪਹਿਲਾਂ ਉਹਨਾਂ ਫੌਂਟਸ ਦੀਆਂ ਇੰਸਟਾਲੇਸ਼ਨ ਫਾਈਲਾਂ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਉਹਨਾਂ ਨੂੰ ਅਨਜ਼ਿਪ ਕਰਨਾ ਪਵੇਗਾ।

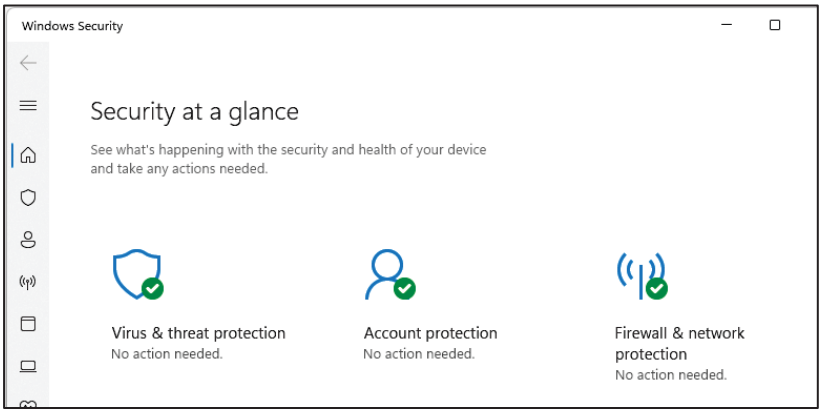
6.5 ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮਜ਼ (UTILITY PROGRAMS)

ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮ ਸਿਸਟਮ ਸਾਫਟਵੇਅਰਾਂ ਦੀ ਸ਼੍ਰੇਣੀ ਵਿੱਚ ਆਉਂਦੇ ਹਨ। ਇਹ ਪ੍ਰੋਗਰਾਮਜ਼ ਜਾਂ ਟੂਲਜ਼ ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਸਹੀ ਅਤੇ ਨਿਰਵਿਘਨ ਕੰਮਕਾਜ ਨੂੰ ਬਣਾਈ ਰੱਖਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੇ ਹਨ। ਇਹ ਪ੍ਰੋਗਰਾਮਜ਼ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਕੰਮਕਾਜ ਦੇ ਪ੍ਰਬੰਧਣ (organize), ਰੱਖ-ਰਖਾਅ (maintain) ਅਤੇ ਉਸਨੂੰ ਅਨੁਕੂਲ (optimize) ਬਣਾਉਣ ਲਈ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਨੂੰ ਸਹਾਇਤਾ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ।

ਯੂਟੀਲਿਟੀ ਸਾਫਟਵੇਅਰ ਦੁਆਰਾ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਕੁੱਝ ਆਮ ਕੰਮ ਇਸ ਤਰ੍ਹਾਂ ਹਨ: ਵਾਇਰਸ ਲੱਭਣਾ ਅਤੇ ਮਿਟਾਉਣਾ, ਡਾਟਾ-ਬੈਕਅੱਪ, ਫਾਈਲ-ਕੰਪਰੈਸ਼ਨ, ਡਿਸਕ ਮੈਨੇਜਮੈਂਟ, ਡਿਸਕ ਕਲੀਨਅੱਪ, ਡਿਸਕ ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ, ਫਾਈਲ ਮੈਨੇਜਮੈਂਟ ਆਦਿ। ਇਸ ਪਾਠ ਵਿੱਚ ਅਸੀਂ ਇਹਨਾਂ ਵਿੱਚੋਂ ਕੁੱਝ ਟੂਲਸ ਬਾਰੇ ਪਹਿਲਾਂ ਹੀ ਅਧਿਐਨ ਕਰ ਚੁੱਕੇ ਹਾਂ। ਆਓ ਹੁਣ ਕੁੱਝ ਹੋਰ ਉਪਯੋਗੀ ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮਾਂ ਅਤੇ ਟੂਲਸ ਦਾ ਅਧਿਐਨ ਕਰੀਏ:

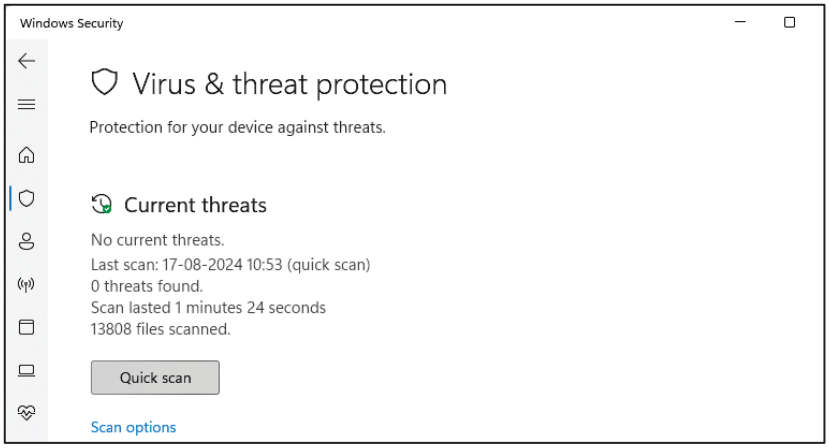
6.5.1 ਐਂਟੀਵਾਇਰਸ (Antivirus): ਵਾਇਰਸ ਇੱਕ ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰ ਹੈ ਜੋ ਸਿਸਟਮ ਦੇ ਕੰਮ ਕਰਨ ਦੀ ਸਪੀਡ ਨੂੰ ਰੋਲੀ ਅਤੇ ਸਿਸਟਮ ਨੂੰ ਖਰਾਬ ਕਰ ਦਿੰਦਾ ਹੈ। ਐਂਟਾਵਾਇਰਸ ਇਕ ਯੂਟੀਲਿਟੀ ਸਾਫਟਵੇਅਰ ਹੈ ਜੋ ਸਾਡੇ ਕੰਪਿਊਟਰ ਨੂੰ ਵਾਇਰਸ-ਮੁਕਤ ਰੱਖਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦਾ ਹੈ। ਇਹ ਉਸ ਸਮੇਂ ਸਾਨੂੰ ਆਪਣੇ ਆਪ ਸੂਚਿਤ ਕਰਦਾ ਹੈ, ਜਦੋਂ ਇਸਨੂੰ ਕੰਪਿਊਟਰ ਵਿਚ ਕਿਸੇ ਵੀ ਖਤਰਨਾਕ ਫਾਈਲ ਦਾ ਪਤਾ ਲੱਗਦਾ ਹੈ ਅਤੇ ਇਹ ਆਟੋਮੈਟਿਕਲੀ ਅਜਿਹੀਆਂ ਫਾਈਲਾਂ ਨੂੰ ਹਟਾ ਦਿੰਦਾ ਹੈ। ਐਂਟੀਵਾਇਰਸ ਸਾਫਟਵੇਅਰ ਦੀਆਂ ਉਦਾਹਰਨਾਂ ਹਨ: McAfee Antivirus, Avast, Avira, Quick heal, Windows Defender ਆਦਿ। ਵਿੰਡੋਜ਼ ਡਿਫੈਂਡਰ ਇੱਕ ਬਿਲਟ-ਇਨ ਐਂਟੀਵਾਇਰਸ ਅਤੇ ਐਂਟੀਮਾਲਵੇਅਰ ਹੈ, ਜੋ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮਾਂ ਲਈ ਮਾਈਕਰੋਸਾਫਟ ਦੁਆਰਾ ਪ੍ਰਦਾਨ ਕੀਤਾ ਗਿਆ ਹੈ। ਇਹ ਸਿਕਿਉਰਿਟੀ ਐਪ ਕੰਪਿਊਟਰਾਂ ਨੂੰ ਕਈ ਤਰ੍ਹਾਂ ਦੇ ਖਤਰਿਆਂ ਤੋਂ ਬਚਾਉਂਦੀ ਹੈ, ਜਿਵੇਂ ਕਿ: ਵਾਇਰਸ, ਸਪਾਈਵੇਅਰ, ਰੈਨਸਮਵੇਅਰ, ਅਤੇ ਹੋਰ ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰਜ਼। ਵਿੰਡੋਜ਼ ਡਿਫੈਂਡਰ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਸਿਸਟਮ ਨੂੰ ਸਕੈਨ ਕਰਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ:

1. ਰਨ (Run) ਵਿੰਡੋ ਨੂੰ ਖੋਲ੍ਹਣ ਲਈ ਕੀਬੋਰਡ ਤੋਂ Win + R ਦਬਾਓ।
2. "windowsdefender:" ਟਾਈਪ ਕਰੋ ਅਤੇ Enter ਬਟਨ ਦਬਾਓ ਜਾਂ OK ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ, ਇਹ ਹੇਟਾਂ ਦਿੱਤੀ "Windows Security" ਵਿੰਡੋ ਦਿਖਾਏਗਾ।



ਚਿੱਤਰ 6.17: ਵਿੰਡੋਜ਼ ਸਿਕਿਉਰਿਟੀ

3. ਵਾਇਰਸ ਅਤੇ ਥਰੈਟ ਪ੍ਰੋਟੈਕਸ਼ਨ ਆਪਸ਼ਨ ਦੀ ਚੋਣ ਕਰੋ, ਇਹ ਹੇਠਾਂ ਅਨੁਸਾਰ ਵਿੰਡੋ ਦਿਖਾਏਗਾ:



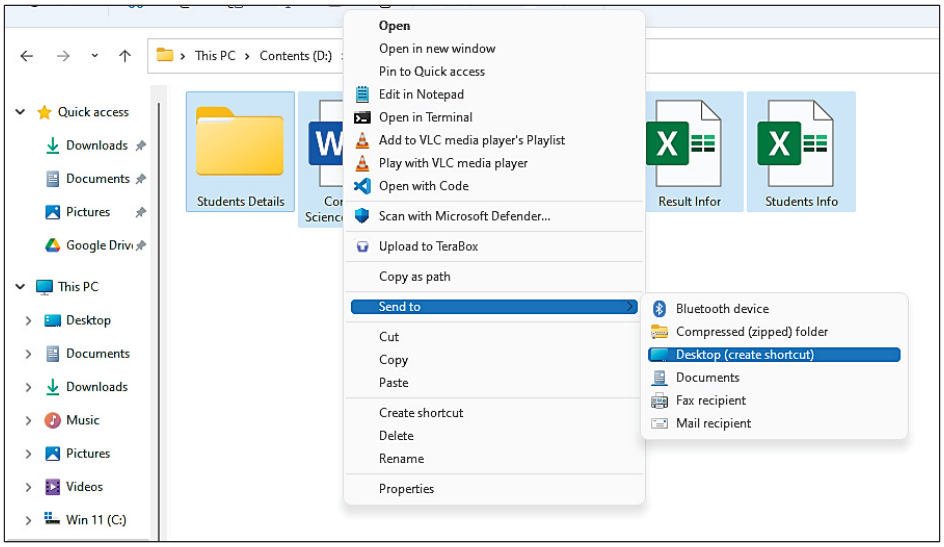
ਚਿੱਤਰ 6.18: ਵਿੰਡੋਜ਼ ਡਿਫੈਂਡਰ ਦੀ ਸਕੈਨਿੰਗ ਆਪਸ਼ਨ

4. ਖਤਰਨਾਕ ਫਾਈਲਾਂ ਨੂੰ ਸਕੈਨ ਕਰਨ ਲਈ ਕੁਵਿਕ ਸਕੈਨ (Quick Scan) ਤੇ ਕਲਿੱਕ ਕਰੋ। ਸਕੈਨ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਜੇਕਰ ਕੋਈ ਇਨਫੈਕਟਿਡ ਫਾਈਲਾਂ ਹੋਣ ਤਾਂ ਇਹ ਉਹਨਾਂ ਦੀ ਸੂਚੀ ਦਿਖਾਉਂਦਾ ਹੈ। ਜੇਕਰ ਇਨਫੈਕਟਿਡ ਫਾਈਲਾਂ ਦਾ ਪਤਾ ਲਗ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਉਹਨਾਂ ਨੂੰ ਹਟਾਉਣ ਲਈ ਦਿੱਤੇ ਨਿਰਦੇਸ਼ਾਂ ਅਨੁਸਾਰ ਅੱਗੇ ਵਧੋ।

6.5.2 ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ ਟੂਲਜ਼ (File Compression tools): ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ ਨੂੰ ਫਾਈਲ ਜ਼ਿਪਿੰਗ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਘੱਟ ਡਿਸਕ ਸਪੇਸ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਇੱਕ ਫਾਈਲ (ਜਾਂ ਫਾਈਲਾਂ) ਨੂੰ ਪੈਕ ਕਰਨ ਦੀ ਪ੍ਰਕਿਰਿਆ ਹੈ। ਕੰਪਰੈਸ਼ਨ ਸਾਫਟਵੇਅਰ ਸਾਨੂੰ ਬਹੁਤ ਸਾਰੀਆਂ ਫਾਈਲਾਂ ਲੈਣ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਇੱਕ ਸਿੰਗਲ ਫਾਈਲ ਵਿੱਚ ਕੰਪਰੈੱਸ (compress) ਕਰਨ ਦੀ ਇਜਾਜ਼ਤ ਦਿੰਦਾ ਹੈ, ਜੋ ਕਿ ਅਸਲ ਫਾਈਲਾਂ ਦੇ ਸੰਯੁਕਤ ਆਕਾਰ (combined size) ਤੋਂ ਛੋਟੀ ਹੋ ਜਾਂਦੀ ਹੈ। ਕੰਪਰੈੱਸਡ ਫ਼ਾਈਲਾਂ ਘੱਟ ਸਟੋਰੇਜ ਸਪੇਸ ਲੈਂਦੀਆਂ ਹਨ। ਉਹਨਾਂ ਨੂੰ ਅਨਕੰਪਰੈੱਸਡ ਫਾਈਲਾਂ ਨਾਲੋਂ ਵਧੇਰੇ ਤੇਜ਼ੀ ਨਾਲ ਦੂਜੇ ਕੰਪਿਊਟਰਾਂ ਵਿੱਚ ਟ੍ਰਾਂਸਫਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਕੰਪਰੈੱਸਡ ਫਾਈਲਾਂ ਨੂੰ ਈਮੇਲਾਂ ਰਾਹੀਂ ਸ਼ੇਅਰ ਕਰਨਾ ਵੀ ਆਸਾਨ ਹੋ ਜਾਂਦਾ ਹੈ। ਮਾਰਕੀਟ ਵਿੱਚ ਬਹੁਤ ਸਾਰੇ ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ ਸਾਫਟਵੇਅਰ ਉਪਲਬਧ ਹਨ: WinZip, WinRAR, 7-zip ਆਮ ਤੌਰ ਤੇ ਵਰਤੇ ਜਾਂਦੇ ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ ਟੂਲਜ਼ ਹਨ। ਲਗਭਗ ਹਰ ਕੰਪਿਊਟਰ ਵਿੱਚ ਇੱਕ ਇਨਬਿਲਟ ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ ਸਹੂਲਤ ਹੁੰਦੀ ਹੈ। ਵਿੰਡੋਜ਼ OS ਵਿੱਚ ਡੀ-ਕੰਪ੍ਰੈਸ (ਅਨਜ਼ਿਪ) ਕਰਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ:

ਵਿੰਡੋ ਦੀ ਜ਼ਿਪ ਯੂਟੀਲਿਟੀ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਫਾਈਲ ਕੰਪਰੈਸ਼ਨ (ਜ਼ਿਪਿੰਗ) ਲਈ ਸਟੈਪ:

1. ਉਸ ਫੋਲਡਰ ਜਾਂ ਫਾਈਲ ਤੇ ਮਾਊਸ ਦਾ ਸੱਜਾ ਕਲਿੱਕ (Right click) ਕਰੋ, ਜਿਨ੍ਹਾਂ ਨੂੰ ਅਸੀਂ ਜ਼ਿਪ ਜਾਂ ਕੰਪਰੈੱਸ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।
2. ਪੌਪਅੱਪ ਮੀਨੂੰ (popup menu) ਵਿਚੋਂ Send To –> Compressed (zipped) folder ਆਪਸ਼ਨ ਚੁਣੋ।



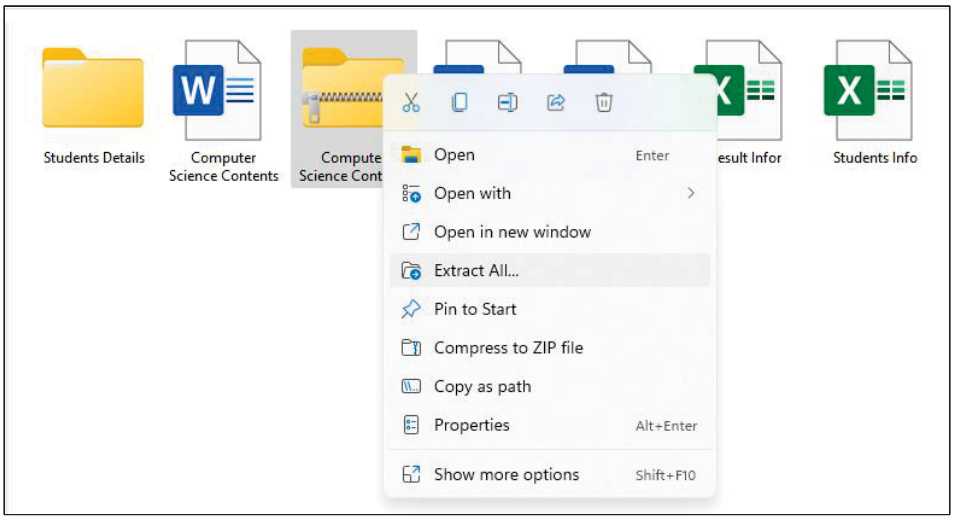
ਚਿੱਤਰ 6.19: ਵਿੰਡੋ ਦੀ ਜ਼ਿਪ ਯੂਟੀਲਿਟੀ

3. ਉਸੇ ਲੋਕੇਸ਼ਨ ਉਪਰ ਇੱਕ ਨਵਾਂ ਜ਼ਿਪਡ ਫੋਲਡਰ ਬਣ ਜਾਵੇਗਾ।

ਵਿੰਡੋਜ਼ ਵਿੱਚ ਅਸੀਂ ਜ਼ਿਪ ਕੀਤੀਆਂ ਫਾਈਲਾਂ ਅਤੇ ਫੋਲਡਰਾਂ ਨਾਲ ਉਸੇ ਤਰ੍ਹਾਂ ਕੰਮ ਕਰ ਸਕਦੇ ਹਾਂ ਜਿਵੇਂ ਅਸੀਂ ਅਨਕੰਪਰੈਂਸਡ ਫਾਈਲਾਂ ਅਤੇ ਫੋਲਡਰਾਂ ਨਾਲ ਕੰਮ ਕਰਦੇ ਹਾਂ।

ਵਿੰਡੋ ਦੀ ਜ਼ਿਪ (Zip) ਯੂਟਿਲਟੀ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਫਾਈਲਾਂ ਨੂੰ ਡੀਕੰਪਰੈਸ (ਅਨਜ਼ਿਪ) ਕਰਨਾ:

1. ਉਸ ਜ਼ਿਪ ਕੀਤੇ ਫੋਲਡਰ ਤੇ ਮਾਊਸ ਦਾ ਸੱਜਾ ਕਲਿੱਕ ਕਰੋ ਜਿਸ ਨੂੰ ਅਸੀਂ ਅਨਜ਼ਿਪ (ਐਬਸਟਰੈਕਟ) ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।
2. **Extract All** ਆਪਸ਼ਨ ਦੀ ਚੌਣ ਕਰੋ ਅਤੇ ਫਿਰ ਨਿਰਦੇਸ਼ਾਂ ਦੀ ਪਾਲਣਾ ਕਰੋ।



ਚਿੱਤਰ 6.20: ਅਨਜ਼ਿਪ ਦੀ ਵਰਤੋ ਕਰਦੇ ਹੋਏ

6.5.3. ਫਾਇਲ ਮੈਨੇਜਮੈਂਟ ਸਿਸਟਮ (File Management System): ਇਹ ਯੂਟੀਲਿਟੀ ਸਾਫਟਵੇਅਰ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀਆਂ ਫਾਈਲਾਂ ਨੂੰ ਮੈਨੇਜ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਸਾਫਟਵੇਅਰ ਸਿਸਟਮ ਦੀਆਂ ਫਾਈਲਾਂ ਨੂੰ ਬ੍ਰਾਊਜ਼ ਕਰਨ, ਵਿਵਸਥਿਤ ਕਰਨ, ਲੱਭਣ ਅਤੇ ਪ੍ਰੀਵਿਊ ਕਰਨ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੇ ਹਨ। ਵਿੰਡੋਜ਼ ਐਕਸਪਲੋਰਰ (Windows Explorer) ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦਾ ਡਿਫਾਲਟ ਫਾਈਲ ਮੈਨੇਜਮੈਂਟ ਟੂਲ ਹੈ।

6.5.4 ਬੈਕਅੱਪ ਅਤੇ ਰੀਸਟੋਰ (Backup and Restore): ਬੈਕਅੱਪ, ਅਸਲ ਫਾਈਲ ਦੀ ਇੱਕ ਕਾਪੀ ਜਾਂ ਮਲਟੀਪਲ ਫਾਈਲਾਂ ਦਾ ਇੱਕ ਸੈੱਟ ਹੁੰਦਾ ਹੈ, ਜੋ ਔਰਿਜਨਲ ਤੋਂ ਇੱਕ ਵੱਖਰੀ ਲੋਕੇਸ਼ਨ ਤੇ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਇੱਕ DVD, ਇੱਕ ਐਕਸਟਰਨਲ ਡਰਾਈਵ ਜਾਂ ਇੰਟਰਨੈੱਟ ਤੇ ਕਿਸੇ ਹੋਰ ਥਾਂ ਤੇ। **ਬੈਕਅੱਪ** ਦਾ ਮੁੱਖ ਟੀਚਾ ਡਾਟਾ ਦੀ ਇੱਕ ਕਾਪੀ ਤਿਆਰ ਕਰਨਾ ਹੈ, ਜੋ ਪ੍ਰਾਇਮਰੀ ਡਾਟਾ ਫੇਲ ਹੋਣ ਤੇ ਮੁੜ ਪ੍ਰਾਪਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਡਾਟਾ ਰਿਕਵਰੀ ਦੀ ਲੋੜ ਉਦੋਂ ਹੁੰਦੀ ਹੈ ਜਦੋਂ ਡਾਟਾ ਖਰਾਬ ਹੋ ਜਾਂਦਾ ਹੈ, ਇਹ ਅਕਸਰ **ਰੀਸਟੋਰ** ਵਜੋਂ ਜਾਣੀ ਜਾਂਦੀ ਹੈ। ਡਾਟਾ ਖਰਾਬ ਹੋਣ ਤੋਂ ਬਾਅਦ ਡਾਟਾ ਰਿਕਵਰ ਕਰਨ ਦੀ ਪ੍ਰਕਿਰਿਆ ਨੂੰ **ਡਾਟਾ ਰਿਕਵਰੀ** ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਬੈਕਅੱਪ ਅਤੇ ਰਿਕਵਰੀ ਸਾਫਟਵੇਅਰ ਦੀਆਂ ਕਈ ਕਿਸਮਾਂ ਮਾਰਕੀਟ ਵਿੱਚ ਉਪਲਬਧ ਹਨ। ਅਸੀਂ ਆਪਣੇ ਮਹੱਤਵਪੂਰਨ ਡਾਟਾ ਦੇ ਬੈਕਅਪ ਅਤੇ ਰਿਕਵਰੀ ਦੇ ਉਦੇਸ਼ ਲਈ ਕਿਸੇ ਵੀ ਸਾਫਟਵੇਅਰ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੇ ਹਾਂ।

6.5.5 ਡਿਸਕ ਮੈਨੇਜਮੈਂਟ ਟੂਲਸ (Disk Management Tools): ਇਹ ਯੂਟੀਲਿਟੀ ਸਾਫਟਵੇਅਰ ਡਿਸਕਾਂ ਤੇ ਡਾਟਾ ਨੂੰ ਮੈਨੇਜ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਹਨਾਂ ਟੂਲਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ, ਅਸੀਂ ਡਿਸਕ ਡਰਾਈਵਾਂ ਦਾ ਪਾਰਟੀਸ਼ਨ, ਅਤੇ ਪਾਰਟੀਸ਼ਨ ਨੂੰ ਮੁੜ ਆਕਾਰ ਦੇਣ ਆਦਿ ਵਰਗੇ ਫੰਕਸ਼ਨ ਲਾਗੂ ਕਰ ਸਕਦੇ ਹਾਂ। ਡਿਸਕ ਮੈਨੇਜਮੈਂਟ ਟੂਲਸ ਦੀਆਂ ਉਦਾਹਰਣਾਂ ਹਨ: ਮਿਨੀਟੂਲ ਪਾਰਟੀਸ਼ਨ ਵਿਜ਼ਾਰਡ, ਪੋਰਾਗਾਨ ਪਾਰਟੀਸ਼ਨ ਮੈਨੇਜਰ ਆਦਿ।

6.6 ਸਾਫਟਵੇਅਰ ਅੱਪਡੇਟ ਅਤੇ ਅੱਪਗਰੇਡ (SOFTWARE UPDATE AND UPGRADE)

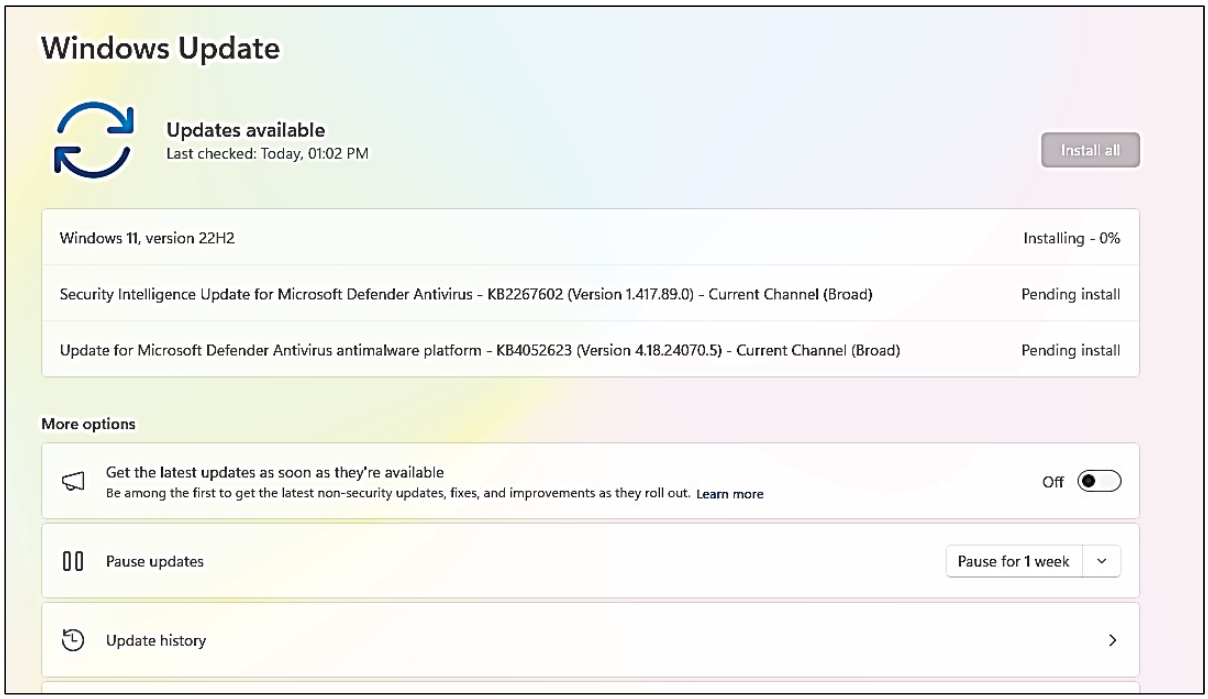
ਅੱਪਡੇਟ ਅਤੇ ਅੱਪਗ੍ਰੇਡ ਇੱਕ ਐਪ ਜਾਂ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਬਦਲਾਅ ਕਰਨ ਦੇ ਦੋ ਵੱਖ-ਵੱਖ ਤਰੀਕੇ ਹਨ। ਪਰ ਮੁੱਖ ਅੰਤਰ ਬਹੁਤ ਸਾਰੀਆਂ ਸੋਧਾਂ ਅਤੇ ਉਹਨਾਂ ਸੋਧਾਂ ਦੀ ਮਹੱਤਤਾ ਵਿੱਚ ਹੈ। ਇੱਕ ਸਾਫਟਵੇਅਰ ਅੱਪਡੇਟ ਵਿੱਚ ਬੱਗ ਫਿਕਸ ਅਤੇ ਹੋਰ ਛੋਟੇ ਸੁਧਾਰ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ, ਜਦੋਂ ਕਿ ਇੱਕ ਸਾਫਟਵੇਅਰ ਅੱਪਗਰੇਡ ਸਾਫਟਵੇਅਰ ਦੇ ਵਰਜਨ ਨੂੰ ਬਦਲਦਾ ਹੈ।

- **ਸਾਫਟਵੇਅਰ ਅੱਪਡੇਟ (Software Update):** ਇੱਕ ਅੱਪਡੇਟ ਇੱਕ ਪੈਚ ਹੁੰਦਾ ਹੈ ਜੋ ਅਕਸਰ ਉਤਪਾਦ ਦੇ ਰਿਲੀਜ਼ ਹੋਣ ਤੋਂ ਬਾਅਦ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਹੱਲ ਕਰਨ ਲਈ ਉਪਲਬਧ ਕਰਵਾਇਆ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਅਸੀਂ ਕੋਈ ਅੱਪਡੇਟ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਇਹ ਕਿਸੇ ਐਪ ਜਾਂ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਇਸ ਤਰ੍ਹਾਂ ਬਦਲਾਅ ਕਰਦਾ ਹੈ ਕਿ ਇਹ ਇਸਦੇ ਮੂਲ ਢਾਂਚੇ ਨੂੰ ਪ੍ਰਭਾਵਿਤ ਨਹੀਂ ਕਰਦਾ। ਇਸ ਲਈ ਸਾਡੇ ਕੰਪਿਊਟਰ ਵਿੱਚ ਅਕਸਰ ਕੀਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਤਬਦੀਲੀਆਂ ਜਿਵੇਂ ਕਿ ਬੱਗ ਫਿਕਸ, ਸੁਰੱਖਿਆ ਪੈਚ, ਨਵੇਂ ਹਾਰਡਵੇਅਰ ਲਈ ਸਮਰਥਨ ਸ਼ਾਮਲ ਕਰਨਾ, ਆਦਿ ਨੂੰ ਅੱਪਡੇਟ ਕਿਹਾ ਜਾ ਸਕਦਾ ਹੈ। ਇੱਕ ਅੱਪਡੇਟ ਆਕਾਰ ਵਿੱਚ ਛੋਟਾ ਹੁੰਦਾ ਹੈ, ਅਤੇ ਇਸਨੂੰ ਕਰਨ ਵਿੱਚ ਕੁਝ ਮਿੰਟ ਲੱਗ ਸਕਦੇ ਹਨ। ਅੱਪਡੇਟ ਅਕਸਰ ਮੁਫ਼ਤ ਹੁੰਦੇ ਹਨ ਅਤੇ ਉਹ ਜ਼ਰੂਰੀ ਹੁੰਦੇ ਹਨ। ਸਾਡੀਆਂ ਐਪਲੀਕੇਸ਼ਨਾਂ ਨੂੰ ਸੁਰੱਖਿਅਤ ਅਤੇ ਕੁਸ਼ਲਤਾ ਨਾਲ ਚਲਾਉਣ ਲਈ, ਸਾਨੂੰ ਮਹੀਨੇ ਵਿੱਚ ਘੱਟੋ-ਘੱਟ ਇੱਕ ਵਾਰ ਕੰਪਿਊਟਰ ਅੱਪਡੇਟ ਦੀ ਜਾਂਚ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ। ਅਸੀਂ ਇਹਨਾਂ ਅੱਪਡੇਟ ਸੈਟਿੰਗਾਂ ਨੂੰ ਵੀ ਵਿਵਸਥਿਤ ਕਰ ਸਕਦੇ ਹਾਂ ਤਾਂ ਕਿ ਅੱਪਡੇਟ ਆਟੋਮੈਟਿਕਲੀ ਹੋਣ।
- **ਸਾਫਟਵੇਅਰ ਅੱਪਗਰੇਡ (Software Upgrade):** ਇੱਕ ਅੱਪਗਰੇਡ ਪੁਰਾਣੇ ਵਰਜਨ ਨੂੰ ਇੱਕ ਨਵੇਂ ਵਰਜਨ ਨਾਲ ਬਦਲਣਾ ਹੈ। ਜਦੋਂ ਸਾਫਟਵੇਅਰ ਵਿੱਚ ਕੀਤੀਆਂ ਤਬਦੀਲੀਆਂ ਦਾ ਇੱਕ ਸੈੱਟ ਮਹੱਤਵਪੂਰਨ ਹੁੰਦਾ ਹੈ, ਤਾਂ ਅਸੀਂ ਇਸਨੂੰ ਅੱਪਗ੍ਰੇਡ ਕਹਿੰਦੇ ਹਾਂ। ਉਦਾਹਰਨ ਲਈ: ਵਿੰਡੋਜ਼ 10 ਤੋਂ ਵਿੰਡੋਜ਼ 11 ਵਿੱਚ ਬਦਲਣ ਨੂੰ ਅੱਪਗਰੇਡ ਕਿਹਾ ਜਾਵੇਗਾ, ਇਹ ਅੱਪਡੇਟ ਨਹੀਂ ਅਖਵਾਏਗਾ। ਇੱਕ ਅੱਪਗਰੇਡ ਵਿੱਚ ਜ਼ਿਆਦਾਤਰ GUI ਵਿੱਚ ਮਹੱਤਵਪੂਰਨ ਤਬਦੀਲੀਆਂ ਅਤੇ ਕਈ ਤਰ੍ਹਾਂ ਦੀਆਂ ਨਵੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਅਤੇ ਵਿਕਲਪ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ, ਜੋ ਕਿਸੇ ਸਾਫਟਵੇਅਰ ਜਾਂ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੇ ਮੌਜੂਦਾ ਵਰਜਨ ਵਿੱਚ ਨਹੀਂ ਹੁੰਦੇ। ਇਸ ਦਾ ਆਕਾਰ ਕਈ ਗੀਗਾਬਾਈਟ ਤੱਕ ਜਾ ਸਕਦਾ ਹੈ। ਅਪਗਰੇਡਸ ਅਕਸਰ ਮਹਿੰਗੇ ਹੁੰਦੇ ਹਨ ਅਤੇ ਇਹਨਾਂ ਨੂੰ ਲਾਗੂ ਕਰਨਾ ਅਕਸਰ ਜ਼ਰੂਰੀ ਵੀ ਨਹੀਂ ਹੁੰਦਾ।

6.6.1 ਵਿੰਡੋ ਨੂੰ ਕਿਵੇਂ ਅਪਡੇਟ ਕਰਨਾ ਹੈ (How to Update Window):

ਜੇਕਰ ਅਸੀਂ ਆਪਣੇ PC ਤੋਂ ਵੱਧ ਤੋਂ ਵੱਧ ਲਾਭ ਉਠਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਇਸ ਦਾ ਸਭ ਤੋਂ ਵਧੀਆ ਤਰੀਕਾ ਇਹ ਹੋਵੇਗਾ ਕਿ ਸਾਡੇ ਕੋਲ ਵਿੰਡੋਜ਼ ਦਾ ਨਵੀਨਤਮ ਵਰਜਨ ਹੋਵੇ। ਜਦੋਂ ਅਸੀਂ ਅੱਪਡੇਟ ਦੀ ਜਾਂਚ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਸਾਡਾ PC ਨਵੀਨਤਮ ਡਿਵਾਈਸ ਡ੍ਰਾਈਵਰਾਂ ਦੀ ਭਾਲ ਵੀ ਕਰਦਾ ਹੈ, ਜੋ ਸਾਡੇ PC ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਬਿਹਤਰ ਬਣਾਉਣ ਵਿੱਚ ਵੀ ਮਦਦ ਕਰ ਸਕਦਾ ਹੈ। ਅਸੀਂ ਇਸ ਅਧਿਆਇ ਦੇ ਅਗਲੇ ਭਾਗਾਂ ਵਿੱਚ ਡਿਵਾਈਸ ਡਰਾਈਵਰਾਂ ਬਾਰੇ ਚਰਚਾ ਕਰਾਂਗੇ। ਵਿੰਡੋਜ਼ ਅਪਡੇਟਾਂ ਦੀ ਜਾਂਚ ਕਰਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਸਟੈਪਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ:

1. **Start –> Settings –> Windows Update –> Check for updates** ਤੇ ਕਲਿੱਕ ਕਰੋ।
2. ਇਹਨਾਂ ਵਿੱਚੋਂ ਇੱਕ ਕਰੋ:
   - ਜੇਕਰ ਸਟੇਟਸ ਕਹਿੰਦਾ ਹੈ "ਤੁਸੀਂ ਅੱਪ ਟੂ ਡੇਟ ਹੈ", ਤਾਂ ਜਾਰੀ ਰੱਖੋ ਅਤੇ ਆਪਸ਼ਨਲ ਅੱਪਡੇਟਾਂ ਦੀ ਜਾਂਚ ਕਰੋ।
   - ਜੇਕਰ ਸਟੇਟਸ ਕਹਿੰਦੇ ਹਾ "ਅੱਪਡੇਟ ਉਪਲਬਧ ਹਨ," ਤਾਂ ਉਹਨਾਂ ਨੂੰ ਡਾਊਨਲੋਡ ਕਰਨ ਲਈ ਕਲਿੱਕ ਕਰੋ.
3. ਉਹ ਅੱਪਡੇਟ ਚੁਣੋ ਜੋ ਤੁਸੀਂ ਇੰਸਟਾਲ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹੋ, ਫਿਰ ਇੰਸਟਾਲ ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।
4. ਆਪਣੇ PC ਨੂੰ ਰੀ-ਸਟਾਰਟ ਕਰੋ।

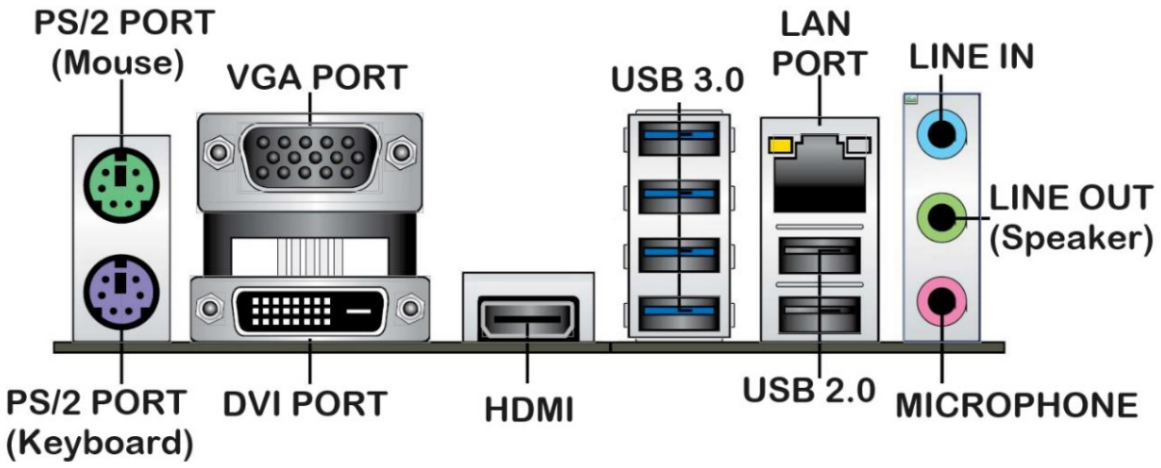


ਚਿੱਤਰ 6.21: ਵਿੰਡੋਜ ਅਪਡੇਟ ਸਕਰੀਨ

6.7 ਪੋਰਟਸ ਅਤੇ ਉਹਨਾਂ ਦੀਆਂ ਕਿਸਮਾਂ (PORTS & THEIR TYPES)

ਕੰਪਿਊਟਰ ਹਾਰਡਵੇਅਰ ਅਨੁਸਾਰ ਇੱਕ ਪੋਰਟ ਕੰਪਿਊਟਰ ਅਤੇ ਪੈਰੀਫਿਰਲ ਡਿਵਾਈਸਾਂ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਵਜੋਂ ਕੰਮ ਕਰਦੀ ਹੈ। ਅਸੀਂ ਇਹਨਾਂ ਪੋਰਟਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨਾਲ ਕਈ ਹਾਰਡਵੇਅਰ ਕੰਪੋਨੈਂਟ ਜਿਵੇਂ ਕਿ ਮਾਨੀਟਰ, ਵੈਬਕੈਮ, ਸਪੀਕਰ ਆਦਿ ਨੂੰ ਜੋੜ ਸਕਦੇ ਹਾਂ।

ਜ਼ਿਆਦਾਤਰ ਪੋਰਟਸ ਸਿਸਟਮ ਯੂਨਿਟ ਦੇ ਪਿਛਲੇ ਪਾਸੇ ਉਪਲਬਧ ਹੁੰਦੀਆਂ ਹਨ (ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ)। ਇਹ ਪੋਰਟਸ ਵਿਸ਼ੇਸ਼ ਡਿਵਾਈਸਾਂ ਨੂੰ ਅਟੈਚ ਕਰਨ ਲਈ ਬਣਾਈਆਂ ਗਈਆਂ ਹੁੰਦੀਆਂ ਹਨ। ਇਹਨਾਂ ਪੋਰਟਸ ਦੀ ਪਲੇਸਮੈਂਟ ਕੰਪਿਊਟਰ ਤੋਂ ਕੰਪਿਊਟਰ ਤੱਕ ਵੱਖਰੀ ਹੁੰਦੀ ਹੈ। ਕੁੱਝ ਪੋਰਟਸ ਕਲਰ-ਕੋਡਿਡ ਹੋ ਸਕਦੀਆਂ ਹਨ। ਕਲਰ-ਕੋਡ ਇਹ ਨਿਰਦਾਰਤ ਕਰਨ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੇ ਹਨ ਕਿ ਕਿਸੇ ਖਾਸ ਡਿਵਾਈਸ ਨਾਲ ਕਿਹੜੀ ਪੋਰਟ ਵਰਤੀ ਜਾਣੀ ਹੈ।



ਚਿੱਤਰ 6.22: ਪੋਰਟਸ ਅਤੇ ਉਹਨਾਂ ਦੀਆਂ ਕਿਸਮਾਂ

ਕੰਪਿਊਟਰ ਪੋਰਟਸ ਦੇ ਵੱਖ-ਵੱਖ ਫੰਕਸ਼ਨ ਅਤੇ ਡਿਜ਼ਾਈਨ ਕਨੈਕਟਰ ਹੁੰਦੇ ਹਨ। ਪੋਰਟ ਅਸਲ ਵਿੱਚ ਮਦਰਬੋਰਡ ਦੇ ਸਲਾਟ ਹੁੰਦੇ ਹਨ ਜਿਸ ਵਿੱਚ ਬਾਹਰੀ ਡਿਵਾਈਸ ਦੀ ਇੱਕ ਕੇਬਲ ਲਗਾਈ ਜਾਂਦੀ ਹੈ। ਆਓ ਹੁਣ ਕੰਪਿਊਟਰਾਂ ਵਿੱਚ ਉਪਲਬਧ ਕੁੱਝ ਮਹੱਤਵਪੂਰਨ ਕਿਸਮਾਂ ਦੀਆਂ ਪੋਰਟਸ ਬਾਰੇ ਚਰਚਾ ਕਰੀਏ:

- **PS/2 ਪੋਰਟ:** ਇਹ ਪੋਰਟ ਪੁਰਾਣੇ ਕਿਸਮ ਦੇ ਕੰਪਿਊਟਰ ਕੀਬੋਰਡ ਅਤੇ ਮਾਊਸ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਜ਼ਿਆਦਾਤਰ ਪੁਰਾਣੇ ਕੰਪਿਊਚਰ ਦੋ PS/2 ਪੋਰਟਸ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ: ਇੱਕ ਮਾਊਸ ਲਈ ਅਤੇ ਦੂਜਾ ਕੀਬੋਰਡ ਲਈ। ਮਾਊਸ ਪੋਰਟ ਹਰੇ ਰੰਗ ਦੇ ਕੋਡ ਵਿੱਚ ਆਉਂਦਾ ਹੈ ਅਤੇ ਕੀਬੋਰਡ ਪੋਰਟ ਮੇਜੈਂਟਾ ਰੰਗ ਵਿੱਚ ਆਉਂਦਾ ਹੈ। ਕਲਰ-ਕੋਡ ਪਛਾਣ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।


- **ਯੂਨੀਵਰਸਲ ਸੀਰੀਅਲ ਬਸ (Universal Serial Bus (USB)) ਪੋਰਟ:** ਇਹ ਇੱਕ ਬਹੁਤ ਹੀ ਪ੍ਰਸਿੱਧ ਅਤੇ ਬਹੁਮੁਖੀ ਕਿਸਮ ਦਾ ਪੋਰਟ ਹੈ। ਇਹ ਬਾਹਰੀ ਹਾਰਡ ਡਿਸਕ, ਪ੍ਰਿੰਟਰ, ਸਕੈਨਰ, ਮਾਊਸ, ਕੀਬੋਰਡ, ਆਦਿ ਵਰਗੇ ਬਾਹਰੀ USB ਡਿਵਾਈਸਾਂ ਦੀਆਂ ਸਾਰੀਆਂ ਕਿਸਮਾਂ ਨੂੰ ਜੋੜ ਸਕਦਾ ਹੈ। ਇਹ ਪੋਰਟ 1997 ਵਿੱਚ ਪੇਸ਼ ਕੀਤੀ ਗਈ ਸੀ। ਜ਼ਿਆਦਾਤਰ ਕੰਪਿਊਟਰ USB ਪੋਰਟ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ। USB ਪੋਰਟ ਆਮ ਤੌਰ ਤੇ ਦੋ ਕਲਰ-ਕੋਡਜ਼ ਵਿੱਚ ਉਪਲਬਧ ਹੁੰਦੇ ਹਨ: ਬਲੈਕ ਅਤੇ ਬਲੂ ਕਲਰ ਕੋਡ ਵਿਚ। ਬਲੈਕ ਕੋਡੇਡ USB ਨੂੰ USB 2.0 ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜੋ ਕਿ ਬਲੂ USB ਕੋਡੇਡ ਪੋਰਟ ਨਾਲੋਂ ਹੌਲੀ ਹੁੰਦਾ ਹੈ। ਬਲੂ ਕੋਡੇਡ USB ਪੋਰਟ ਦੱਸਦਾ ਹੈ ਕਿ ਪੋਰਟ USB 3.0 ਐਕਸੈਸ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। USB ਪੋਰਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ, ਅਸੀਂ ਇੱਕ ਡਿਵਾਈਸ ਦੀ ਬੈਟਰੀ ਨੂੰ ਵੀ ਚਾਰਜ ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਤੇਜ਼ ਰਫਤਾਰ ਨਾਲ ਡਾਟਾ ਵੀ ਟ੍ਰਾਂਸਫਰ ਕਰ ਸਕਦੇ ਹਾਂ।

- **ਵਿਜ਼ੁਅਲ ਗ੍ਰਾਫਿਕਸ ਐਡਪਟਰ (Visual Graphics Adapter (VGA)) ਪੋਰਟ:** ਇਸਨੂੰ ਮਾਨੀਟਰ ਪੋਰਟ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਮਾਨੀਟਰ ਨੂੰ ਕੰਪਿਊਟਰ ਦੇ ਵੀਡੀਓ ਕਾਰਡ ਨਾਲ ਜੋੜਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। VGA ਇੱਕ ਪੁਰਾਣਾ ਐਨਾਲਾਗ ਕਨੈਕਟਰ ਹੈ। ਇਹ DVI, HDMI ਪੋਰਟਾਂ ਤੋਂ ਪਹਿਲਾਂ ਵਿਆਪਕ ਤੌਰ ਤੇ ਵਰਤਿਆ ਜਾਂਦਾ ਸੀ।

- **ਹਾਈ-ਡੈਫੀਨੇਸ਼ਨ ਮਲਟੀਮੀਡੀਆ ਇੰਟਰਫੇਸ (High Definition Multimedia Interface (HDMI)) ਪੋਰਟ:** HDMI ਪੋਰਟਸ HDMI ਕੇਬਲਾਂ ਤੋਂ ਕਨੈਕਸ਼ਨ ਪ੍ਰਾਪਤ ਕਰਦੀਆਂ ਹਨ। ਇਹਨਾਂ ਪੋਰਟਸ ਦੀ ਵਰਤੋਂ ਹਾਈ-ਡੈਫੀਨੇਸ਼ਨ ਆਡੀਓ ਅਤੇ ਵਿਜ਼ੁਅਲ ਸਿਗਨਲਾਂ ਨੂੰ ਸੰਚਾਰਿਤ ਕਰਨ ਅਤੇ ਪਰਾਪਤ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਕੰਪਿਊਟਰ, ਟੈਲੀਵਿਜ਼ਨ ਅਤੇ ਹੋਰ ਮਲਟੀਮੀਡੀਆ ਡਿਵਾਈਸਾਂ ਵਿੱਚ ਅਕਸਰ HDMI ਪੋਰਟਸ ਹੁੰਦੇ ਹਨ। ਬਹੁਤ ਸਾਰੇ ਆਧੁਨਿਕ ਮਾਨੀਟਰ ਉੱਚ ਗੁਣਵੱਤਾ ਵਿੱਚ ਵਿਜ਼ੁਅਲ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ HDMIs ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਨ। ਇਹ ਪੋਰਟਸ ਜ਼ਿਆਦਾਤਰ ਲੈਪਟਾਪ ਵਿੱਚ ਪ੍ਰੋਜੈਕਟਰਾਂ ਅਤੇ ਵੱਡੇ ਆਕਾਰ ਦੀਆਂ LCD/LED ਸਕ੍ਰੀਨਾਂ ਨਾਲ ਕੁਨੈਕਸ਼ਨ ਬਨਾਉਣ ਲਈ ਵੀ ਉਪਲਬਧ ਹੁੰਦੇ ਹਨ।

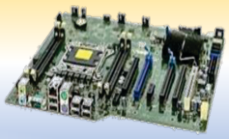
- **ਡਿਜੀਟਲ ਵੀਡੀਓ ਇੰਟਰਫੇਸ (Digital Video Interface (DVI)) ਪੋਰਟ:** ਇਹ ਪੋਰਟ ਫਲੈਟ ਪੈਨਲ LCD ਮਾਨੀਟਰ ਨੂੰ ਕੰਪਿਊਟਰ ਦੇ ਵੀਡੀਓ ਗ੍ਰਾਫਿਕਸ ਕਾਰਡਾਂ ਨਾਲ ਜੋੜਦਾ ਹੈ। ਇਹ VGA ਨਾਲੋਂ ਸ਼ਾਰਪ ਅਤੇ ਬਿਹਤਰ

ਤਸਵੀਰ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਹ ਇੱਕ ਵਿਲੱਖਣ ਕਨੈਕਟਰ ਹੁੰਦਾ ਹੈ ਕਿਉਂਕਿ ਇਹ ਡਿਜੀਟਲ ਅਤੇ ਐਨਾਲਾਗ ਦੋਵੇਂ ਕਿਸਮਾਂ ਦੇ ਸਿਗਨਲ ਪ੍ਰਾਪਤ ਕਰ ਸਕਦਾ ਹੈ।

- **ਈਥਰਨੈੱਟ ਪੋਰਟ (Eternet Port):** ਇਸ ਪੋਰਟ ਨੂੰ LAN ਪੋਰਟ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਹ ਸਾਡੇ PC ਨੂੰ ਇੱਕ ਨੈਟਵਰਕ ਅਤੇ ਹਾਈ-ਸਪੀਡ ਇੰਟਰਨੈਟ ਨਾਲ ਜੋੜਦਾ ਹੈ। ਇੱਕ RJ-45 ਕਨੈਕਟਰ ਦੀ ਵਰਤੋਂ ਨੈੱਟਵਰਕ ਕੇਬਲ ਨੂੰ ਕੰਪਿਊਟਰ ਨਾਲ ਕੁਨੈਕਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਪੋਰਟ ਇੱਕ ਈਥਰਨੈੱਟ ਕਾਰਡ ਤੇ ਉਪਲਬਧ ਹੁੰਦਾ ਹੈ। ਇਸ ਪੋਰਟ ਉਪਰ ਨੈੱਟਵਰਕ ਦੀ ਬੈਂਡਵਿਡਥ ਦੇ ਆਦਾਰ ਤੇ ਡਾਟਾ 10 ਮੈਗਾਬਾਈਟ ਤੋਂ 1000 ਮੈਗਾਬਾਈਟ ਪ੍ਰਤੀ ਸਕਿੰਟ ਤੇ ਯਾਤਰਾ ਕਰਦਾ ਹੈ।

- **ਪਾਵਰ ਕੁਨੈਕਟਰ (Power Connector):** ਉਪਰੋਕਤ ਪੋਰਟਾਂ ਤੋਂ ਇਲਾਵਾ, ਮਦਰਬੋਰਡ ਵਾਪਰ ਕੁਨੈਕਸ਼ਨ ਲਈ ਇੱਕ ਕੁਨੈਕਟਰ ਵੀ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਹ ਸਿਸਟਮ ਯੂਨਿਟ ਦੇ ਪਾਵਰ ਸਪਲਾਈ ਬਾਕਸ (SMPS) ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਕੁਨੈਕਟਰ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨੂੰ ਬਿਜਲੀ ਸਪਲਾਈ ਮੁਹੱਈਆ ਕਰਨ ਲਈ ਵਰਤਿਆ ਗਿਆ ਹੈ, ਇਹ ਤਿੰਨ-ਪੱਖੀ ਪਲੱਗ (three-pronged plug) ਹੈ, ਜੋ ਪਾਵਰ ਕੇਬਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੰਪਿਊਟਰ ਨੂੰ ਪਾਵਰ-ਬਾਰ ਜਾਂ ਵਾਲ ਸੋਕਟ ਨਾਲ ਜੋੜਦਾ ਹੈ।

ਇੱਕ ਮਦਰਬੋਰਡ ਕੰਪਿਊਟਰ ਵਿੱਚ ਪ੍ਰਿੰਟਿਡ ਸਰਕਟ ਬੋਰਡ (PCB) ਹੁੰਦਾ ਹੈ। ਇਹ ਕੰਪਿਊਟਰ ਦਾ ਕੇਂਦਰੀ ਸੰਚਾਰ ਬੈਕਬੋਨ ਕੁਨੈਕਟੀਵਿਟੀ ਪੁਆਇੰਟ ਹੈ, ਜਿਸ ਰਾਹੀਂ ਕੰਪਿਊਟਰ ਦੇ ਸਾਰੇ ਹਿੱਸੇ ਅਤੇ ਬਾਹਰੀ ਪੈਰੀਫਿਰਲ ਜੁੜਦੇ ਹਨ। ਸਾਰੀਆਂ ਪੋਰਟਸ ਇਸ ਮਦਰਬੋਰਡ ਦਾ ਹਿੱਸਾ ਹੁੰਦੀਆਂ ਹਨ।



6.8 ਡਿਵਾਈਸ ਡਰਾਈਵਰ (DEVICE DRIVERS):

ਡਰਾਈਵਰ ਇੱਕ ਅਜਿਹਾ ਸਾਫਟਵੇਅਰ ਹੁੰਦਾ ਹੈ ਜਿਸਦੀ ਵਰਤੋ ਕਿਸੇ ਡਿਵਾਈਸ ਦੁਆਰਾ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨਾਲ ਕੰਮ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਜਦੋਂ ਵੀ ਅਸੀਂ ਕਿਸੇ ਵੀ ਡਿਵਾਈਸ ਨੂੰ ਪੋਰਟਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨਾਲ ਕੁਨੈਕਟ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਕੰਪਿਊਟਰ ਉਸ ਨਾਲ ਕੁਨੈਕਸਨ ਬਨਾਉਣ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰਦਾ ਹੈ। ਕੰਪਿਊਟਰ ਨਵੇਂ ਕੁਨੈਕਟ ਕੀਤੇ ਡਿਵਾਈਸ ਦੇ ਡਰਾਈਵਰਾਂ ਨੂੰ ਇੰਸਟਾਵਲ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ ਕਰਦਾ ਹੈ ਤਾਂ ਜੋ ਡਿਵਾਈਸ ਅਤੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਇੱਕ ਦੂਜੇ ਨਾਲ ਸੰਚਾਰ ਕਰ ਸਕਣ। ਕੁਝ ਡਿਵਾਈਸਾਂ ਲਈ, ਵਿੰਡੋਜ਼ OS ਆਪਣੇ ਆਪ ਡਰਾਈਵਰਾਂ ਨੂੰ ਇੰਸਟਾਲ ਕਰ ਦਿੰਦਾ ਹੈ। ਪਰ, ਕੁਝ ਡਿਵਾਈਸਾਂ ਖਾਸ ਤੌਰ ਤੇ ਬਾਹਰੀ ਡਿਵਾਈਸਾਂ ਲਈ, ਸਾਨੂੰ ਆਪਣੇ ਆਪ ਡਰਾਈਵਰਾਂ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ। ਕੁੱਝ ਹਾਲਾਤਾਂ ਵਿੱਚ ਸਾਨੂੰ ਡਰਾਈਵਰਾਂ ਨੂੰ ਮੈਨੂਅਲੀ ਡਾਊਨਲੋਡ ਕਰਨ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ। ਨਵੇਂ ਡਰਾਈਵਰਾਂ ਨੂੰ ਡਾਊਨਲੋਡ ਕਰਨ ਲਈ ਸਾਨੂੰ ਡਿਵਾਈਸ ਨਿਰਮਾਤਾ (manufacturer) ਦੀ ਵੈੱਬਸਾਈਟ ਤੇ ਜਾਣ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ। ਡਰਾਈਵਰ ਫਾਈਲ ਨੂੰ ਡਾਊਨਲੋਡ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਸਾਨੂੰ ਇਸਨੂੰ ਆਪਣੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਵਿੱਚ ਇੰਸਟਾਲ ਕਰਨਾ ਪਵੇਗਾ। ਜ਼ਿਆਦਾਤਰ ਡਰਾਈਵਰ ਐਗਜ਼ੀਕਿਊਟੇਬਲ ਫਾਈਲਾਂ ਦੇ ਰੂਪ ਵਿੱਚ ਉਪਲਬਧ ਹੰਦੇ ਹਨ। ਐਗਜ਼ੀਕਿਊਟੇਬਲ ਫਾਈਲ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਡ੍ਰਾਈਵਰ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਲਈ, ਸਾਨੂੰ ਸਿਰਫ਼ ਫਾਈਲ ਤੇ ਡਬਲ ਕਲਿੱਕ ਕਰਨ ਅਤੇ ਆਨ-ਸਕ੍ਰੀਨ ਨਿਰਦੇਸ਼ਾਂ ਦੀ ਪਾਲਣਾ ਕਰਨ ਦੀ ਲੋੜ ਪਵੇਗੀ।

ਜੇਕਰ ਸਾਡੀ ਡਿਵਾਈਸ ਸਹੀ ਢੰਗ ਨਾਲ ਕੰਮ ਨਹੀਂ ਕਰ ਰਹੀ ਹੈ, ਤਾਂ ਸਾਨੂੰ ਇਹ ਜਾਂਚ ਕਰਨੀ ਪਵੇਗੀ ਕਿ ਡਿਵਾਈਸ ਦਾ ਡਰਾਈਵਰ ਸਹੀ ਢੰਗ ਨਾਲ ਇੰਸਟਾਲ ਹੋ ਗਿਆ ਹੈ ਜਾਂ ਨਹੀ। ਨੁਕਸਦਾਰ (ਫਾਲਟੀ) ਡਰਾਈਵਰ ਹਮੇਸ਼ਾ ਸਾਡੇ PC ਵਿੱਚ

ਸਮੱਸਿਆਵਾਂ ਪੈਦਾ ਕਰ ਸਕਦੇ ਹਨ। ਅਜਿਹੀਆਂ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਹੱਲ ਕਰਨ ਲਈ, ਸਾਨੂੰ ਉਸ ਡਿਵਾਈਸ ਲਈ ਡਰਾਈਵਰਾਂ ਨੂੰ ਅੱਪਡੇਟ ਜਾਂ ਰੀ-ਇੰਸਟਾਲ ਕਰਨ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ।

6.8.1 ਪਲੱਗ ਐਂਡ ਪਲੇ ਡਿਵਾਈਸਿਸ (Plug and Play (PnP) Devices):

ਪਲੱਗ ਐਂਡ ਪਲੇ ਡਿਵਾਈਸਾਂ ਨੂੰ ਅਕਸਰ PnP ਡਿਵਾਈਸ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਸ਼ਬਦ ਦਾ ਅਰਥ ਇਹ ਹੈ ਕਿ ਇਹ ਡਿਵਾਈਸਾਂ ਜਿਵੇਂ ਹੀ ਪੋਰਟਸ ਨਾਲ ਜੁੜਦੀਆਂ ਹਨ, ਇਹ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨਾਲ ਕੰਮ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰ ਦਿੰਦੀਆਂ ਹਨ। ਯੂਜਰਸ ਨੂੰ ਅਜਿਹੀਆਂ ਡਿਵਾਈਸਾਂ ਲਈ ਖੁਦ ਡਰਾਈਵਰ ਇੰਸਟਾਲ ਕਰਨ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਪੈਂਦੀ, ਇਸਦੀ ਬਜਾਏ, ਕੰਪਿਊਟਰ ਆਪਣੇ ਆਪ ਹੀ ਨਵੇਂ ਜੁੜੇ ਡਿਵਾਈਸ ਨੂੰ ਪਛਾਣ ਲੈਂਦਾ ਹੈ ਅਤੇ ਲੋੜ ਪੈਣ ਤੇ ਡਰਾਈਵਰਾਂ ਨੂੰ ਲੋਡ ਕਰ ਦਿੰਦਾ ਹੈ ਅਤੇ ਇਸ ਨਾਲ ਕੰਮ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰ ਦਿੰਦਾ ਹੈ।

ਚਿੱਤਰ 6.23: ਪਲਗ ਅਤੇ ਪਲੇ

ਉਦਾਹਰਨ ਲਈ, ਜੇਕਰ ਅਸੀਂ ਇੱਕ ਪਲੱਗ-ਐਂਡ-ਪਲੇ ਕੀਬੋਰਡ ਨੂੰ ਆਪਣੇ ਕੰਪਿਊਟਰ ਦੇ USB ਪੋਰਟ ਨਾਲ ਕੁਨੈਕਟ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਇਹ ਪਲੱਗ ਇਨ ਕਰਨ ਦੇ ਕੁੱਝ ਸਕਿੰਟਾਂ ਦੇ ਅੰਦਰ ਕੰਮ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰ ਦੇਵੇਗਾ। ਧਿਆਨ ਵਿੱਚ ਰੱਖਣ ਵਾਲੀ ਗੱਲ ਇਹ ਹੈ ਕਿ ਅੰਦਰੂਨੀ ਭਾਗਾਂ (ਇੰਟਰਨਲ ਕੰਪੋਨੈਂਟਸ) ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਲਈ ਆਮ ਤੌਰ ਤੇ ਕੰਪਿਊਟਰ ਨੂੰ ਬੰਦ ਕਰਨ ਦੀ ਲੋੜ ਪੈਂਦੀ ਹੈ, ਜਦੋਂ ਕਿ ਬਾਹਰੀ ਡਿਵਾਈਸਾਂ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਦਾ ਕੰਮ ਕੰਪਿਊਟਰ ਦੇ ਚੱਲਦੇ ਸਮੇਂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

6.9 PC ਸਿਕਿਉਰਟੀ ਅਤੇ ਇਸਦੇ ਟੂਲਸ (PC SECURITY AND ITS TOOLS):

PC ਸਿਕਿਉਰਟੀ ਨੂੰ IT ਸਿਕਿਉਰਟੀ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਵਿਚ ਉਹ ਤਕਨੀਕਾਂ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ ਜੋ ਪੀਸੀ, ਲੈਪਟਾਪ ਅਤੇ ਹੋਰ ਨਿੱਜੀ ਡਿਵਾਈਸਾਂ ਨੂੰ ਇੱਕ ਨੈਟਵਰਕ ਤੇ ਸੁਰੱਖਿਅਤ ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਇਹ ਸਿਕਿਉਰਟੀ ਨਿੱਜੀ ਅਤੇ ਜਨਤਕ ਦੋਵੇਂ ਤਰ੍ਹਾਂ ਦੇ ਕੰਪਿਊਟਰ ਨੈਟਵਰਕਾਂ ਨਾਲ ਸੰਬੰਧਿਤ ਹੈ। ਸਧਾਰਨ ਸ਼ਬਦਾਂ ਵਿੱਚ, ਅਸੀਂ ਕਹਿ ਸਕਦੇ ਹਾਂ ਕਿ PC ਸਿਕਿਉਰਟੀ ਇੱਕ ਪ੍ਰੋਟੇਕਸ਼ਨ ਸਿਸਟਮ ਹੈ ਜੋ ਕੰਪਿਊਟਰ ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਡਾਟਾ ਅਤੇ ਜਾਣਕਾਰੀ ਨੂੰ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ (unauthorized access), ਦੁਰਵਰਤੋਂ ਅਤੇ ਚੋਰੀ ਤੋਂ ਬਚਾਉਂਦੀ ਹੈ। ਕੰਪਿਊਟਰ ਸੁਰੱਖਿਆ ਵੱਖ-ਵੱਖ ਐਪਲੀਕੇਸ਼ਨਾਂ ਅਤੇ ਸਿਸਟਮਾਂ ਨੂੰ ਖਤਰਨਾਕ ਗਤੀਵਿਧੀਆਂ ਤੋਂ ਬਚਾਉਂਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੇ ਕਾਰਨਾਂ ਕਰਕੇ ਪੀਸੀ ਸਿਕਿਉਰਟੀ ਨੂੰ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ:

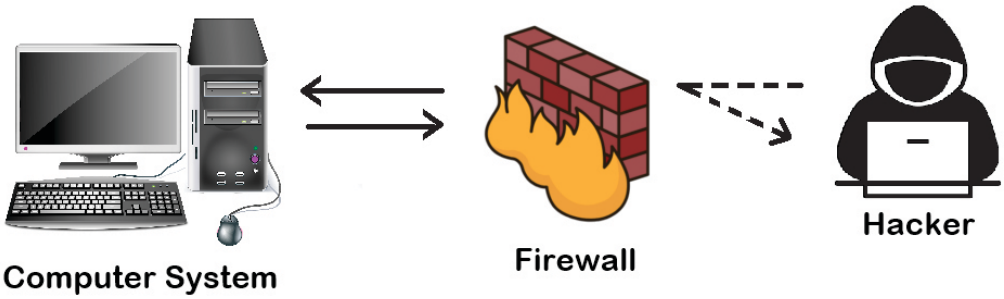
- ਡਾਟਾ ਚੋਰੀ ਨੂੰ ਰੋਕਣ ਲਈ
- ਹਾਰਡਵੇਅਰ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਣ ਤੋਂ ਰੋਕਣ ਲਈ
- ਕਿਸੇ ਵੀ ਸਾਫਟਵੇਅਰ ਜਾਂ ਸੇਵਾ ਵਿੱਚ ਰੁਕਾਵਟ ਨੂੰ ਰੋਕਣ ਲਈ

ਕੰਪਿਊਟਰ ਸਿਕਿਉਰਟੀ ਇਹ ਯਕੀਨੀ ਬਣਾਉਂਦੀ ਹੈ ਕਿ ਸਾਡਾ ਸਿਸਟਮ, ਡਾਟਾ ਅਤੇ ਨੈੱਟਵਰਕ ਵੱਖ-ਵੱਖ ਖਤਰਿਆਂ ਜਿਵੇਂ ਕਿ: ਵਾਇਰਸ, ਹੈਕਿੰਗ ਅਤੇ ਚੋਰੀ ਵਰਗੇ ਖਤਰਿਆਂ ਤੋਂ ਸੁਰੱਖਿਅਤ ਹਨ। ਐਂਟੀਵਾਇਰਸ, ਫਾਇਰਵਾਲ, ਕਲਾਉਡ ਸਟੋਰੇਜ, ਐਨਕ੍ਰਿਪਸ਼ਨ ਆਦਿ ਆਮ ਬੁਨਿਆਦੀ ਹਿੱਸੇ ਹਨ ਜੋ ਪੀਸੀ ਸਿਕਿਉਰਟੀ ਨਾਲ ਸੰਬੰਧਿਤ ਹਨ:

- **ਐਂਟੀਵਾਇਰਸ ਸਾਫਟਵੇਅਰ (Antivirus Software):** ਕੰਪਿਊਟਰ ਵਾਇਰਸ ਇੱਕ ਕਿਸਮ ਦਾ ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰ ਹੈ (ਜਿਸਨੂੰ ਮਾਲਵੇਅਰ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ) ਜੋ ਸਾਡੇ ਡਾਟਾ ਅਤੇ ਹੋਰ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਇਨਫੈਕਟ (infects) ਕਰਦਾ ਹੈ। ਇਹ ਵਾਇਰਸ ਤੇਜ਼ੀ ਨਾਲ ਫੈਲਦੇ ਹਨ ਅਤੇ ਡਾਟਾ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਂਦੇ ਹਨ। ਇੱਥੋਂ ਤੱਕ ਕਿ ਇਹ ਪੂਰੀ ਤਰ੍ਹਾਂ ਸਿਸਟਮ ਨੂੰ ਬੰਦ ਵੀ ਕਰ ਸਕਦੇ ਹਨ। ਐਂਟੀਵਾਇਰਸ ਸਾਫਟਵੇਅਰ ਸਾਡੇ ਸਿਸਟਮ ਤੋਂ ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਸਾਫਟਵੇਅਰ (ਮਾਲਵੇਅਰ) ਨੂੰ ਸਕੈਨ ਅਤੇ ਖੋਜ ਕਰਨ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦਾ ਹੈ। ਇਹਨਾਂ ਮਾਲਵੇਅਰਾਂ ਵਿੱਚ ਵਾਇਰਸ, ਵਾਰਮਸ, ਸਪਾਈਵੇਅਰ, ਕੀਲੌਗਰ, ਰੈਨਸਮਵੇਅਰ ਅਤੇ ਹੋਰ ਬਹੁਤ ਕੁਝ ਸ਼ਾਮਲ ਹਨ। ਐਂਟੀਵਾਇਰਸ ਸਾਫਟਵੇਅਰ ਸਾਡੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨੂੰ ਉਹਨਾਂ ਪੈਟਰਨਾਂ ਲਈ ਸਕੈਨ ਕਰਨ ਦਾ ਕੰਮ ਕਰਦਾ ਹੈ, ਜੋ ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰ ਦੀ ਮੌਜੂਦਗੀ ਨੂੰ ਦਰਸਾ ਸਕਦੇ ਹਨ। ਈਮੇਲ ਸਕੈਨਿੰਗ ਐਂਟੀਵਾਇਰਸ ਸੌਫਟਵੇਅਰ ਦੀ ਇੱਕ ਹੋਰ ਸੇਵਾ ਹੈ ਜੋ ਪੁਸ਼ਟੀ ਕਰਦੀ ਹੈ ਕਿ ਈਮੇਲ

ਖਤਰਨਾਕ ਅਟੈਚਮੈਂਟਾਂ ਤੋਂ ਮੁਕਤ ਹਨ। ਐਂਟੀਵਾਇਰਸ ਸਾਫਟਵੇਅਰ ਵਿੱਚ ਆਮ ਤੌਰ ਤੇ ਨਵੇਂ ਵਾਇਰਸਾਂ ਅਤੇ ਖਤਰਿਆਂ ਲਈ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਸਕੈਨ ਕਰਨ ਲਈ ਇੱਕ ਆਟੋ-ਅਪਡੇਟ ਫੀਚਰ ਹੁੰਦਾ ਹੈ।

- **ਫਾਇਰਵਾਲ (Firewall):** ਫਾਇਰਵਾਲ ਇੱਕ ਸਿਕਿਓਰਿਟੀ ਸਿਸਟਮ ਹੈ, ਜੋ ਸਾਡੇ ਸਿਸਟਮ ਨੂੰ ਇੱਕ ਨੈੱਟਵਰਕ ਨਾਲ ਜੁੜੇ ਹੋਣ ਤੇ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ (unauthorized access) ਦੀ ਇਜਾਜ਼ਤ ਨਹੀਂ ਦਿੰਦਾ। ਇਹ ਸੁਰੱਖਿਆ ਨਿਯਮਾਂ ਦੇ ਆਧਾਰ ਤੇ ਇਨਕਮਿੰਗ ਅਤੇ ਆਊਟਗੋਇੰਗ ਨੈੱਟਵਰਕ ਟ੍ਰੈਫਿਕ ਦੀ ਨਿਗਰਾਨੀ ਅਤੇ ਕੰਟਰੋਲ ਕਰਦਾ ਹੈ। ਜੇਕਰ ਨੈੱਟਵਰਕ ਵਿੱਚ ਅਣਅਧਿਕਾਰਤ ਸੁਨੇਹੇ ਪਾਏ ਜਾਂਦੇ ਹਨ ਅਤੇ ਉਹ ਸੁਰੱਖਿਆ ਨਿਯਮਾਂ ਦੀ ਪੂਰਤੀ ਨਹੀਂ ਕਰਦੇ ਹਨ, ਤਾਂ ਉਹਨਾਂ ਨੂੰ ਤੁਰੰਤ ਬਲਾਕ ਕਰ ਦਿੱਤਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ, ਇਹ ਸਾਡੇ ਕੰਪਿਊਟਰ ਤੇ ਖਤਰਨਾਕ ਆਵਾਜਾਈ ਨੂੰ ਰੋਕ ਕੇ ਸਾਡੇ ਸਿਸਟਮ ਦੀ ਰੱਖਿਆ ਕਰਦਾ ਹੈ। ਫਾਇਰਵਾਲ ਨੂੰ ਸਾਡੇ ਕੰਪਿਊਟਰ ਅਤੇ ਨੈੱਟਵਰਕ ਦੇ ਵਿਚਕਾਰ ਇੱਕ ਵਰਚੁਅਲ ਰੁਕਾਵਟ ਦੇ ਰੂਪ ਵਿੱਚ ਜਾਣਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਵਿੰਡੋਜ਼ ਡਿਫੈਂਡਰ ਫਾਇਰਵਾਲ ਨੂੰ ਵਿੰਡੋਜ਼ ਦੇ ਲੇਟੈਸਟ ਵਰਜ਼ਨ ਵਿੱਚ ਬਣਾਇਆ ਗਿਆ ਹੈ ਜੋ ਸਾਡੇ ਪੀਸੀ ਦੀ ਸੁਰੱਖਿਆ ਵਿੱਚ ਮਦਦ ਕਰਦਾ ਹੈ।



ਚੱਤਰ 6.24: ਫਾਇਰਵਾਰ ਸਿਕਿਓਰਿਟੀ ਸਿਸਟਮ

- **ਕਲਾਉਡ ਸਟੋਰੇਡ (Cloud Storage):** ਸਾਡੇ PC ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਡਾਟਾ ਨੂੰ ਖਤਰਨਾਕ ਹੈਕਰਾਂ ਤੋਂ ਸੁਰੱਖਿਅਤ ਕਰਨ ਲਈ, ਇਸਨੂੰ ਕਲਾਉਡ ਸਟੋਰੇਜ ਉੱਤੇ ਸਟੋਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਇਹ ਸਾਡੇ ਡਾਟਾ ਨੂੰ ਸੁਰੱਖਿਅਤ ਕਰਨ ਵਿੱਚ ਮਦਦ ਕਰਦਾ ਹੈ ਅਤੇ PC ਸੁਰੱਖਿਆ ਵਿੱਚ ਯੋਗਦਾਨ ਪਾਉਂਦਾ ਹੈ।

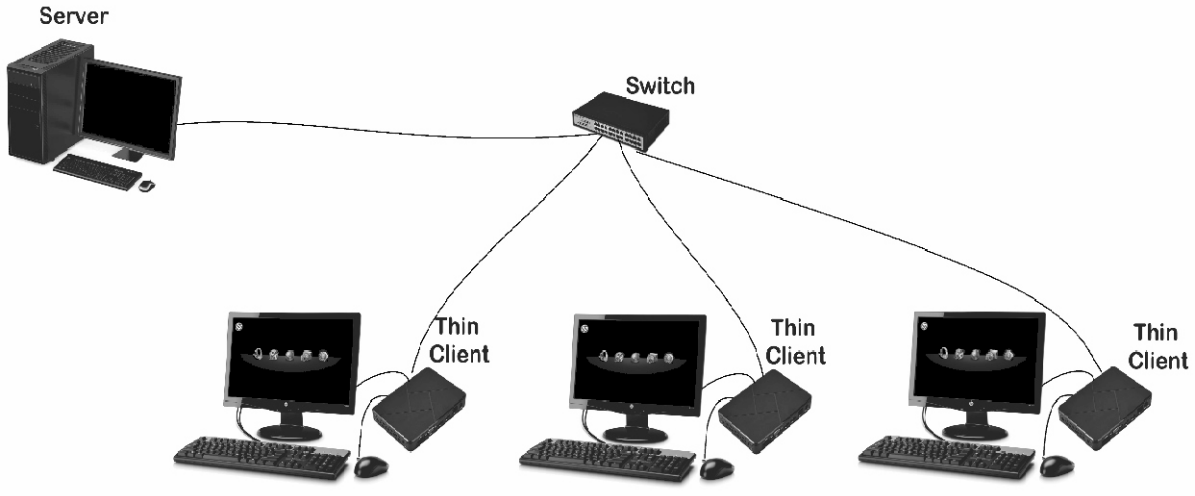
- **ਏਨਕ੍ਰਿਪਸ਼ਨ (Encryption):** ਏਨਕ੍ਰਿਪਸ਼ਨ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ (unauthorized access) ਨੂੰ ਰੋਕਣ ਲਈ ਸਾਡੇ ਡਾਟਾ ਨੂੰ ਕੋਡਸ ਵਿੱਚ ਬਦਲਣ ਦਾ ਇੱਕ ਤਰੀਕਾ ਹੈ। ਇਹ ਸਾਡੇ ਡਾਟਾ ਨੂੰ ਲਾਕ ਕਰਨ ਵਰਗਾ ਹੁੰਦਾ ਹੈ। ਜੇਕਰ ਕੋਈ ਇਸ ਐਨਕ੍ਰਿਪਟਿਡ ਡਾਟਾ ਨੂੰ ਚੋਰੀ ਕਰਦਾ ਹੈ, ਤਾਂ ਉਹ ਇਸਨੂੰ ਡੀਕ੍ਰਿਪਸ਼ਨ ਕੀਅ ਤੋਂ ਬਿਨਾਂ ਨਹੀਂ ਸਮਝ ਸਕੇਗਾ। ਇਹ ਤਕਨੀਕ ਖਾਸ ਤੌਰ ਤੇ ਸੰਵੇਦਨਸ਼ੀਲ ਜਾਣਕਾਰੀ ਜਿਵੇਂ ਕਿ: ਕ੍ਰੈਡਿਟ ਕਾਰਡ ਨੰਬਰ, ਨਿੱਜੀ ਵੇਰਵੇ ਆਦਿ ਦੀ ਸੁਰੱਖਿਆ ਲਈ ਲਾਭਦਾਇਕ ਹੁੰਦੀ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ, ਇਹ ਤਕਨੀਕ ਸਾਡੇ ਡਾਟਾ ਨੂੰ ਸੁਰੱਖਿਅਤ ਕਰਕੇ ਕੰਪਿਊਟਰ ਸੁਰੱਖਿਆ ਨੂੰ ਬਹੁਤ ਵਧਾਉਂਦੀ ਹੈ।

6.10 ਥਿਨ ਕਲਾਇੰਟ ਟੈਕਨੋਲੋਜੀ (THIN CLIENT TECHNOLOGY):

ਬਹੁਤੇ ਲੋਕ ਲੋਕਲ ਡਰਾਈਵ ਤੇ ਇੰਸਟਾਲਡ ਸਾਫਟਵੇਅਰ ਵਾਲੇ ਡੈਸਕਟਾਪ ਕੰਪਿਊਟਰਾਂ ਤੋਂ ਜਾਣੂ ਹਨ। ਇਸ ਸਿਸਟਮ ਵਿੱਚ, ਯੂਜ਼ਰ ਲੋਕਲ ਹਾਰਡਵੇਅਰ ਨਾਲ ਸਿੱਧਾ ਇੰਟਰਐਕਟ ਕਰਦਾ ਹੈ ਅਤੇ ਹਰ ਚੀਜ਼ ਨੂੰ ਇੱਕ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੁਆਰਾ ਨਿਯੰਤਰਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਪਰ ਥਿਨ ਕਲਾਇੰਟ ਟੈਕਨੋਲੋਜੀ ਵਿੱਚ, ਕੇਵਲ ਇੱਕ ਮਾਨੀਟਰ, ਨੈਟਵਰਕ ਕਾਰਡ, ਮਾਊਸ ਅਤੇ ਕੀਬੋਰਡ ਹੀ ਐਂਡ ਯੂਜਰ ਲਈ ਉਪਲਬਧ ਹੁੰਦਾ ਹੈ, ਅਤੇ ਹੋਰ ਹਾਰਡਵੇਅਰ ਅਤੇ ਸੇਵਾਵਾਂ ਨੈਟਵਰਕ ਤੇ ਚਲਦੀਆਂ ਹਨ।

ਥਿਨ ਕਲਾਇੰਟ ਬੁਨਿਆਦੀ ਕੰਪਿਊਟਿੰਗ ਯੰਤਰ ਹਨ, ਜਿੱਥੇ ਸੇਵਾਵਾਂ ਅਤੇ ਸਾਫਟਵੇਅਰ ਨੂੰ ਇੱਕ ਕੇਂਦਰੀ ਸਰਵਰ ਤੋਂ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਡਿਵਾਈਸਾਂ ਦੀ ਸੀਮਿਤ ਕੰਪਿਊਟਿੰਗ ਸਮਰੱਥਾ ਹੁੰਦੀ ਹੈ। ਇਹਨਾਂ ਡਿਵਾਈਸਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ, ਯੂਜ਼ਰ ਕੇਂਦਰੀ ਸਰਵਰ ਨਾਲ ਡਾਟਾ ਦਾ ਆਦਾਨ-ਪ੍ਰਦਾਨ ਕਰਕੇ ਵਧੇਰੇ ਗੁੰਝਲਦਾਰ ਅਤੇ ਗਣਨਾਵਾਂ ਕਰਨ ਵਾਲੇ ਕੰਮ ਕਰ ਸਕਦੇ ਹਨ। ਥਿਨ-ਕਲਾਇੰਟਸ ਇੱਕ ਕੇਂਦਰੀ ਸਰਵਰ ਲਈ ਐਕਸੈਸ ਪੁਆਇੰਟ ਵਜੋਂ ਕੰਮ ਕਰਦੇ ਹਨ। ਇਹ

ਯੰਤਰ ਕੰਪਿਊਟੇਸ਼ਨਲ ਕੰਮਾਂ ਲਈ ਸਰਵਰ ਉਪਰ ਬਹੁਤ ਜ਼ਿਆਦਾ ਨਿਰਭਰ ਕਰਦੇ ਹਨ। ਉਹ ਮੁੱਖ ਤੌਰ ਤੇ ਪ੍ਰੋਸੈਸਡ ਡਾਟਾ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਅਤੇ ਸਰਵਰ ਨੂੰ ਯੂਜ਼ਰ ਇਨਪੁਟਸ ਭੇਜਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।



ਚਿੱਤਰ 6.25: ਥਿਨ ਕਲਾਇੰਟ ਟੈਕਨੋਲੋਜੀ

ਉਹ ਸੰਸਥਾਵਾਂ ਜੋ ਇੱਕ ਥਿਨ ਕਲਾਇੰਟ ਵਾਤਾਵਰਣ ਦੀ ਵਰਤੋਂ ਨੂੰ ਤਰਜੀਹ ਦਿੰਦੀਆਂ ਹਨ, ਕਈ ਤਰ੍ਹਾਂ ਦੇ ਲਾਭਾਂ ਨੂੰ ਮਹਿਸੂਸ ਕਰ ਸਕਦੀਆਂ ਹਨ। ਸੰਸਥਾ ਵਿੱਚ ਸਾਰੇ ਕਰਮਚਾਰੀਆਂ ਲਈ ਨਵੇਂ ਕੰਪਿਊਟਰ ਲਗਾਉਣਾ ਬਹੁਤ ਮਹਿੰਗਾ ਸਾਬਤ ਹੁੰਦਾ ਹੈ। ਥਿਨ ਕਲਾਇੰਟ ਕੰਪਿਊਟਿੰਗ ਕਰਮਚਾਰੀਆਂ ਲਈ ਇੰਨੇ ਸਾਰੇ ਕੰਪਿਊਟਰ ਖਰੀਦਣ ਦੀ ਲਾਗਤ ਨੂੰ ਘਟਾਉਂਦੀ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਥਿਨ ਕਲਾਇੰਟ ਟੈਕਨੋਲੋਜੀ ਵੱਡੀਆਂ ਸੰਸਥਾਵਾਂ ਦੀ ਲਾਗਤ ਘਟਾ ਕੇ ਫਲੈਕਸੀਬਲ ਅਤੇ ਸੈਂਟਰਲਾਈਜਡ ਕੰਪਿਊਟਿੰਗ ਹੱਲ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।

ਯਾਦ ਰੱਖਣ ਯੋਗ ਗੱਲਾਂ

1. ਰੱਖ-ਰਖਾਵ ਦੀਆਂ ਉਹ ਵੱਖ-ਵੱਖ ਗਤੀਵਿਧੀਆਂ ਜਿਹਨਾਂ ਨਾਲ ਅਸੀਂ ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਜਾਂ ਕੰਪਿਊਟਰ ਦੇ ਭਾਗਾਂ ਨੂੰ ਉਹਨਾਂ ਦੇ ਕੰਮ ਕਾਜ ਦੀ ਚੰਗੀ ਸਥਿਤੀ ਵਿੱਚ ਰੱਖਦੇ ਹਾਂ, ਨੂੰ ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।
2. ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਵਿੱਚ ਕੰਪਿਊਟਰ ਦੇ ਭੌਤਿਕ ਭਾਗਾਂ, ਜਿਵੇਂ ਕਿ ਇਸਦਾ ਕੀਬੋਰਡ, ਮਾਊਸ, ਹਾਰਡ ਡਰਾਈਵ ਆਦਿ ਦੀ ਦੇਖਬਾਲ ਕਰਨਾ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ।
3. ਸੌਫਟਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਯੂਜ਼ਰ ਦੀਆਂ ਜ਼ਰੂਰਤਾਂ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਸਾਫਟਵੇਅਰ ਨੂੰ ਬਦਲਣ, ਸੋਧਣ ਅਤੇ ਅਪਡੇਟ ਕਰਨ ਦੀ ਪ੍ਰਕਿਰਿਆ ਹੈ।
4. ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਉਹ ਫਾਈਲਾਂ ਹੁੰਦੀਆਂ ਹਨ, ਜੋ ਟੈਂਪਰੇਰੀ ਤੌਰ ਤੇ ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਬਣਾਈਆਂ ਜਾਂਦੀਆਂ ਹਨ।
5. ਡਿਸਕ ਕਲੀਨ-ਅੱਪ ਉਹ ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ ਸਹੂਲਤ ਹੈ, ਜੋ ਮਾਈਕ੍ਰੋਸਾਫਟ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿੱਚ ਸ਼ਾਮਲ ਹੈ ਅਤੇ ਇਸਨੂੰ ਹਾਰਡ ਡਰਾਈਵ ਤੋਂ ਜਗ੍ਹਾ ਖਾਲੀ ਕਰਨ ਲਈ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਹੈ।
6. ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਸਾਡੀ ਹਾਰਡ ਡਰਾਈਵ ਵਿੱਚ ਫੈਲੇ ਡਾਟਾ ਦੇ ਸਾਰੇ ਟੁਕੜਿਆਂ ਨੂੰ ਦੁਬਾਰਾ ਇਕੱਠੇ ਕਰਦਾ ਹੈ।
7. ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ (OS) ਇੱਕ ਸਾਫਟਵੇਅਰ ਹੈ, ਜੋ ਯੂਜ਼ਰ ਅਤੇ ਕੰਪਿਊਟਰ ਹਾਰਡਵੇਅਰ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਵਜੋਂ ਕੰਮ ਕਰਦਾ ਹੈ।

8. ਜਦੋਂ ਅਸੀਂ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦਾ ਪਾਵਰ ਬਟਨ ਦਬਾਉਂਦੇ ਹਾਂ, ਤਾਂ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਲੋਡ ਹੋਣਾ ਸ਼ੁਰੂ ਹੋ ਜਾਂਦਾ ਹੈ, ਇਹ ਪ੍ਰਕਿਰਿਆ ਨੂੰ ਬੂਟਿੰਗ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

9. ਰੀਜਨਲ ਸੈਟਿੰਗਾਂ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੀਆਂ ਉਹ ਸੈਟਿੰਗਜ਼ ਹੁੰਦੀਆਂ ਹਨ, ਜੋ ਯੂਜ਼ਰ ਦੀ ਲੋਕੇਸ਼ਨ ਨਾਲ ਸਬੰਧਤ ਹੁੰਦੀਆਂ ਹਨ, ਜਿਵੇਂ ਕਿ ਲੈਂਗੁਏਜ, ਕਰੰਸੀ, ਟਾਈਮ ਜੋਨ ਆਦਿ।

10. ਫੌਂਟ ਇੱਕ ਟਾਈਪਫੇਸ ਅਤੇ ਹੋਰ ਗੁਣਾਂ ਦਾ ਸੁਮੇਲ ਹੁੰਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਆਕਾਰ, ਪਿੱਚ, ਅਤੇ ਸਪੇਸਿੰਗ।

11. ਯੂਟੀਲਟੀ ਪ੍ਰੋਗਰਾਮਜ਼ ਜਾਂ ਟੂਲਜ਼ ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਸਹੀ ਅਤੇ ਨਿਰਵਿਘਨ ਕੰਮਕਾਜ ਨੂੰ ਬਣਾਈ ਰੱਖਣ ਵਿੱਚ ਸਾਡੀ ਮਦਦ ਕਰਦੇ ਹਨ। ਇਹ ਪ੍ਰੋਗਰਾਮਜ਼ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਕੰਮਕਾਜ ਦੇ ਪ੍ਰਬੰਧਣ (organize), ਰੱਖ-ਰਖਾਅ (maintain) ਅਤੇ ਉਸਨੂੰ ਅਨੁਕੂਲ (optimize) ਬਣਾਉਣ ਲਈ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਨੂੰ ਸਹਾਇਤਾ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ।

12. ਸਾਫਟਵੇਅਰ ਅੱਪਡੇਟ ਵਿੱਚ ਬੱਗ ਫਿਕਸ ਅਤੇ ਹੋਰ ਛੋਟੇ ਸੁਧਾਰ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ, ਜਦੋਂ ਕਿ ਇੱਕ ਸਾਫਟਵੇਅਰ ਅੱਪਗਰੇਡ ਸਾਫਟਵੇਅਰ ਦੇ ਵਰਜਨ ਨੂੰ ਬਦਲਦਾ ਹੈ।

13. ਕੰਪਿਊਟਰ ਹਾਰਡਵੇਅਰ ਵਿੱਚ, ਇੱਕ ਪੋਰਟ ਕੰਪਿਊਟਰ ਅਤੇ ਹੋਰ ਕੰਪਿਊਟਰਾਂ ਜਾਂ ਪੈਰੀਫਿਰਲ ਡਿਵਾਈਸਾਂ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਵਜੋਂ ਕੰਮ ਕਰਦੀ ਹੈ।

14. ਪਲੱਗ ਐਂਡ ਪਲੇ ਡਿਵਾਈਸਾਂ ਨੂੰ ਅਕਸਰ PnP ਡਿਵਾਈਸ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਸ਼ਬਦਾ ਦਾ ਅਰਥ ਇਹ ਹੈ ਕਿ ਡਿਵਾਈਸਾਂ ਜਿਵੇਂ ਹੀ ਪੋਰਟਾਂ ਨਾਲ ਜੁੜਦੀਆਂ ਹਨ ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਨਾਲ ਕੰਮ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰ ਦਿੰਦੀਆਂ ਹਨ।

15. PC ਸਿਕਿਉਰਟੀ ਇੱਕ ਪ੍ਰੋਟੈਕਸ਼ਨ ਸਿਸਟਮ ਹੈ, ਜੋ ਕੰਪਿਊਟਰ ਵਿੱਚ ਸਟੋਰ ਕੀਤੇ ਡਾਟਾ ਅਤੇ ਜਾਣਕਾਰੀ ਨੂੰ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ, ਦੁਰਵਰਤੋਂ ਅਤੇ ਚੋਰੀ ਤੋਂ ਬਚਾਉਂਦੀ ਹੈ।

ਅਭਿਆਸ

ੳ. ਬਹੁ-ਵਿਕਲਪੀ ਪ੍ਰਸ਼ਨ:

1. ਉਹ ਸਾਫਟਵੇਅਰ ਜੋ ਯੂਜ਼ਰ ਅਤੇ ਡਿਵੈਲਪਰ ਲਈ ਸਹਾਇਕ ਭੂਮਿਕਾ ਨਿਭਾਉਂਦੇ ਹਨ, ਨੂੰ _____ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ੳ. ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ
ਅ. ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮ
ਏ. ਪ੍ਰੋਟੈਕਸ਼ਨ ਟੂਲਸ
ਸ. ਡਿਵੈਲਪਰ ਟੂਲ

2. _____ ਸਾਡੀ ਹਾਰਡ ਡਰਾਈਵ ਵਿੱਚ ਡਾਟਾ ਦੇ ਸਾਰੇ ਖਿੰਡੇ ਹੋਏ ਟੁਕੜਿਆਂ ਨੂੰ ਚੁੱਕਦਾ ਹੈ ਅਤੇ ਇੱਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀ ਬਿਹਤਰ ਕਾਰਗੁਜ਼ਾਰੀ ਲਈ ਉਹਨਾਂ ਨੂੰ ਦੁਬਾਰਾ ਇਕੱਠੇ ਰੱਖਦਾ ਹੈ।

ੳ. ਫਰੈਗਮੈਂਟੇਸ਼ਨ
ਅ. ਡਿਸਕ ਕਲੀਨਅਪ
ਏ. ਡੀਫਰੈਗਮੈਂਟੇਸ਼ਨ
ਸ. ਕੋਈ ਵੀ ਨਹੀਂ

3. _____ ਇੱਕ ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ ਹੈ ਜੋ ਯੂਜ਼ਰ ਅਤੇ ਕੰਪਿਊਟਰ ਹਾਰਡਵੇਅਰ ਵਿਚਕਾਰ ਇੱਕ ਇੰਟਰਫੇਸ ਵਜੋਂ ਕੰਮ ਕਰਦਾ ਹੈ।

ੳ. ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮ
ਅ. ਐਂਟੀਵਾਇਰਸ ਪ੍ਰੋਗਰਾਮ
ਏ. ਪ੍ਰੋਟੈਕਸ਼ਨ ਸਿਸਟਮ
ਸ. ਆਪਰਟਿੰਗ ਸਿਸਟਮ

4. ਹੇਠਾਂ ਦਿੱਤੇ ਮੋਡ ਵਿੱਚੋਂ ਕਿਹੜਾ ਕੰਪਿਊਟਰ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦਾ ਡਾਇਗਨੌਸਟਿਕ ਮੋਡ ਹੈ ?

ੳ. ਡਿਵੈਲਪਰ ਮੋਡ
ਅ. ਸੇਫ ਮੋਡ
ੲ. ਆਪਰੇਟਿੰਗ ਮੋਡ
ਸ. ਡਾਇਗਨੋਸਟਿਕ ਮੋਡ

5. ਸਾਡੇ ਸਿਸਟਮ ਨੂੰ ਮਾਲਵੇਅਰ ਤੋਂ ਬਚਾਉਣ ਲਈ ਹੇਠ ਲਿਖੀਆਂ ਵਿੱਚੋਂ ਕਿਹੜੀ ਸਹੂਲਤ ਵਰਤੀ ਜਾ ਸਕਦੀ ਹੈ ?

ੳ. ਕੰਪਰੈਸ਼ਨ ਟੂਲ
ਅ. ਐਂਟੀਵਾਇਰਸ ਟੂਲ
ੲ. ਡਿਸਕ ਮੈਨੇਜਮੈਂਟ ਟੂਲ
ਸ. ਕੋਈ ਵੀ ਨਹੀਂ

6. ਇੱਕ _____ ਵਿੱਚ ਬੱਗ ਫਿਕਸ ਅਤੇ ਹੋਰ ਛੋਟੇ ਸੁਧਾਰ ਸ਼ਾਮਲ ਹਨ।

ੳ. ਸਾਫਟਵੇਅਰ ਅਪਡੇਟ
ਅ. ਸਾਫਟਵੇਅਰ ਅਪਗਰੇਡ
ੲ. ਡਿਸਕ ਮੈਨੇਜਮੈਂਟ ਟੂਲ
ਸ. ਉਪਰੋਕਤ ਸਾਰੇ

7. _____ ਇੱਕ ਸਿਕਿਉਰਟੀ ਸਿਸਟਮ ਹੈ, ਜੋ ਸਾਡੇ ਸਿਸਟਮ ਨੂੰ ਇੱਕ ਨੈਟਵਰਕ ਨਾਲ ਜੁੜੇ ਹੋਣ ਤੇ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ ਦੀ ਇਜਾਜ਼ਤ ਨਹੀਂ ਦਿੰਦਾ।

ੳ. ਇਨਕ੍ਰਿਪਸ਼ਨ
ਅ. ਫਾਇਰਵਾਲ
ੲ. ਐਂਟੀਵਾਇਰਸ
ਸ. ਉਪਰੋਕਤ ਸਾਰੇ

ਅ. ਪੂਰੇ ਰੂਪ ਲਿਖੋ:

1. PnP
2. USB
3. PCB
4. DVI
5. OS
6. VGA

ੲ. ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

1. ਕੰਪਿਊਟਰ ਮੈਨਟੇਨੈਂਸ ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਭਾਵ ਹੈ ?
2. ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੀ ਕਾਰਗੁਜ਼ਾਰੀ ਨੂੰ ਸੁਧਾਰਨ ਲਈ ਕਿਸੇ ਚਾਰ ਤਕਨੀਕਾਂ ਬਾਰੇ ਲਿਖੋ।
3. ਬੂਟਿੰਗ ਕੀ ਹੈ ?
4. ਕੰਪਿਊਟਰਾਂ ਵਿੱਚ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਵੱਖ ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਪੋਰਟਸ ਦੇ ਨਾਮ ਲਿਖੋ ?
5. ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਵਿੱਚ ਨਵੇਂ ਫੋਂਟਸ ਕਿਵੇਂ ਇੰਸਟਾਲ ਕਰ ਸਕਦੇ ਹਾਂ ?
6. ਯੂਟੀਲਿਟੀ ਪ੍ਰੋਗਰਾਮ ਕੀ ਹੁੰਦੇ ਹਨ ?
7. ਡਿਵਾਈਸ ਡਰਾਈਵਰ ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਭਾਵ ਹੈ ?

ਸ. ਵੱਡੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

1. ਹਾਰਡਵੇਅਰ ਮੈਨਟੈਨੈਂਸ ਕੀ ਹੈ ? ਹਾਰਡਵੇਅਰ ਮੈਨਟੇਨੈਂਸ ਲਈ ਦਿਸ਼ਾ ਨਿਰਦੇਸ਼ਾਂ ਦੀ ਵਿਆਖਿਆ ਕਰੋ ?
2. ਪੋਰਟਸ ਕੀ ਹਨ ? ਕੋਈ ਦੋ ਪੋਰਟਸ ਦੀ ਵਿਸਥਾਰ ਨਾਲ ਵਿਆਖਿਆ ਕਰੋ ?
3. PC ਸਿਕਿਉਰਟੀ ਬਾਰੇ ਤੁਸੀਂ ਕੀ ਜਾਣਦੇ ਹੋ ? ਆਪਣੇ PC ਨੂੰ ਸਿਕਿਓਰ ਕਰਨ ਲਈ ਕਿਸੇ ਦੋ ਤਕਨੀਕਾਂ ਦਾ ਵਰਣਨ ਕਰੋ ?
4. ਅਪਡੇਟ ਅਤੇ ਅਪਗਰੇਡ ਵਿੱਚ ਕੀ ਅੰਤਰ ਹੈ ? ਵਿਆਖਿਆ ਕਰੋ ?

ਲੈਬ ਐਕਟੀਵਿਟੀ

- ਪਲੱਗ ਐਂਡ ਪਲੇ ਡਿਵਾਈਸਾਂ ਦੀ ਸੂਚੀ ਬਣਾਓ।
- ਆਪਣੀ ਕੰਪਿਊਟਰ ਲੈਬ ਵਿੱਚ ਕੰਪਿਊਟਰਾਂ ਤੇ ਉਪਲਬਧ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਪੋਰਟਾਂ ਦੀ ਪੜਚੋਲ ਕਰੋ।
- ਵਿੰਡੋਜ਼ ਨੂੰ ਸੇਫ ਮੋਡ ਵਿੱਚ ਸ਼ੁਰੂ ਕਰੋ।
- ਆਪਣੇ ਕੰਪਿਊਟਰ ਦੇ ਅਨ-ਨਸੈਸਰੀ ਸਟਾਰਟ ਅੱਪ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਬੰਦ ਕਰੋ।
- ਆਪਣੇ ਕੰਪਿਊਟਰ ਦੀਆਂ ਟੈਂਪਰੇਰੀ ਫਾਈਲਾਂ ਨੂੰ ਮਿਟਾਓ।
- ਆਪਣੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਤੇ ਡਿਸਕ ਡੀਫ੍ਰੈਗਮੈਂਟੇਸ਼ਨ ਚਲਾਓ।
- ਆਪਣੇ ਕੰਪਿਊਟਰ ਵਿੱਚ ਕੁੱਝ ਫੌਂਟ ਡਾਊਨਲੋਡ ਕਰੋ ਅਤੇ ਇੰਸਟਾਲ ਕਰੋ।
- ਵੱਖ-ਵੱਖ ਪ੍ਰਕਾਰ ਦੇ ਪੋਰਟਸ ਅਤੇ ਉਹਨਾਂ ਨਾਲ ਜੁੜਣ ਵਾਲੇ ਡਿਵਾਈਸਾਂ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੋਇਆਂ ਕੋਈ ਚਾਰਟ/ਮਾਡਲ ਤਿਆਰ ਕਰੋ।

ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਨਾਲ ਜਾਣ-ਪਛਾਣ

ਅੱਜਕੱਲ੍ਹ, ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਇੱਕ ਵੱਡੀ ਚਿੰਤਾ ਬਣ ਗਈ ਹੈ ਕਿਉਂਕਿ ਅਸੀਂ ਹਮੇਸ਼ਾਂ ਡਿਜੀਟਲ ਗੈਜੇਟਸ ਦੇ ਸੰਪਰਕ ਵਿੱਚ ਰਹਿੰਦੇ ਹਾਂ ਅਤੇ ਆਪਣੇ ਕੀਮਤੀ ਵਿੱਤੀ ਸਰੋਤਾਂ ਨੂੰ ਸੰਭਾਲਦੇ ਹਾਂ। ਸਾਡੀ ਜੀਵਨਸ਼ੈਲੀ ਵਿੱਚ ਬੇਸ਼ੱਕ ਸੁਧਾਰ ਹੋਇਆ ਹੈ ਪਰ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਦਾ ਖਤਰਾ ਵੀ ਵਧਿਆ ਹੈ। ਇਸ ਲਈ ਅੱਜ ਕੱਲ ਸੁਰੱਖਿਅਤ ਰਹਿਣ ਲਈ ਸਾਈਬਰ ਅਪਰਾਧ ਅਤੇ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਬਾਰੇ ਜਾਗਰੂਕਤਾ ਜ਼ਰੂਰੀ ਹੈ।

ਪਾਠ ਦੇ ਉਦੇਸ਼

- ✓ ਸਾਈਬਰ ਅਪਰਾਧ ਨਾਲ ਜਾਣ ਪਛਾਣ, ਪਰਿਭਾਸ਼ਾ ਅਤੇ ਸਾਈਬਰ ਅਪਰਾਧ ਦੀਆਂ ਕਿਸਮਾਂ, ਹੈਕਿੰਗ, ਫਿਸ਼ਿੰਗ, ਰੈਨਸਮਵੇਅਰ ਅਤੇ ਧੋਖਾਧੜੀ ਦੀਆਂ ਈਮੇਲਾਂ।
- ✓ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਦੀ ਮਹੱਤਤਾ, ਸੁਰੱਖਿਅਤ ਬ੍ਰਾਊਜ਼ਿੰਗ, ਸੁਰੱਖਿਆ ਉਪਾਅ, ਸਾਈਬਰ ਟ੍ਰੋਲ ਅਤੇ ਧੱਕੇਸ਼ਾਹੀ।
- ✓ ਵੈੱਬ ਸਾਈਟਾਂ ਤੱਕ ਸੁਰੱਖਿਅਤ ਢੰਗ ਨਾਲ ਪਹੁੰਚ ਕਰਨਾ, ਸਾਈਬਰ ਨੈਤਿਕਤਾ, ਮਾਲਵੇਅਰ, ਵਾਇਰਸ, ਟਰੋਜਨ, ਫਿਸ਼ਿੰਗ ਅਤੇ ਪਛਾਣ ਤਸਦੀਕ।
- ✓ ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ, ਈ-ਵੇਸਟ ਦੇ ਨਿਪਟਾਰੇ ਦੇ ਢੰਗ।
- ✓ ਸਾਈਬਰ ਕਾਨੂੰਨ, ਆਈ.ਟੀ. ਐਕਟ 2000

ਇਸ ਪਾਠ ਨੂੰ ਪੜ੍ਹਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਜਾਣ ਸਕਾਂਗੇ

- ✓ ਵਿਦਿਆਰਥੀ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਅਤੇ ਸਾਈਬਰ ਅਪਰਾਧ ਦੀਆਂ ਕਈ ਕਿਸਮਾਂ ਬਾਰੇ ਸਿੱਖਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਬਾਰੇ ਵੀ ਸਿੱਖਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀ ਸਿੱਖਣਗੇ ਕਿ ਵੈੱਬਸਾਈਟਾਂ ਨੂੰ ਸੁਰੱਖਿਅਤ ਢੰਗ ਨਾਲ ਕਿਵੇਂ ਬਰਾਊਜ਼ ਕਰਨਾ ਹੈ।
- ✓ ਵਿਦਿਆਰਥੀ ਵੱਖ-ਵੱਖ ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਬਾਰੇ ਸਿੱਖਣਗੇ।
- ✓ ਵਿਦਿਆਰਥੀਆਂ ਦੁਆਰਾ ਈ-ਵੇਸਟ ਮੈਨੇਜਮੈਂਟ ਬਾਰੇ ਸਿੱਖਿਆ ਜਾਵੇਗਾ।
- ✓ ਵਿਦਿਆਰਥੀ ਵੱਖ-ਵੱਖ ਸਾਈਬਰ ਕਾਨੂੰਨਾਂ ਬਾਰੇ ਜਾਣ ਸਕਣਗੇ।

07
ਪਾਠ

ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਅਤੇ ਨੈਤਿਕਤਾ

ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਮੁੱਖ ਤੌਰ 'ਤੇ ਡਿਜੀਟਲ ਬੁਨਿਆਦੀ ਢਾਂਚੇ, ਨੈੱਟਵਰਕਾਂ ਅਤੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਡਿਜੀਟਲ ਹਮਲਿਆਂ ਤੋਂ ਬਚਾਉਣ ਨਾਲ ਸਬੰਧਤ ਹੈ। ਇਹ ਸਾਈਬਰ-ਹਮਲੇ ਅਣਚਾਹੀ ਪਹੁੰਚ, ਤਬਦੀਲੀ ਜਾਂ ਕੀਮਤੀ ਜਾਣਕਾਰੀ ਨੂੰ ਨਸ਼ਟ ਕਰਨ ਤੇ ਕੇਂਦਰਿਤ ਹੁੰਦੇ ਹਨ। ਇਸ ਵਿੱਚ ਸੰਚਾਰ ਦੇ ਡਿਜੀਟਲ ਮਾਧਿਅਮ ਰਾਹੀਂ ਉਪਭੋਗਤਾਵਾਂ ਤੋਂ ਵਿੱਤੀ ਲਾਭ ਵੀ ਸ਼ਾਮਲ ਹਨ।

ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਨੂੰ ਲਾਗੂ ਕਰਨਾ ਅੱਜ ਇੱਕ ਚੁਣੌਤੀ ਬਣ ਗਿਆ ਹੈ ਕਿਉਂਕਿ ਅਸੀਂ ਹਮੇਸ਼ਾਂ ਡਿਜੀਟਲ ਸਰੋਤਾਂ ਨਾਲ ਘਿਰੇ ਰਹਿੰਦੇ ਹਾਂ ਜੋ ਸਾਡੀ ਨਿੱਜੀ ਅਤੇ ਵਿੱਤੀ ਸੰਪੱਤੀ ਨਾਲ ਸਬੰਧਤ ਹਨ। ਇਹ ਹਮਲਾਵਰਾਂ ਅਤੇ ਧੋਖੇਬਾਜ਼ਾਂ ਨਾਲ ਸਾਡੀ ਪਛਾਣ ਚੋਰੀ ਕਰਨ ਅਤੇ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਨਿਯਮਾਂ ਪ੍ਰਤੀ ਸਾਡੀ ਅਣਜਾਣਤਾ ਦੀ ਦੁਰਵਰਤੋਂ ਕਰਨ ਦਾ ਮੌਕਾ ਦਿੰਦਾ ਹੈ। ਆਓ ਪਹਿਲਾਂ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦੇ ਸਾਈਬਰ ਖਤਰਿਆਂ ਨੂੰ ਸਮਝੀਏ ਅਤੇ ਫਿਰ ਅਸੀਂ ਵੱਖੋ-ਵੱਖਰੇ ਤੌਰ ਤੇ ਇਹਨਾਂ ਹਮਲਿਆਂ ਤੋਂ ਪੀੜਤ ਹੋਣ ਤੋਂ ਸੁਰੱਖਿਅਤ ਰਹਿਣ ਲਈ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਨੈਤਿਕਤਾ ਬਾਰੇ ਪੜ੍ਹਾਂਗੇ।

7.1 ਸਾਈਬਰ ਅਪਰਾਧ ਨਾਲ ਜਾਣ-ਪਛਾਣ (Inroduction to Cyber Crime)

ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਵਿੱਚ ਗੈਰ-ਕਾਨੂੰਨੀ ਗਤੀਵਿਧੀਆਂ ਸ਼ਾਮਲ ਹੋ ਸਕਦੀਆਂ ਹਨ ਜੋ ਚੋਰੀ, ਧੋਖਾਧੜੀ, ਜਾਅਲਸਾਜ਼ੀ, ਮਾਣਹਾਨੀ ਜਾਂ ਸ਼ਰਾਰਤ ਲਈ ਇੱਕ ਸਾਧਨ ਵਜੋਂ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਇੱਕ ਅਪਰਾਧ ਹੈ ਜਿਸ ਵਿੱਚ ਇੱਕ ਕੰਪਿਊਟਰ, ਪੋਰਟੇਬਲ ਕੰਪਿਊਟਿੰਗ ਡਿਵਾਈਸ ਅਤੇ/ਜਾਂ ਇੱਕ ਨੈੱਟਵਰਕ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ। ਕਿਸੇ ਵੀ ਡਿਜੀਟਲ ਕੰਪਿਊਟਿੰਗ ਯੰਤਰ ਨੂੰ ਵੱਖ-ਵੱਖ ਤਰੀਕਿਆਂ ਨਾਲ ਇੱਕ ਟੂਲ ਦੇ ਨਾਲ-ਨਾਲ ਇੱਕ ਟੀਚੇ ਵਜੋਂ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਦਾ ਉਦੇਸ਼ ਕਿਸੇ ਵਿਅਕਤੀ, ਦੇਸ਼ ਦੀ ਸੁਰੱਖਿਆ ਜਾਂ ਵਿੱਤੀ ਸੰਪਤੀਆਂ ਲਈ ਖਤਰਾ ਹੈ।

ਕਿਸੇ ਵਿਅਕਤੀ ਜਾਂ ਲੋਕਾਂ ਦੇ ਸਮੂਹਾਂ ਦੇ ਵਿਰੁੱਧ ਇਲੈਕਟ੍ਰਾਨਿਕ ਸਾਧਨਾਂ ਦੁਆਰਾ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੀ ਸੰਪਤੀ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਣ ਜਾਂ ਸਰੀਰਕ ਜਾਂ ਮਾਨਸਿਕ ਸਦਮੇ ਦਾ ਕਾਰਨ ਬਣਨ ਨੂੰ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

ਇਲੈਕਟ੍ਰਾਨਿਕ ਸਾਧਨ ਆਸਾਨੀ ਨਾਲ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਦਾ ਸ਼ਿਕਾਰ ਹੋ ਸਕਦੇ ਹਨ ਕਿਉਂਕਿ ਆਧੁਨਿਕ ਦੂਰਸੰਚਾਰ ਨੈਟਵਰਕ ਜਿਵੇਂ ਕਿ ਇੰਟਰਨੈਟ (ਨੈਟਵਰਕ ਰਾਹੀਂ ਬਣਾਏ ਚੈਟ ਰੂਮ, ਈਮੇਲ, ਨੋਟਿਸ ਬੋਰਡ ਅਤੇ ਸਮੂਹ) ਅਤੇ ਮੋਬਾਈਲ ਫੋਨ (ਬਲੂਟੂਥ/ਐਸ.ਐਮ.ਐਸ./ਐਮ.ਐਮ.ਐਸ.) ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵੱਡੇ ਭੂਗੋਲਿਕ ਖੇਤਰਾਂ ਦੇ ਵਿਚਕਾਰ ਦੂਰੀ ਨੂੰ ਪੂਰਾ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਅਪਰਾਧੀ ਦੂਰ ਬੈਠ ਕੇ ਵੀ ਇਹ ਅਪਰਾਧ ਨੂੰ ਕਰ ਸਕਦੇ ਹਨ।

7.1.1 ਸਾਈਬਰ ਅਪਰਾਧ ਦੇ ਕਾਰਨ (Causes of Cyber Crime):

ਸਾਈਬਰ-ਅਪਰਾਧੀ ਹਮੇਸ਼ਾ ਵਿੱਤੀ ਸਰੋਤਾਂ, ਪ੍ਰਸਿੱਧੀ ਜਾਂ ਦੌਲਤ ਦੇ ਰੂਪ ਵਿੱਚ ਵਧੇਰੇ ਲਾਭ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਇੱਕ ਆਸਾਨ ਤਰੀਕਾ ਚੁਣਦੇ ਹਨ। ਬੈਂਕਾਂ, ਕੈਸੀਨੋ, ਕੰਪਨੀਆਂ, ਵਿੱਤੀ ਫਰਮਾਂ ਅਤੇ ਸਰਕਾਰੀ ਦਫਤਰਾਂ ਵਰਗੀਆਂ ਸੰਸਥਾਵਾਂ ਉਹਨਾਂ ਦੇ ਪ੍ਰਮੁੱਖ ਨਿਸ਼ਾਨਾ ਕੇਂਦਰ ਹੁੰਦੀਆਂ ਹਨ ਜਿੱਥੇ ਰੋਜ਼ਾਨਾਂ ਵੱਡੀ ਮਾਤਰਾ ਵਿੱਚ ਵਿੱਤੀ ਲੈਣ-ਦੇਣ ਹੁੰਦਾ ਹੈ ਜਾਂ ਸੰਵੇਦਨਸ਼ੀਲ ਜਾਣਕਾਰੀ ਹੁੰਦੀ ਹੈ। ਹਮਲਾਵਰਾਂ ਦਾ ਮੁੱਖ ਨਿਸ਼ਾਨਾ ਕਿਸੇ ਸਿਸਟਮ ਤੋਂ ਸੁਰੱਖਿਆ ਦੀ ਕੋਈ ਕਮੀ ਲੱਭਣਾ ਜਾਂ ਖਤਰਿਆਂ ਬਾਰੇ ਮਨੁੱਖਾਂ ਦੀ ਅਣਜਾਣਤਾ ਦਾ ਪਤਾ ਲਗਾਉਣਾ ਹੁੰਦਾ ਹੈ। ਦੁਨੀਆ ਭਰ ਵਿਚ ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਦੀ ਗਿਣਤੀ ਰੋਜ਼ਾਨਾ ਵਧ ਰਹੀ ਹੈ। ਇਸ ਸਮੱਸਿਆ ਦੇ ਹੱਲ ਤੇ ਹੀ ਸਾਈਬਰ-ਅਪਰਾਧੀਆਂ ਤੋਂ ਸੁਰੱਖਿਆ ਅਤੇ ਸੁਰੱਖਿਆ ਲਈ ਸਾਰੇ ਸੁਰੱਖਿਆ ਉਪਾਵਾਂ ਅਤੇ ਕਾਨੂੰਨਾਂ ਤੋਂ ਜਾਣੂ ਹੋਣਾ ਜ਼ਰੂਰੀ ਹੈ।

7.1.2 ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਦੀਆਂ ਕਿਸਮਾਂ (Types of Cyber Crimes):

ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਨੂੰ ਕਈ ਕਿਸਮਾਂ ਵਿੱਚ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਆਓ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਦੀਆਂ ਕੁਝ ਕਿਸਮਾਂ ਬਾਰੇ ਚਰਚਾ ਕਰੀਏ:

1. **ਵਿਅਕਤੀਆਂ ਦੇ ਖਿਲਾਫ ਅਪਰਾਧ (Crime against persons):** ਵਿਅਕਤੀਆਂ ਦੇ ਖਿਲਾਫ ਕੀਤੇ ਗਏ ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਅਪਰਾਧ ਸ਼ਾਮਲ ਹਨ ਜਿਵੇਂ ਕਿ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਚੋਰੀ ਕਰਨਾ, ਕੰਪਿਊਟਰ ਦੀ ਵਰਤੋਂ ਨਾਲ ਪਰੇਸ਼ਾਨ ਕਰਨਾ ਜਿਵੇਂ ਕਿ ਈ-ਮੇਲ, ਸਾਈਬਰ-ਸਟਾਕਿੰਗ, ਪਾਇਰੇਸੀ, ਪੋਸਟਿੰਗ ਜਾਂ ਅਸ਼ਲੀਲ ਸਮੱਗਰੀ ਦਾ ਪ੍ਰਸਾਰ। ਕਦੇ-ਕਦੇ ਬਹੁਤ ਸਾਰੇ ਸਾਈਬਰ ਅਪਰਾਧ ਨੌਜਵਾਨ ਪੀੜ੍ਹੀ ਨੂੰ ਬੁਰੀ ਤਰ੍ਹਾਂ ਪ੍ਰਭਾਵਿਤ ਕਰਦੇ ਹਨ ਅਤੇ ਨਾ ਪੂਰਣਯੋਗ ਅਸਰ ਛੱਡ ਜਾਂਦੇ ਹਨ।
2. **ਸਰਕਾਰ ਵਿਰੁੱਧ ਅਪਰਾਧ (Crime against Government):** ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਦੀ ਇਹ ਸ਼੍ਰੇਣੀ ਸਰਕਾਰ ਵਿਰੁੱਧ ਅਪਰਾਧਾਂ ਨਾਲ ਸਬੰਧਤ ਹੈ। ਸਾਈਬਰ ਅੱਤਵਾਦ ਇਸ ਸ਼੍ਰੇਣੀ ਵਿੱਚ ਇੱਕ ਵੱਖਰੀ ਕਿਸਮ ਦਾ ਅਪਰਾਧ ਹੈ। ਇੰਟਰਨੈੱਟ ਦੇ ਵਾਧੇ ਨੇ ਦਿਖਾਇਆ ਹੈ ਕਿ ਸਾਈਬਰ ਸਪੇਸ ਦੇ ਮਾਧਿਅਮ ਦੀ ਵਰਤੋਂ ਵਿਅਕਤੀਆਂ ਅਤੇ ਸਮੂਹਾਂ ਦੁਆਰਾ ਅੰਤਰਰਾਸ਼ਟਰੀ ਸਰਕਾਰਾਂ ਨੂੰ ਧਮਕੀ ਦੇਣ ਦੇ ਨਾਲ-ਨਾਲ ਕਿਸੇ ਦੇਸ਼ ਦੇ ਨਾਗਰਿਕਾਂ ਨੂੰ ਡਰਾਉਣ ਲਈ ਵੀ ਕੀਤੀ ਜਾ ਰਹੀ ਹੈ।
3. **ਜਾਇਦਾਦ ਦੇ ਵਿਰੁੱਧ ਅਪਰਾਧ (Crime against Property):** ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਦੀ ਤੀਜੀ ਸ਼੍ਰੇਣੀ ਜਾਇਦਾਦ ਦੇ ਸਾਰੇ ਰੂਪਾਂ ਵਿਰੁੱਧ ਸਾਈਬਰ ਅਪਰਾਧ ਹੈ। ਇਹਨਾਂ ਜੁਰਮਾਂ ਵਿੱਚ ਅਣਅਧਿਕਾਰਤ ਕੰਪਿਊਟਰ ਭੰਨਤੋੜ, ਹਾਨੀਕਾਰਕ ਪ੍ਰੋਗਰਾਮਾਂ ਦਾ ਪ੍ਰਸਾਰਣ ਅਤੇ ਜਾਣਕਾਰੀ ਤੇ ਅਣਅਧਿਕਾਰਕ ਕਬਜ਼ੇ ਵੀ ਸ਼ਾਮਲ ਹੈ।

ਆਉ ਇਹਨਾਂ ਸਾਰੀਆਂ ਕਿਸਮਾਂ ਦੇ ਸਾਈਬਰ ਖਤਰਿਆਂ ਬਾਰੇ ਵਿਸਥਾਰ ਵਿੱਚ ਚਰਚਾ ਕਰੀਏ:

1. **ਹੈਕਿੰਗ (HACKING):** ਹੈਕਿੰਗ ਖਤਰਨਾਕ ਹਮਲਾਵਰਾਂ ਦੀਆਂ ਰਣਨੀਤੀਆਂ ਅਤੇ ਕਾਰਵਾਈਆਂ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਕੰਪਿਊਟਰ ਸਿਸਟਮ, ਐਪਲੀਕੇਸ਼ਨ ਜਾਂ ਡਾਟਾ ਤੱਕ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ ਪ੍ਰਾਪਤ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਹੈ। ਇਸਦਾ ਮੁੱਖ ਉਦੇਸ਼ ਕੀਮਤੀ ਜਾਣਕਾਰੀ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਣ ਜਾਂ ਹੇਰਾਫੇਰੀ ਕਰਨ ਲਈ ਇਲੈਕਟ੍ਰਾਨਿਕ ਡਿਵਾਈਸਾਂ ਜਿਵੇਂ ਕਿ ਕੰਪਿਊਟਰ, ਸਮਾਰਟਫੋਨ, ਟੈਬਲੇਟ ਅਤੇ ਨੈਟਵਰਕ ਦੀ ਦੁਰਵਰਤੋਂ ਹੋ ਸਕਦਾ ਹੈ। ਹੈਕਿੰਗ ਵਿੱਚ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਜਾਂ ਨੈੱਟਵਰਕ ਵਿੱਚ ਸਰੋਤਾਂ ਦੀ ਪਛਾਣ ਕਰਨ ਅਤੇ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ ਕਰਨ ਲਈ ਕਾਰਵਾਈਆਂ ਦਾ ਇੱਕ ਸਮੂਹ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ। ਇਸ ਨੂੰ ਸਾਈਬਰ ਅਪਰਾਧੀਆਂ ਦੁਆਰਾ ਗੈਰ-ਕਾਨੂੰਨੀ ਗਤੀਵਿਧੀ ਅਤੇ ਡਾਟਾ ਚੋਰੀ ਮੰਨਿਆ ਜਾਂਦਾ ਹੈ। ਅਪਰਾਧੀ ਕਿਸੇ ਵਿਅਕਤੀ ਦੇ ਕੰਪਿਊਟਰ ਤੱਕ ਪਹੁੰਚ ਬਣਾਉਣ ਲਈ ਵੱਖ-ਵੱਖ ਸਾਫਟਵੇਅਰਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ ਅਤੇ ਵਿਅਕਤੀ ਨੂੰ ਸ਼ਾਇਦ ਇਹ ਪਤਾ ਨਾ ਹੋਵੇ ਕਿ ਉਸ ਦਾ ਕੰਪਿਊਟਰ ਹੈਕ ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਕਿਸੇ ਰਿਮੋਟ ਲੋਕੇਸ਼ਨ ਤੋਂ ਉਸ ਤੱਕ ਪਹੁੰਚ ਕੀਤੀ ਜਾ ਰਹੀ ਹੈ।
2. **ਜਾਸੂਸੀ (SPYING):** ਜਾਸੂਸੀ ਵਿਅਕਤੀਆਂ, ਸੰਸਥਾਵਾਂ, ਕੰਪਨੀਆਂ ਜਾਂ ਸੰਸਥਾਵਾਂ ਤੋਂ ਗੁਪਤ ਜਾਣਕਾਰੀ ਪ੍ਰਾਪਤ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰਨ ਦੀ ਇੱਕ ਗਤੀਵਿਧੀ ਹੈ। ਇਸ ਕਿਸਮ ਦੀਆਂ ਕਾਰਵਾਈਆਂ ਦੇ ਪਿੱਛੇ ਮੁੱਖ ਉਦੇਸ਼ ਲਾਭ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਡਾਟਾ ਜਾਂ ਨਿੱਜੀ ਸਮੱਗਰੀ ਦੀ ਦੁਰਵਰਤੋਂ ਤੇ ਕੇਂਦਰਿਤ ਹੈ। ਇਸ ਵਿੱਚ ਵਿਰੋਧੀਆਂ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਗਤੀਵਿਧੀਆਂ ਤੇ ਨਿੱਜੀ ਉਦੇਸ਼ਾਂ ਲਈ ਗੁਪਤ ਨਜ਼ਰ ਰੱਖਣਾ ਵੀ ਸ਼ਾਮਲ ਹੋ ਸਕਦਾ ਹੈ।

ਚਿੱਤਰ 7.1 **ਸਾਈਬਰ ਖਤਰੇ**

ਇਵਸਡ੍ਰੋਪਿੰਗ (Eavesdropping) ਕੁਝ ਮਹੱਤਵਪੂਰਨ ਜਾਂ ਗੁਪਤ ਜਾਣਕਾਰੀ ਇਕੱਠੀ ਕਰਨ ਲਈ ਕਿਸੇ ਦੀ ਸਹਿਮਤੀ ਤੋਂ ਬਿਨਾਂ ਦੂਜਿਆਂ ਦੀ ਨਿੱਜੀ ਗੱਲਬਾਤ ਜਾਂ ਸੰਚਾਰ ਨੂੰ ਗੁਪਤ ਜਾਂ ਚੋਰੀ-ਛਿਪੇ ਸੁਣਨ ਦਾ ਤਰੀਕਾ ਹੈ।

3. **ਫਿਸ਼ਿੰਗ (PHISHING):** ਫਿਸ਼ਿੰਗ ਇੱਕ ਅਜਿਹਾ ਹਮਲਾ ਹੈ ਜੋ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਜਿਵੇਂ ਕਿ ਕ੍ਰੈਡਿਟ ਕਾਰਡ ਨੰਬਰ, ਬੈਂਕ ਜਾਣਕਾਰੀ ਜਾਂ ਪਾਸਵਰਡ ਆਦਿ ਨੂੰ ਪ੍ਰਗਟ ਕਰਨ ਲਈ ਕੁਝ ਆਕਰਸ਼ਕ ਪੇਸ਼ਕਸ਼ਾਂ ਦੇ ਕੇ ਜਾਂ ਕਿਸੇ ਝੂਠੇ ਵੈੱਬ ਸਰੋਤ ਨੂੰ ਗੁੰਮਰਾਹ ਕਰਕੇ ਸਾਡੇ ਪੈਸੇ ਜਾਂ ਸਾਡੀ ਪਛਾਣ ਨੂੰ ਚੋਰੀ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰ ਸਕਦਾ ਹੈ। ਇਹ ਈ-ਮੇਲ ਭੇਜ ਕੇ ਕਿਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜੋ ਕਿਸੇ ਜਾਇਜ਼ ਸੰਸਥਾ ਤੋਂ ਹੋਣ ਦਾ ਝੂਠਾ ਦਾਅਵਾ ਕਰਦੀ ਹੈ। ਅਜਿਹੀ ਈਮੇਲ ਪ੍ਰਾਪਤ ਕਰਤਾ ਨੂੰ ਗੁਪਤ ਜਾਣਕਾਰੀ ਪ੍ਰਦਾਨ ਕਰਨ ਲਈ ਪ੍ਰੇਰਿਤ ਕਰਦੀ ਹੈ ਜਿਵੇਂ ਕਿ ਬੈਂਕ ਖਾਤੇ ਦੇ ਵੇਰਵੇ, ਪਿੰਨ ਜਾਂ ਪਾਸਵਰਡ ਅਤੇ ਇਹਨਾਂ ਵੇਰਵਿਆਂ ਦੀ ਵਰਤੋਂ ਵੈਬਸਾਈਟ ਦੇ ਮਾਲਕਾਂ ਦੁਆਰਾ ਧੋਖਾਧੜੀ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

4. **ਰੈਨਸਮਵੇਅਰ (RANSOMWARE):** ਰੈਨਸਮਵੇਅਰ ਇੱਕ ਕਿਸਮ ਦਾ ਮਾਲਵੇਅਰ ਹੈ ਜੋ ਪੀੜਤ ਦੇ ਡਾਟਾ ਜਾਂ ਡਿਵਾਈਸ ਨੂੰ ਲਾਕ ਕਰਦਾ ਹੈ ਅਤੇ ਇਸ ਨੂੰ ਲਾਕ ਰੱਖਣ ਦੀ ਧਮਕੀ ਦਿੰਦਾ ਹੈ ਜਦੋਂ ਤੱਕ ਪੀੜਤ ਹਮਲਾਵਰ ਨੂੰ ਫਿਰੌਤੀ ਅਦਾ ਨਹੀਂ ਕਰਦਾ। ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਹਮਲੇ ਕੀਮਤੀ ਡਾਟਾ ਨੂੰ ਇੱਕ ਗੁਪਤ ਕੋਡ ਵਿੱਚ ਬਦਲਦੇ ਹਨ ਅਤੇ ਡਾਟਾ ਨੂੰ ਉਹਨਾਂ ਦੇ ਅਸਲ ਰੂਪ ਵਿੱਚ ਲਿਆਉਣ ਲਈ ਉਹਨਾਂ ਤੋਂ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੇ ਲਾਭ ਦੀ ਮੰਗ ਕਰਦੇ ਹਨ। ਅਸੀਂ ਨਿਯਮਤ ਜਾਂ ਲਗਾਤਾਰ ਡਾਟਾ ਬੈਕਅੱਪ ਬਣਾ ਕੇ ਇਸ ਤਰ੍ਹਾਂ ਦੀਆਂ ਕਾਰਵਾਈਆਂ ਤੋਂ ਸੁਰੱਖਿਅਤ ਰਹਿ ਸਕਦੇ ਹਾਂ।

5. **ਟ੍ਰੋਲਿੰਗ (TROLLING):** ਟਰੋਲ ਉਹ ਯੂਜ਼ਰ ਹੁੰਦੇ ਹਨ ਜੋ ਜਨਤਕ ਟਿੱਪਣੀ ਕਰਨ ਵਾਲੀਆਂ ਜਗ੍ਹਾਵਾਂ ਵਿੱਚ ਭੜਕਾਊ ਟਿੱਪਣੀਆਂ ਕਰਦੇ ਹਨ। ਉਹ ਬਲੌਗ ਪੋਸਟਾਂ ਜਾਂ ਆਨਲਾਈਨ ਸੋਸ਼ਲ ਸਾਈਟਾਂ ਤੇ ਟਿੱਪਣੀ ਕਰਦੇ ਹਨ ਅਤੇ ਮੁੱਖ ਤੌਰ ਤੇ ਦੂਜੇ ਮਹਿਮਾਨਾਂ ਦਾ ਧਿਆਨ ਖਿੱਚਣ ਅਤੇ ਚਰਚਾ ਨੂੰ ਵਿਗਾੜਨ ਦਾ ਉਦੇਸ਼ ਰੱਖਦੇ ਹਨ। ਅਜਿਹੀਆਂ ਟਿੱਪਣੀਆਂ ਨਫ਼ਰਤ, ਨਸਲਵਾਦੀ ਜਾਂ ਅਪਮਾਨਜਨਕ ਹੋ ਸਕਦੀਆਂ ਹਨ। ਇਸ ਤਰ੍ਹਾਂ ਦੀਆਂ ਕਾਰਵਾਈਆਂ ਦਾ ਮੁੱਖ ਨਿਸ਼ਾਨਾ ਸਮਗਰੀ, ਲੇਖਕ ਜਾਂ ਹੋਰ ਟਿੱਪਣੀਕਾਰ ਬਣਦੇ ਹਨ।

6. **ਸਾਈਬਰ ਧੱਕੇਸ਼ਾਹੀ (CYBER BULLYING):** ਟ੍ਰੋਲ ਆਨਲਾਈਨ ਭਾਈਚਾਰਿਆਂ ਲਈ ਪਰੇਸ਼ਾਨੀ ਹੋਣ ਤੇ ਧਿਆਨ ਕੇਂਦਰਿਤ ਕਰਦੇ ਹਨ ਪਰੰਤੂ ਸਾਈਬਰ ਧੱਕੇਸ਼ਾਹੀ ਵਿਅਕਤੀਆਂ ਨੂੰ ਨਿਸ਼ਾਨਾ ਬਣਾਉਂਦੀ ਹੈ। ਆਮ ਤੌਰ ਤੇ ਭੜਕਾਊ ਬਿਆਨਾਂ ਨੂੰ ਪੋਸਟ ਕਰਨ ਦੀ ਬਜਾਏ ਉਹ ਨਿਰਾਦਰ ਜਾਂ ਡਰਾਉਣ ਦੇ ਟੀਚੇ ਨਾਲ ਇੱਕ ਵਿਅਕਤੀ ਬਾਰੇ ਭੜਕਾਊ ਗੱਲਾਂ ਪੋਸਟ ਕਰਦੇ ਹਨ। ਇਹ ਵਿਅਕਤੀ ਦੇ ਸੰਬੰਧ ਵਿੱਚ ਭਾਵੁਕ ਸੰਦੇਸ਼ਾਂ, ਨਿੱਜੀ ਤਸਵੀਰਾਂ ਜਾਂ ਨਿੱਜੀ ਵੀਡੀਓ ਦਾ ਰੂਪ ਵਿੱਚ ਹੋ ਸਕਦੀਆਂ ਹਨ। ਅਜਿਹੇ ਹਮਲਾਵਰ ਜਾਣਕਾਰੀ ਨੂੰ ਜਨਤਕ ਤੌਰ ਤੇ ਪੋਸਟ ਕਰ ਸਕਦੇ ਹਨ ਜਾਂ ਤਾਹਨੇ ਮਾਰਨ ਦੇ ਰੂਪ ਵਜੋਂ ਇਸਨੂੰ ਸਿਰਫ਼ ਆਪਣੇ ਨਿਸ਼ਾਨੇ ਤੇ ਭੇਜ ਸਕਦੇ ਹਨ।

ਸਾਈਬਰ ਧੱਕੇਸ਼ਾਹੀ ਆਪਣੇ ਪੀੜਤਾਂ ਨੂੰ ਅਪਮਾਨਿਤ ਕਰਨਾ ਅਤੇ ਦੁੱਖ ਪਹੁੰਚਾਉਣਾ ਚਾਹੁੰਦੇ ਹਨ। ਸਾਈਬਰਬੁਲੀਜ਼ ਆਪਣੇ ਲਈ ਧਿਆਨ ਨਹੀਂ ਚਾਹੁੰਦੇ, ਪਰ ਆਪਣੇ ਸ਼ਿਕਾਰ ਤੇ ਨਕਾਰਾਤਮਕ ਅਸਰ ਛੱਡਣਾ ਚਾਹੁੰਦੇ ਹਨ। ਉਹ ਸਿਰਫ਼ ਆਪਣੇ ਪੀੜਤਾਂ ਲਈ ਮੁਸੀਬਤ ਪੈਦਾ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਨ।

7. **ਸਾਈਬਰ ਸਟਾਕਿੰਗ (CYBER STALKING):** ਸਾਈਬਰ ਸਟਾਕਿੰਗ ਇੱਕ ਕਿਸਮ ਦਾ ਸਾਈਬਰ ਅਪਰਾਧ ਹੈ ਜੋ ਕਿਸੇ ਵਿਅਕਤੀ ਨੂੰ ਉਹਨਾਂ ਦੀਆਂ ਗਤੀਵਿਧੀਆਂ ਦਾ ਪਿੱਛਾ ਕਰਕੇ ਪਰੇਸ਼ਾਨ ਕਰਨ ਲਈ ਇੰਟਰਨੈਟ ਅਤੇ ਤਕਨਾਲੋਜੀ ਦੀ ਵਰਤੋਂ ਕਰਦਾ ਹੈ। ਇਸਨੂੰ ਸਾਈਬਰ ਧੱਕੇਸ਼ਾਹੀ ਅਤੇ ਵਿਅਕਤੀਗਤ ਤੌਰ ਤੇ ਪਿੱਛਾ ਕਰਨ ਦਾ ਇੱਕ ਵਿਸਥਾਰ ਮੰਨਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਸਟਾਕਰ ਪੀੜਤ ਦੀਆਂ ਕਾਰਵਾਈਆਂ ਨੂੰ ਆਨਲਾਈਨ ਟਰੈਕ ਕਰਦੇ ਹਨ ਅਤੇ ਉਹਨਾਂ ਦੀਆਂ ਗਤੀਵਿਧੀਆਂ ਜਾਂ ਹੋਰ ਨਿੱਜੀ ਚੀਜ਼ਾਂ ਦੀ ਵਰਤੋਂ ਆਪਣੇ ਨਿੱਜੀ ਮੰਤਵਾਂ ਲਈ ਕਰਦੇ ਹਨ। ਜੇਕਰ ਉਹ ਦੇਖਦੇ ਹਨ ਕਿ ਸਾਈਬਰ-ਸਟਾਕਿੰਗ ਦਾ ਕੋਈ ਅਸਰ ਨਹੀਂ ਹੋ ਰਿਹਾ ਹੈ ਤਾਂ ਉਹ ਸਾਈਬਰ-ਸਟਾਕਿੰਗ ਦੇ ਨਾਲ-

ਨਾਲ ਆਫ਼ ਲਾਈਨ ਪਿੱਛਾ ਕਰਨਾ ਸ਼ੁਰੂ ਕਰ ਦਿੰਦੇ ਹਨ ਤਾਂ ਜੋ ਇਹ ਯਕੀਨੀ ਬਣਾਇਆ ਜਾ ਸਕੇ ਕਿ ਪੀੜਤ ਤੇ ਵੱਧ ਤੋਂ ਵੱਧ ਅਸਰ ਹੋ ਸਕੇ। ਇਹ ਟੈਕਸਟ ਸੁਨੇਹਿਆਂ, ਈ-ਮੇਲਾਂ, ਸੋਸ਼ਲ ਮੀਡੀਆ ਪੋਸਟਾਂ ਅਤੇ ਹੋਰ ਮਾਧਿਅਮਾਂ ਦਾ ਰੂਪ ਲੈਂਦਾ ਹੈ। ਸਾਈਬਰ ਸਟਾਕਿੰਗ ਕਈ ਕਿਸਮਾਂ ਨਾਲ ਅਸਰਦਾਰ ਹੋ ਸਕਦੀ ਹੈ ਜਿਸ ਵਿੱਚ ਸ਼ਾਮਲ ਹਨ: ਪੀੜਤ ਨੂੰ ਪਰੇਸ਼ਾਨ ਕਰਨਾ, ਸ਼ਰਮਿੰਦਾ ਕਰਨਾ ਅਤੇ ਅਪਮਾਨਿਤ ਕਰਨਾ।

8. **ਪਾਇਰੇਸੀ (PIRACY):** ਪਾਇਰੇਸੀ ਇੱਕ ਕਿਸਮ ਦੀ ਸਾਹਿਤਕ ਚੋਰੀ ਹੈ ਜੋ ਕਿਸੇ ਹੋਰ ਵਿਅਕਤੀ ਦੇ ਕੰਮ ਜਾਂ ਵਿਚਾਰ ਨੂੰ ਨਿੱਜੀ ਰੂਪ ਵਜੋਂ ਪੇਸ਼ ਕਰਨ ਦਾ ਕੰਮ ਹੈ। ਇਹ ਅਪਰਾਧ ਉਦੋਂ ਵਾਪਰਦਾ ਹੈ ਜਦੋਂ ਕੋਈ ਵਿਅਕਤੀ ਕਾਪੀਰਾਈਟ ਦੀ ਉਲੰਘਣਾ ਕਰਦਾ ਹੈ ਅਤੇ ਸੰਗੀਤ, ਫ਼ਿਲਮਾਂ, ਗੇਮਾਂ ਅਤੇ ਸਾਫਟਵੇਅਰ ਡਾਊਨਲੋਡ ਕਰਦਾ ਹੈ। ਅੱਜ ਕੱਲ੍ਹ ਨਿਆਂ ਪ੍ਰਣਾਲੀ ਇਸ ਤਰ੍ਹਾਂ ਦੀਆਂ ਗਤੀਵਿਧੀਆਂ ਨੂੰ ਸਾਈਬਰ-ਅਪਰਾਧ ਵਜੋਂ ਸੰਬੋਧਿਤ ਕਰ ਰਹੀ ਹੈ ਅਤੇ ਬਹੁਤ ਸਾਰੇ ਕਾਨੂੰਨ ਹਨ ਜੋ ਲੋਕਾਂ ਨੂੰ ਪਾਇਰੇਸੀ ਤੋਂ ਰੋਕਦੇ ਹਨ। ਸਮੱਗਰੀ ਲੇਖਕ, ਫਿਲਮ ਨਿਰਮਾਤਾ ਅਤੇ ਨਿਰਦੇਸ਼ਕ ਅਕਸਰ ਇਸ ਅਪਰਾਧ ਦਾ ਨਿਸ਼ਾਨਾ ਬਣਦੇ ਹਨ।

9. **ਪਛਾਣ ਦੀ ਚੋਰੀ (IDENTITY THEFT):** ਪਛਾਣ ਚੋਰ ਆਮ ਤੌਰ ਤੇ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਜਿਵੇਂ ਕਿ ਪਾਸਵਰਡ, ਆਈਡੀ ਨੰਬਰ, ਕ੍ਰੈਡਿਟ ਕਾਰਡ ਨੰਬਰ ਪ੍ਰਾਪਤ ਕਰਨਾ ਹੁੰਦਾ ਹੈ ਅਤੇ ਪੀੜਤ ਦੇ ਨਾਮ ਤੇ ਧੋਖਾਧੜੀ ਨਾਲ ਕੰਮ ਕਰਨ ਲਈ ਉਹਨਾਂ ਦੀ ਦੁਰਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹਨਾਂ ਸੰਵੇਦਨਸ਼ੀਲ ਵੇਰਵਿਆਂ ਦੀ ਵਰਤੋਂ ਵੱਖ-ਵੱਖ ਗੈਰ-ਕਾਨੂੰਨੀ ਉਦੇਸ਼ਾਂ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ ਜਿਸ ਵਿੱਚ ਕਰਜ਼ੇ ਲਈ ਅਰਜ਼ੀ ਦੇਣਾ, ਆਨਲਾਈਨ ਖਰੀਦਦਾਰੀ ਕਰਨਾ, ਜਾਂ ਪੀੜਤ ਦੇ ਵਿੱਤੀ ਡਾਟਾ ਤੱਕ ਪਹੁੰਚ ਕਰਨਾ ਸ਼ਾਮਲ ਹੈ। ਇਸ ਨਾਲ ਪੀੜਤ ਨੂੰ ਮਹੱਤਵਪੂਰਨ ਆਰਥਿਕ ਨੁਕਸਾਨ ਹੋ ਸਕਦਾ ਹੈ।

ਪਛਾਣ ਦੀ ਚੋਰੀ, ਫਿਸ਼ਿੰਗ ਅਤੇ ਹੋਰ ਸੋਸ਼ਲ ਇੰਜਨੀਅਰਿੰਗ ਤਕਨੀਕਾਂ ਨਾਲ ਨੇੜਿਓਂ ਜੁੜੀ ਹੋਈ ਹੈ ਜੋ ਅਕਸਰ ਪੀੜਤ ਤੋਂ ਸੰਵੇਦਨਸ਼ੀਲ ਜਾਣਕਾਰੀ ਨੂੰ ਖੋਹਣ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ।

7.2 ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਈਡ (C.I.A. TRIAD)

ਸੀ.ਆਈ.ਏ. ਨੂੰ ਗੁਪਤਤਾ, ਅਖੰਡਤਾ ਅਤੇ ਉਪਲਬਧਤਾ ਮੰਨਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਨੂੰ ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਈਡ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਇੱਕ ਆਮ ਮਾਡਲ ਹੈ ਜੋ ਸੁਰੱਖਿਆ ਪ੍ਰਣਾਲੀਆਂ ਦੇ ਵਿਕਾਸ ਲਈ ਆਧਾਰ ਬਣਉਂਦਾ ਹੈ। ਉਹਨਾਂ ਦੀ ਵਰਤੋਂ ਸੀਮਾਵਾਂ ਅਤੇ ਉਹਨਾਂ ਦੇ ਹੱਲ ਲੱਭਣ ਦੇ ਤਰੀਕਿਆਂ ਦਾ ਪਤਾ ਲਗਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਕਿਸੇ ਕਾਰੋਬਾਰ ਜਾਂ ਕਿਸੇ ਵੀ ਸੰਸਥਾ ਦੇ ਸੰਚਾਲਨ ਲਈ ਗੁਪਤਤਾ, ਅਖੰਡਤਾ ਅਤੇ ਜਾਣਕਾਰੀ ਦੀ ਉਪਲਬਧਤਾ ਮਹੱਤਵਪੂਰਨ ਹੈ। ਇਹ ਸੁਰੱਖਿਆ ਟੀਮਾਂ ਨੂੰ ਮਾਰਗਦਰਸ਼ਨ ਕਰਨ ਵਿੱਚ ਮਦਦ ਕਰਦਾ ਹੈ ਕਿਉਂਕਿ ਉਹ ਵੱਖ-ਵੱਖ ਤਰੀਕਿਆਂ ਨੂੰ ਦਰਸਾਉਂਦੇ ਹਨ ਜਿਸ ਵਿੱਚ ਉਹ ਹਰੇਕ ਮੁਸੀਬਤ ਨੂੰ ਹੱਲ ਕਰ ਸਕਦੇ ਹਨ।

ਚਿੱਤਰ 7.2 ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਈਡ

ਸਿਧਾਂਤਕ ਤੌਰ ਤੇ ਜਦੋਂ ਸਾਰੇ ਤਿੰਨ ਮਾਪਦੰਡ ਪੂਰੇ ਹੋ ਜਾਂਦੇ ਹਨ ਤਾਂ ਸੰਗਠਨ ਦਾ ਸੁਰੱਖਿਆ ਪ੍ਰੋਫਾਈਲ ਮਜ਼ਬੂਤ ਹੁੰਦਾ ਹੈ ਅਤੇ ਖਤਰੇ ਨੂੰ ਸੰਭਾਲਣ ਲਈ ਬਿਹਤਰ ਢੰਗ ਮੁਹੱਈਆ ਕਰਾਉਂਦਾ ਹੈ। ਆਉ ਇਹਨਾਂ ਤਿੰਨਾਂ ਸ਼ਬਦਾਂ ਨੂੰ ਵਿਸਥਾਰ ਵਿੱਚ ਸਮਝੀਏ।

1. **ਗੁਪਤਤਾ (Confidentiality):** ਗੁਪਤਤਾ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਦੇ ਸਭ ਤੋਂ ਮਹੱਤਵਪੂਰਨ ਪਹਿਲੂਆਂ ਵਿੱਚੋਂ ਇੱਕ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਕੋਈ ਵੀ ਜਾਣਕਾਰੀ ਜੋ ਗੁਪਤ ਰੱਖਣ ਲਈ ਹੈ, ਨੂੰ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ ਤੋਂ ਸੁਰੱਖਿਅਤ ਰੱਖਿਆ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਡਾਟਾ ਨੂੰ ਗੁਪਤ ਮੰਨੇ ਜਾਣ ਲਈ ਇਸ ਨੂੰ ਕੁਝ ਮਾਪਦੰਡਾਂ ਨੂੰ ਪੂਰਾ ਕਰਨਾ

ਚਾਹੀਦਾ ਹੈ ਜਿਸ ਵਿੱਚ ਗੁਪਤ ਹੋਣਾ ਅਤੇ ਸੰਗਠਨ ਦੇ ਸਹੀ ਕੰਮਕਾਜ ਲਈ ਜ਼ਰੂਰੀ ਹੋਣਾ ਵੀ ਸ਼ਾਮਲ ਹੈ। ਗੁਪਤ ਰੱਖਣ ਵਾਲੀ ਜਾਣਕਾਰੀ ਨੂੰ ਇਸਦੀ ਸੰਵੇਦਨਸ਼ੀਲਤਾ ਦੇ ਅਨੁਸਾਰ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਗੁਪਤ ਮੰਨੇ ਜਾਣ ਵਾਲੇ ਡਾਟਾ ਲਈ ਵਿਸ਼ੇਸ਼ ਸੁਰੱਖਿਆ ਤਰੀਕਿਆਂ ਦੀ ਵੀ ਲੋੜ ਹੋ ਸਕਦੀ ਹੈ, ਜਿਵੇਂ ਕਿ ਪਾਸਵਰਡ-ਸੁਰੱਖਿਅਤ ਫਾਈਲਾਂ ਜਾਂ ਐਨਕ੍ਰਿਪਸ਼ਨ ਸਾਫਟਵੇਅਰ। ਸੰਸਥਾਵਾਂ ਨੂੰ ਆਪਣੇ ਡਾਟਾ ਨੂੰ ਅਣਅਧਿਕਾਰਤ ਪਹੁੰਚ ਤੋਂ ਬਚਾਉਣ ਲਈ ਹਮੇਸ਼ਾ ਕਦਮ ਚੁੱਕਣੇ ਚਾਹੀਦੇ ਹਨ ਅਤੇ ਜਦੋਂ ਸੰਵੇਦਨਸ਼ੀਲ ਜਾਣਕਾਰੀ ਦੀ ਸੁਰੱਖਿਆ ਦੀ ਗੱਲ ਆਉਂਦੀ ਹੈ ਤਾਂ ਸਖ਼ਤ ਤਰੀਕਿਆਂ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ। ਗੁਪਤਤਾ ਦੀ ਰੱਖਿਆ ਕਰਨਾ ਅੰਦਰੂਨੀ ਅਤੇ ਬਾਹਰੀ ਤੌਰ ਤੇ ਜਾਣਕਾਰੀ ਦੇ ਪਹੁੰਚ ਪੱਧਰਾਂ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਅਤੇ ਨਿਯੰਤਰਿਤ ਕਰਨ ਨਾਲ ਸੰਬੰਧਿਤ ਹੋ ਸਕਦਾ ਹੈ।

2. **ਇਕਸਾਰਤਾ (Integrity):** ਇਕਸਾਰਤਾ ਮਤਲਬ ਹੈ ਕਿ ਸਾਡੇ ਸਿਸਟਮ ਵਿੱਚ ਡਾਟਾ ਜਾਂ ਜਾਣਕਾਰੀ ਨੂੰ ਇੱਕਸਾਰ ਬਣਾਈ ਰੱਖਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਜੋ ਇਸਨੂੰ ਅਣਅਧਿਕਾਰਤ ਯੂਜ਼ਰਜ਼ ਦੁਆਰਾ ਸੋਧਿਆ ਜਾਂ ਮਿਟਾਇਆ ਨਾ ਜਾਵੇ। ਇਹ ਡਾਟਾ ਦੀ ਸਟੀਕਤਾ, ਭਰੋਸੇਯੋਗਤਾ ਅਤੇ ਸ਼ੁੱਧਤਾ ਦਾ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਤੱਤ ਹੈ। ਡਾਟਾ ਦੀ ਇਕਸਾਰਤਾ ਨੂੰ ਯਕੀਨੀ ਬਣਾਉਣ ਲਈ ਸਭ ਤੋਂ ਆਸਾਨ ਤਰੀਕੇ ਸਾਡੇ ਡਾਟਾ ਦਾ ਬੈਕਅੱਪ ਲੈਣਾ, ਪਹੁੰਚ ਨਿਯੰਤਰਣਾਂ ਦੀ ਵਰਤੋਂ ਕਰਨਾ, ਸਾਡੇ ਆਡਿਟ ਟ੍ਰੇਲ ਦੀ ਨਿਗਰਾਨੀ ਕਰਨਾ ਅਤੇ ਸਾਡੇ ਡਾਟਾ ਨੂੰ ਐਨਕ੍ਰਿਪਟ ਕਰਨਾ ਹੈ।

3. **ਉਪਲਬਧਤਾ (Availability):** ਸੀ.ਆਈ.ਏ ਟ੍ਰਾਈਡ ਦਾ ਅੰਤਿਮ ਹਿੱਸਾ ਉਪਲਬਧਤਾ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਸਿਸਟਮ ਅਤੇ ਡਾਟਾ ਸਾਰੇ ਸੰਬੰਧਤ ਵਿਅਕਤੀਆਂ ਲਈ ਉਪਲਬਧ ਹੁੰਦੇ ਹਨ ਜਦੋਂ ਉਹਨਾਂ ਨੂੰ ਕਿਸੇ ਵੀ ਸਥਿਤੀ ਵਿੱਚ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਇਸ ਵਿੱਚ ਬਿਜਲੀ ਬੰਦ ਹੋਣ ਜਾਂ ਕੁਦਰਤੀ ਆਫ਼ਤਾਂ ਸ਼ਾਮਲ ਹਨ। ਉਪਲਬਧਤਾ ਤੋਂ ਬਿਨਾਂ, ਭਾਵੇਂ ਅਸੀਂ CIA ਟ੍ਰਾਈਡ ਦੀਆਂ ਹੋਰ ਦੋ ਲੋੜਾਂ ਨੂੰ ਪੂਰਾ ਕਰ ਲਈਏ ਤਾਂ ਵੀ ਸਾਡੇ ਕਾਰੋਬਾਰ ਤੇ ਨਕਾਰਾਤਮਕ ਪ੍ਰਭਾਵ ਪੈ ਸਕਦਾ ਹੈ। ਉਪਲਬਧਤਾ ਨੂੰ ਯਕੀਨੀ ਬਣਾਉਣ ਲਈ, ਸੰਸਥਾ ਲੋੜੀਂਦੇ ਨੈੱਟਵਰਕ, ਸਰਵਰਾਂ ਅਤੇ ਐਪਲੀਕੇਸ਼ਨਾਂ ਦੀ ਵਰਤੋਂ ਕਰ ਸਕਦੀ ਹੈ। ਇਹਨਾਂ ਨੂੰ ਉਪਲਬਧ ਹੋਣ ਲਈ ਪ੍ਰੋਗਰਾਮ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜਦੋਂ ਪ੍ਰਾਇਮਰੀ ਸਿਸਟਮ ਵਿੱਚ ਕੋਈ ਦਿੱਕਤ ਆ ਜਾਂਦੀ ਹੈ। ਬੈਕਅੱਪ ਹੋਣ ਤੋਂ ਇਲਾਵਾ, ਆਈ.ਟੀ. ਆਰਕੀਟੈਕਟਰ ਦਾ ਡਿਜ਼ਾਈਨ ਵੀ ਅਹਿਮ ਭੂਮਿਤਾ ਨਿਭਾਉਂਦਾ ਹੈ।

7.3 ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਤੋਂ ਕਿਵੇਂ ਸੁਰੱਖਿਅਤ ਰਿਹਾ ਜਾਵੇ ? (How to be protected from Cyber Crimes)

ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਤੋਂ ਸੁਰੱਖਿਆ ਲਈ ਹੇਠ ਲਿਖੇ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼ਾਂ ਨੂੰ ਵਿਚਾਰ ਅਧੀਨ ਰੱਖਣਾ ਚਾਹੀਦਾ ਹੈ:

1. **ਇੱਕ ਸੁਰੱਖਿਅਤ ਕਨੈਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰੋ (Use a secure connection):** ਜੇਕਰ ਅਸੀਂ ਆਪਣੀ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਨੂੰ ਆਨਲਾਈਨ ਵਰਤਣ ਜਾ ਰਹੇ ਹਾਂ ਤਾਂ ਯਕੀਨੀ ਬਣਾਓ ਕਿ ਅਸੀਂ ਇੱਕ ਸੁਰੱਖਿਅਤ ਕੁਨੈਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰ ਰਹੇ ਹਾਂ। ਸਾਨੂੰ ਬਿਨਾਂ ਪਾਸਵਰਡ ਸੁਰੱਖਿਆ ਵਾਲੇ ਜਨਤਕ Wi-Fi ਤੋਂ ਬਚਣਾ ਚਾਹੀਦਾ ਹੈ।

2. **ਸਾਡੀਆਂ ਡਿਵਾਈਸਾਂ ਨੂੰ ਸੁਰੱਖਿਅਤ ਰੱਖੋ (Keep our devices secure):** ਭਰੋਸੇਯੋਗ, ਬਹੁ-ਪੱਧਰੀ, ਅੱਪ-ਟੂ-ਡੇਟ ਸੁਰੱਖਿਆ ਵਰਤ ਕੇ ਸਾਡੇ ਲੈਪਟਾਪ, ਸਮਾਰਟਫ਼ੋਨ ਅਤੇ ਟੈਬਲੈੱਟ ਨੂੰ ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰ ਅਤੇ ਹਮਲਾਵਰਾਂ ਤੋਂ ਬਚਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।

3. **ਸ਼ੱਕੀ ਸੰਦੇਸ਼ਾਂ ਅਤੇ ਸਾਈਟਾਂ ਤੋਂ ਦੂਰ ਰਹੋ (Stay away from suspicious messages and sites):** ਸ਼ੱਕੀ ਸਾਈਟਾਂ ਤੇ ਜਾਣ ਜਾਂ ਅਜਿਹੇ ਸੰਦੇਸ਼ਾਂ ਨੂੰ ਖੋਲ੍ਹਣ ਤੋਂ ਬਚੋ ਅਤੇ ਸੰਵੇਦਨਸ਼ੀਲ ਡਾਟਾ ਤੋਂ ਹੋਣ ਵਾਲੇ ਸੋਸ਼ਲ ਇੰਜਨੀਅਰਿੰਗ ਹਮਲਿਆਂ ਨੂੰ ਖੋਜਣ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ।

4. **ਮਜ਼ਬੂਤ ਪਾਸਵਰਡ ਦੀ ਵਰਤੋਂ ਕਰੋ (Use Strong passwords) ਲੰਬਾ:** ਮਜ਼ਬੂਤ ਪਾਸਵਰਡ ਬਣਾਓ ਜੋ ਵਿਲੱਖਨ ਹੋਵੇ ਤੇ ਜਿਸਦਾ ਅੰਦਾਜ਼ਾ ਲਗਾਉਣਾ ਔਖਾ ਹੋਵੇ। ਸਾਨੂੰ ਸਮੇਂ-ਸਮੇਂ ਤੇ ਪਾਸਵਰਡ ਬਦਲਦੇ ਰਹਿਣਾ ਚਾਹੀਦਾ ਹੈ। ਸਾਡੇ ਪਾਸਵਰਡ ਨੂੰ ਘੱਟ ਅਨੁਮਾਨਯੋਗ ਬਣਾਉਣ ਲਈ ਵਰਨਮਾਲਾ ਦੇ ਵੱਡੇ ਅਤੇ ਛੋਟੇ ਅੱਖਰਾਂ, ਸੰਖਿਆਵਾਂ ਅਤੇ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹਾਂ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ।

5. **ਬੈਂਕ ਅਤੇ ਕ੍ਰੈਡਿਟ ਖਾਤਿਆਂ ਦੀ ਨਿਗਰਾਨੀ ਕਰੋ (Monitor our bank and credit accounts) :**

ਸਾਨੂੰ ਸ਼ੱਕੀ ਗਤੀਵਿਧੀ ਲਈ ਨਿਯਮਿਤ ਤੌਰ ਤੇ ਆਪਣੇ ਆਨਲਾਈਨ ਬੈਂਕਿੰਗ ਖਾਤੇ ਅਤੇ ਕ੍ਰੈਡਿਟ ਸਕੋਰ ਦੀ ਜਾਂਚ ਕਰਦੇ ਰਹਿਣਾ ਚਾਹੀਦਾ ਹੈ।

6. **ਸੰਵੇਦਨਸ਼ੀਲ ਡਾਟਾ ਤੋਂ ਸਾਵਧਾਨ ਰਹੋ (Be careful with sensitive data):** ਜੇਕਰ ਅਸੀਂ ਕਿਸੇ ਵੀ ਭੌਤਿਕ ਦਸਤਾਵੇਜ਼ਾਂ ਨੂੰ ਸੁੱਟਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਜਿਸ ਵਿੱਚ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਹੁੰਦੀ ਹੈ, ਤਾਂ ਯਕੀਨੀ ਬਣਾਓ ਕਿ ਅਸੀਂ ਉਹਨਾਂ ਨੂੰ ਨਸ਼ਟ ਕਰਕੇ ਜਾਂ ਉਹਨਾਂ ਨੂੰ ਤੋੜ ਕੇ ਸੁਰੱਖਿਅਤ ਢੰਗ ਨਾਲ ਰੱਦ ਕਰਕੇ ਸੁੱਟੀਏ।

7. **ਬੇਲੋੜੇ ਸ਼ੇਅਰ ਤੋਂ ਬਚੋ (Don't overshare):** ਇਸ ਯੁੱਗ ਵਿੱਚ ਜਿੱਥੇ ਜ਼ਿਆਦਾਤਰ ਉਪਭੋਗਤਾਵਾਂ ਕੋਲ ਇੱਕ ਤੋਂ ਵੱਧ ਸੋਸ਼ਲ ਮੀਡੀਆ ਖਾਤੇ ਹਨ। ਓਵਰਸ਼ੇਅਰਿੰਗ ਇੱਕ ਗੰਭੀਰ ਸਮੱਸਿਆ ਹੋ ਸਕਦੀ ਹੈ। ਸਾਨੂੰ ਆਪਣੀ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਅਤੇ ਰੋਜ਼ਾਨਾ ਦੀ ਰੁਟੀਨ ਨੂੰ ਲੋਕਾਂ ਨਾਲ ਸਾਂਝਾ ਕਰਨ ਤੋਂ ਬਚਣਾ ਚਾਹੀਦਾ ਹੈ।

ਕੁਝ ਮਾਮਲਿਆਂ ਵਿੱਚ, ਅਪਰਾਧੀ ਦਾਅਵਾ ਕਰਦੇ ਹਨ ਕਿ ਪੀੜਤ ਨੇ ਨਸ਼ੀਲੇ ਪਦਾਰਥਾਂ ਜਾਂ ਨਕਲੀ ਪਾਸਪੋਰਟਾਂ ਵਰਗੇ ਗੈਰ-ਕਾਨੂੰਨੀ ਪਾਰਸਲ ਪ੍ਰਾਪਤ ਕੀਤੇ ਜਾਂ ਭੇਜੇ ਹਨ। ਉਹ ਪੀੜਤ ਦੇ ਰਿਸ਼ਤੇਦਾਰਾਂ ਜਾਂ ਦੋਸਤਾਂ ਨੂੰ ਵੀ ਸ਼ਾਮਲ ਕਰਨ ਦੀ ਧਮਕੀ ਵੀ ਦੇ ਸਕਦੇ ਹਨ।

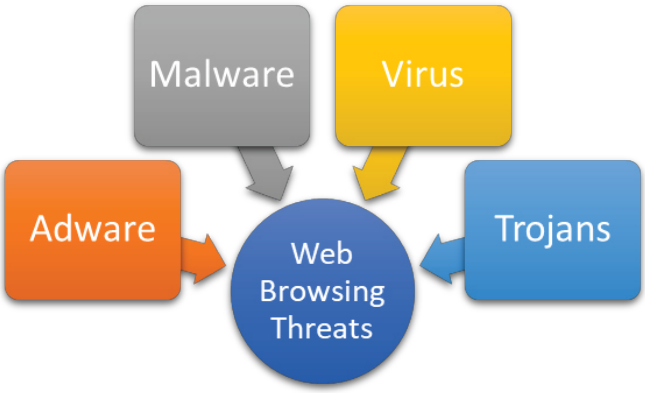
TEST YOURSELF ਉਚਿਤ ਬਕਸੇ 'ਤੇ ਨਿਸ਼ਾਨ ਲਗਾਓ:

	ਕਰੋ	ਨਾ ਕਰੋ
1. ਆਪਣਾ ਪਾਸਵਰਡ ਕਦੇ ਨਾ ਬਦਲੋ।		
2. ਸਾਰੇ ਖਾਤਿਆਂ 'ਤੇ ਗਤੀਵਿਧੀਆਂ ਦੀ ਨਿਗਰਾਨੀ ਕਰੋ		
3. ਗੋਪਨੀਯਤਾ ਅਤੇ ਡਾਟਾ ਦੀ ਉਪਲਬਧਤਾ ਨੂੰ ਯਕੀਨੀ ਬਣਾਓ।		
4. ਵੱਧ ਤੋਂ ਵੱਧ ਜਾਣਕਾਰੀ ਸਾਂਝੀ ਕਰੋ।		
5. ਆਪਣਾ ਪਾਸਵਰਡ ਸਰਲ ਅਤੇ ਆਸਾਨ ਰੱਖੋ।		
6. ਡਾਟਾ ਦੀ ਗੁਪਤਤਾ ਤੋਂ ਬਚੋ।		

7.4 ਵੈੱਬਸਾਈਟਾਂ ਤੱਕ ਸੁਰੱਖਿਅਤ ਢੰਗ ਨਾਲ ਪਹੁੰਚ ਕਰਨਾ (SAFELY ACCESSING WEBSITES)

ਵੱਖ-ਵੱਖ ਵੈੱਬਸਾਈਟਾਂ ਨੂੰ ਬ੍ਰਾਊਜ਼ ਕਰਦੇ ਸਮੇਂ ਯੂਜ਼ਰ ਨੂੰ ਵਾਇਰਸ, ਮਾਲਵੇਅਰ, ਸਪਾਈਵੇਅਰ ਆਦਿ ਸਮੇਤ ਵੱਖ-ਵੱਖ ਖਤਰਿਆਂ ਦਾ ਸਾਹਮਣਾ ਕਰਨਾ ਪੈਂਦਾ ਹੈ। ਹਾਲਾਂਕਿ, ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿੱਚ ਤੁਹਾਡੇ ਕੰਪਿਊਟਰ ਦੀ ਸੁਰੱਖਿਆ ਲਈ ਬਹੁਤ ਸਾਰੀਆਂ ਬਿਲਟ-ਇਨ ਸੁਰੱਖਿਆ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਹੁੰਦੀਆਂ ਹਨ। ਖਤਰਨਾਕ ਵੈੱਬਸਾਈਟਾਂ ਅਕਸਰ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਧੋਖਾ ਦੇਣ ਲਈ ਧੋਖੇਬਾਜ਼ ਵੈੱਬ ਪਤਿਆਂ ਦੀ ਵਰਤੋਂ ਕਰਦੀਆਂ ਹਨ। ਕੁਝ ਵੈੱਬਸਾਈਟਾਂ ਐਡਰੈੱਸ ਬਾਰ ਵਿੱਚ ਇੱਕ ਲਾਕ ਚਿੰਨ੍ਹ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨਗੀਆਂ। ਇਹ ਆਮ ਤੌਰ ਤੇ ਕੁਝ ਖਾਸ ਕਿਸਮ ਦੀਆਂ ਵੈੱਬਸਾਈਟਾਂ, ਜਿਵੇਂ ਕਿ ਆਨਲਾਈਨ ਸਟੋਰਾਂ ਅਤੇ ਬੈਂਕਿੰਗ ਸਾਈਟਾਂ ਨਾਲ ਦੇਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਸਦਾ ਮਤਲਬ ਹੈ ਕਿ ਵੈੱਬਸਾਈਟ ਸੁਰੱਖਿਅਤ ਲੇਅਰਡ ਕਨੈਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰ ਰਹੀ ਹੈ। ਜੋ ਸਾਡੀ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਨੂੰ ਦਾਖਲ ਕਰਨਾ ਸੁਰੱਖਿਅਤ ਬਣਾਉਂਦੀ ਹੈ।

ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਿੰਗ ਦੌਰਾਨ ਕਈ ਸੁਰੱਖਿਆ ਖਤਰੇ ਹੁੰਦੇ ਹਨ ਜਿਨ੍ਹਾਂ ਤੋਂ ਸੁਰੱਖਿਅਤ ਰਹਿਣ ਲਈ ਸਤਰਕ ਰਹਿਣਾ ਬਹੁਤ ਜ਼ਰੂਰੀ ਹੈ। ਇਹਨਾਂ ਵਿੱਚੋਂ ਕੁਝ ਖਤਰੇ ਹੇਠਾਂ ਦਰਸਾਏ ਗਏ ਹਨ:



ਚਿੱਤਰ 7.3 ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਿੰਗ ਖ਼ਤਰੇ

1. **ਐਡਵੇਅਰ (Adware):** ਇਹ ਆਮ ਤੌਰ ਤੇ ਅਣਚਾਹੇ ਸਾਫਟਵੇਅਰ ਦੀ ਇੱਕ ਕਿਸਮ ਹੈ ਜੋ ਤੁਹਾਡੇ ਕੰਪਿਊਟਰ ਜਾਂ ਮੋਬਾਈਲ ਡਿਵਾਈਸ ਤੇ ਪੌਪ-ਅੱਪ ਇਸ਼ਤਿਹਾਰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ। ਇਹ ਅਸਲ ਵਿੱਚ ਮਾਰਕੀਟਿੰਗ ਲਈ ਵਿਕਸਤ ਕੀਤਾ ਗਿਆ ਹੈ। ਜੇਕਰ ਖਤਰਨਾਕ ਕੋਡ ਨੂੰ ਸਪਾਈਵੇਅਰ ਵਿੱਚ ਸ਼ਾਮਿਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਇਸਦੇ ਨੁਕਸਾਨਦੇਹ ਪ੍ਰਭਾਵ ਵੀ ਹੋ ਸਕਦੇ ਹਨ। ਐਡਵੇਅਰ ਦੀ ਰੋਕਥਾਮ ਲਈ ਸੁਚੇਤ ਰਹਿਣਾ ਇਸ ਨੂੰ ਹਟਾਉਣ ਨਾਲੋਂ ਬਹੁਤ ਲਾਭਦਾਇਕ ਹੈ। ਇਸ ਲਈ ਕੂਕੀਜ਼ ਨੂੰ ਸਵੀਕਾਰ ਕਰਦੇ ਸਮੇਂ ਜਾਂ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਵੇਲੇ ਅਨੁਮਤੀਆਂ ਦੇਣ ਵੇਲੇ ਹਮੇਸ਼ਾ ਧਿਆਨ ਰੱਖੋ।
2. **ਮਾਲਵੇਅਰ (Malware):** ਮਾਲਵੇਅਰ ਨੂੰ ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ ਜੋ ਸਾਈਬਰ ਅਪਰਾਧੀਆਂ ਦੁਆਰਾ ਵਿਕਸਤ ਕੀਤੇ ਕਿਸੇ ਵੀ ਘੁਸਪੈਠ ਵਾਲੇ ਸਾਫਟਵੇਅਰ ਹੋ ਸਕਦੇ ਹਨ। ਇਸ ਕਿਸਮ ਦੇ ਸਾਫਟਵੇਅਰਾਂ ਨੂੰ ਹੈਕਰਾਂ ਦੁਆਰਾ ਡਾਟਾ ਚੋਰੀ ਕਰਨ ਅਤੇ ਡਿਜੀਟਲ ਸੰਪਤੀਆਂ ਜਾਂ ਕੰਪਿਊਟਰ ਪ੍ਰਣਾਲੀਆਂ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਣ ਜਾਂ ਨਸ਼ਟ ਕਰਨ ਲਈ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਹੁੰਦਾ ਹੈ। ਆਮ ਮਾਲਵੇਅਰ ਦੀਆਂ ਉਦਾਹਰਨਾਂ ਵਿੱਚ ਸ਼ਾਮਲ ਹਨ ਵਾਇਰਸ, ਵਾਰਮਜ਼, ਟਰੋਜਨ, ਸਪਾਈਵੇਅਰ, ਐਡਵੇਅਰ ਅਤੇ ਰੈਨਸਮਵੇਅਰ ਆਦਿ।
3. **ਵਾਇਰਸ (Virus):** ਇੱਕ ਕੰਪਿਊਟਰ ਵਾਇਰਸ ਹਾਨੀਕਾਰਕ ਪ੍ਰੋਗਰਾਮ ਦੀ ਇੱਕ ਸ਼੍ਰੇਣੀ ਹੈ। ਜਦੋਂ ਇਹ ਪ੍ਰੋਗਰਾਮ ਚਲਾਏ ਜਾਂਦੇ ਹਨ ਤਾਂ ਇਹ ਆਪਣੇ ਆਪ ਦੀਆਂ ਕਾਪੀਆਂ ਬਣਾਉਂਦਾ ਹੈ ਅਤੇ ਸਾਡੇ ਕੰਪਿਊਟਰ ਦੇ ਹੋਰ ਪ੍ਰੋਗਰਾਮਾਂ ਅਤੇ ਫਾਈਲਾਂ ਨੂੰ ਬਦਲਦਾ ਹੈ। ਇਹ ਵਾਇਰਸ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਵਿੱਚ ਆਉਂਦੇ ਹਨ ਅਤੇ ਹਰੇਕ ਕਿਸਮ ਸਾਡੀ ਡਿਵਾਈਸ ਨੂੰ ਵੱਖਰੇ ਢੰਗ ਨਾਲ ਪ੍ਰਭਾਵਿਤ ਕਰ ਸਕਦੀ ਹੈ। ਇੱਕ ਕੰਪਿਊਟਰ ਵਾਇਰਸ ਦਾ ਉਦੇਸ਼ ਦੂਜੇ ਕੰਪਿਊਟਰਾਂ ਵਿੱਚ ਵੀ ਫੈਲਣਾ ਹੁੰਦਾ ਹੈ। ਇਹ ਆਪਣੇ ਆਪ ਨੂੰ ਸਧਾਰਨ ਪ੍ਰੋਗਰਾਮਾਂ ਜਾਂ ਦਸਤਾਵੇਜਾਂ ਨਾਲ ਜੋੜ ਕੇ ਅਜਿਹਾ ਕਰਦਾ ਹੈ, ਜਿਸ ਦੌਰਾਨ ਇਹਨਾਂ ਦਾ ਕੋਡ ਚਲਾਇਆ ਜਾ ਸਕਦਾ ਹੋਵੇ। ਅਜਿਹੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਮੈਕਰੋ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਵਾਇਰਸ ਦੀ ਵਰਤੋਂ ਕਰਨ ਪਿੱਛੇ ਮੁੱਖ ਉਦੇਸ਼ ਨਿੱਜੀ ਲਾਭਾਂ ਲਈ ਕੀਮਤੀ ਡਾਟਾ ਜਾਂ ਉਪਯੋਗੀ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਨਸ਼ਟ ਕਰਨਾ ਹੁੰਦਾ ਹੈ।
4. **ਟਰੋਜਨ (Trojans):** ਇੱਕ ਟਰੋਜਨ ਜਾਂ "ਟਰੋਜਨ ਹੋਰਸ" ਮਾਲਵੇਅਰ ਦੀ ਇੱਕ ਕਿਸਮ ਹੈ ਜੋ ਯੂਜ਼ਰ ਨੂੰ ਹਮਲਾਵਰਾਂ ਦੇ ਅਸਲ ਇਰਾਦੇ ਨੂੰ ਪ੍ਰਗਟ ਕੀਤੇ ਬਿਨਾਂ ਚਾਲਬਾਜ਼ ਤਰੀਕੇ ਰਾਹੀਂ ਇੱਕ ਜਾਇਜ਼ ਪ੍ਰੋਗਰਾਮ ਦਾ ਰੂਪ ਲੈ ਲੈਂਦਾ ਹੈ। ਟ੍ਰੋਜਨਾਂ ਨੂੰ ਐਪਸ, ਗੇਮਾਂ, ਟੂਲਸ ਜਾਂ ਸਾਫਟਵੇਅਰ ਫਾਈਲਾਂ ਲਈ ਡਾਊਨਲੋਡਾਂ ਵਿੱਚ ਪੈਕ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਉਹ ਉਪਭੋਗਤਾ ਦੀ ਜਾਣਕਾਰੀ ਤੋਂ ਬਿਨਾਂ ਫਾਈਲਾਂ ਨੂੰ ਮਿਟਾਉਣ ਜਾਂ ਸੋਧਣ, ਫਾਈਲਾਂ ਨੂੰ ਨਿਰਯਾਤ ਕਰਨ, ਜਾਂ ਡਿਵਾਈਸ ਦੀ ਸਮੱਗਰੀ ਨੂੰ ਬਦਲਣ ਵਰਗੀਆਂ ਕਾਰਵਾਈਆਂ ਕਰ ਸਕਦੇ ਹਨ। ਟਰੋਜਨ ਇੱਕ ਕੰਪਿਊਟਰ ਵਿੱਚ ਇੱਕ ਅਣਅਧਿਕਾਰਤ ਰਸਤਾ ਵੀ ਖੋਲ੍ਹ ਸਕਦੇ ਹਨ, ਜਿਸ ਰਾਹੀਂ ਹਮਲਾਵਰ ਖਤਰਨਾਕ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਸਿਸਟਮ ਤੱਕ ਪਹੁੰਚ ਕਰਨ ਅਤੇ ਨਿੱਜੀ ਅਤੇ ਗੁਪਤ ਜਾਣਕਾਰੀ ਚੋਰੀ ਕਰਨ ਦੀ ਇਜਾਜ਼ਤ ਦਿੱਤੀ ਜਾਂਦੀ ਹੈ। ਸਾਡੇ ਸਿਸਟਮ ਨੂੰ ਟਰੋਜਨ ਹੋਰਸ ਤੋਂ ਸੁਰੱਖਿਅਤ ਰੱਖਣ ਲਈ ਸਾਨੂੰ ਕਦੇ ਵੀ ਅਜਿਹੇ ਸਰੋਤ ਤੋਂ ਸਾਫਟਵੇਅਰ ਡਾਊਨਲੋਡ ਜਾਂ ਇੰਸਟਾਲ ਨਹੀਂ ਕਰਨਾ ਚਾਹੀਦਾ ਜਿਸ ਤੇ ਸਾਨੂੰ ਭਰੋਸਾ ਨਾ ਹੋਵੇ। ਕਦੇ ਵੀ ਕੋਈ ਅਟੈਚਮੈਂਟ ਨਾ ਖੋਲ੍ਹੋ ਜਾਂ ਅਣਜਾਣ ਜਾਂ ਗੈਰ-ਭਰੋਸੇਯੋਗ ਸਰੋਤਾਂ ਤੋਂ ਈਮੇਲ ਰਾਹੀਂ ਪ੍ਰਾਪਤ ਕੀਤਾ ਪ੍ਰੋਗਰਾਮ ਨਾ ਚਲਾਓ।

TEST YOURSELF

ਢੁਕਵੇਂ ਵਿਕਲਪ ਦਾ ਮੇਲ ਕਰੋ:

A	B
1. ਵਾਇਰਸ	ਓ) ਸਮੇਂ-ਸਮੇਂ 'ਤੇ ਇਸ਼ਤਿਹਾਰਾਂ ਨੂੰ ਪੌਪ-ਅੱਪ ਕਰਦਾ ਹੈ
2. ਟਰੋਜਨ	ਅ) ਖਤਰਨਾਕ ਸਾਫਟਵੇਅਰਾਂ ਦਾ ਸਮੂਹ
3. ਐਡਵੇਅਰ	ਏ) ਆਪਣੇ ਆਪ ਨੂੰ ਫੈਲਾਉਣ ਲਈ ਦੁਹਰਾਉਂਦਾ ਹੈ
4. ਮਾਲਵੇਅਰ	ਸ) ਸਾਈਬਰ ਧਮਕੀਆਂ ਤੋਂ ਸੁਚੇਤ ਰਹਿਣ ਲਈ
5. ਸੁਰੱਖਿਅਤ ਬ੍ਰਾਊਜ਼ਿੰਗ	ਹ) ਇਰਾਦੇ ਨੂੰ ਪ੍ਰਗਟ ਕੀਤੇ ਬਿਨਾਂ ਰਹਿਣਾ

7.5 ਸਾਈਬਰ ਨੈਤਿਕਤਾ (CYBER ETHICS)

ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਇੱਕ ਉਪਭੋਗਤਾ ਦੁਆਰਾ ਆਨਲਾਈਨ ਸਰੋਤਾਂ ਦੀ ਸਵੀਕਾਰਯੋਗ ਵਰਤੋਂ ਦੇ ਨਿਯਮਾਂ ਦਾ ਸਮੂਹ ਹੁੰਦੇ ਹਨ। ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਇੰਟਰਨੈੱਟ ਤੇ ਜ਼ਿੰਮੇਵਾਰ ਦੇ ਕੋਡ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਇਹ ਸੁਨਿਸ਼ਚਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਕਿ ਉਪਭੋਗਤਾ ਆਪਣੇ ਆਪ ਨੂੰ ਆਨਸਾਈਨ ਕਰਨ ਲਈ ਆਪਣੀਆਂ ਜ਼ਿੰਮੇਵਾਰੀਆਂ ਨੂੰ ਸਮਝਦੇ ਹਨ। ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਦੇ ਚਾਰ ਬੁਨਿਆਦੀ ਸਿਧਾਂਤ ਹਨ।

- **ਗੋਪਨੀਯਤਾ (Privacy):** ਇਹ ਵਿਅਕਤੀਆਂ ਨਾਲ ਸਬੰਧਤ ਡਾਟਾ ਨੂੰ ਸੁਰੱਖਿਅਤ ਰੱਖਣ ਦੀ ਜ਼ਿੰਮੇਵਾਰੀ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।
- **ਸ਼ੁੱਧਤਾ (Accuracy):** ਜਾਣਕਾਰੀ ਨੂੰ ਪ੍ਰਮਾਣਿਤ ਕਰਨਾ ਅਤੇ ਇਸਦੀ ਸ਼ੁੱਧਤਾ ਨੂੰ ਯਕੀਨੀ ਬਣਾਉਣਾ ਡਾਟਾ ਇਕੱਤਰ ਕਰਨ ਵਾਲਿਆਂ ਦੀ ਜ਼ਿੰਮੇਵਾਰੀ ਹੈ।
- **ਜਾਇਦਾਦ (Property):** ਇਹ ਇਸ ਗੱਲ ਨੂੰ ਸੁਨਿਸ਼ਚਿਤ ਕਰਦਾ ਹੈ ਕਿ ਜਾਣਕਾਰੀ/ਸਾਫਟਵੇਅਰ ਦਾ ਮਾਲਕ ਕੌਣ ਹੈ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਕਿਵੇਂ ਵੇਚਿਆ ਜਾਂ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ।
- **ਪਹੁੰਚ (Access):** ਇਸ ਵਿੱਚ ਸਰੋਤ ਦੀ ਪਹੁੰਚ ਨੂੰ ਨਿਯੰਤਰਿਤ ਕਰਨ ਅਤੇ ਨਿਰਧਾਰਤ ਕਰਨ ਲਈ ਡਾਟਾ ਇਕੱਤਰ ਕਰਨ ਵਾਲਿਆਂ ਦੀ ਜ਼ਿੰਮੇਵਾਰੀ ਸ਼ਾਮਲ ਹੋ ਕਿ ਕਿਸੇ ਵਿਅਕਤੀ ਨੂੰ ਦੂਜਿਆਂ ਬਾਰੇ ਕਿਹੜੀ ਜਾਣਕਾਰੀ ਪ੍ਰਾਪਤ ਕਰਨ ਦਾ ਅਧਿਕਾਰ ਹੈ ਅਤੇ ਜਾਣਕਾਰੀ ਦੀ ਵਰਤੋਂ ਕਿਵੇਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

ਜਦੋਂ ਅਸੀਂ ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਬਾਰੇ ਗੱਲ ਕਰਦੇ ਹਾਂ ਤਾਂ ਸਾਨੂੰ ਇਹ ਯਕੀਨੀ ਬਣਾਉਣਾ ਚਾਹੀਦਾ ਹੈ ਕਿ ਇਹਨਾਂ ਸਾਰੇ ਸਿਧਾਂਤਾਂ ਦੀ ਪਾਲਣਾ ਕੀਤੀ ਜਾਵੇ। ਇਹ ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਤੋਂ ਸੁਰੱਖਿਆ ਸੰਬੰਧੀ ਕੁਝ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼ ਹਨ।

7.6 ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ (E-WASTE MANAGEMENT)

ਈ-ਵੇਸਟ ਦਾ ਅਰਥ ਹੈ ਇਲੈਕਟ੍ਰਾਨਿਕ ਵੇਸਟ। ਇਸ ਵਿੱਚ ਰੱਦ ਕੀਤੇ ਗਏ ਇਲੈਕਟ੍ਰੀਕਲ ਜਾਂ ਇਲੈਕਟ੍ਰਾਨਿਕ ਉਪਕਰਨ ਜਾਂ ਹਿੱਸੇ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਜਦੋਂ ਵੀ ਇਹਨਾਂ ਡਿਵਾਈਸਾਂ ਦੀ ਕਾਰਜਸ਼ੀਲਤਾ ਦੀ ਮਿਆਦ ਖਤਮ ਹੋ ਜਾਂਦੀ ਹੈ ਜਾਂ ਇਹ ਡਿਵਾਈਸਜ ਖਰਾਬ ਹੋ ਜਾਂਦੀਆਂ ਹਨ ਜਾਂ ਤਕਨੀਕੀ ਤਰੱਕੀ ਦੇ ਕਾਰਨ ਉਪਭੋਗਤਾ ਦੁਆਰਾ ਇਸਦੀ ਵਰਤੋਂ ਨਹੀਂ ਕੀਤੀ ਜਾਂਦੀ ਤਾਂ ਇਹ ਉਪਕਰਣ ਜਾਂ ਉਪਕਰਣਾਂ ਦੇ ਹਿੱਸੇ ਈ-ਵੇਸਟ ਦੇ ਅਧੀਨ ਆਉਂਦੇ ਹਨ। ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਜਾਣਦੇ ਹਾਂ, ਤਕਨਾਲੋਜੀ ਰੋਜਾਨਾ ਬਦਲਦੀ ਰਹਿੰਦੀ ਹੈ ਜਿਸ ਕਾਰਨ ਵੱਡੀ ਮਾਤਰਾ ਵਿੱਚ ਇਲੈਕਟ੍ਰਾਨਿਕ ਜਾਂ ਇਲੈਕਟ੍ਰੀਕਲ ਉਪਕਰਣ ਈ-ਵੇਸਟ ਵਿੱਚ ਬਦਲ ਰਹੇ ਹਨ। ਕੁਝ ਆਮ ਈ-ਵੇਸਟ ਤੱਤ ਹਨ ਮੋਬਾਈਲ ਫੋਨ, ਕੰਪਿਊਟਰ, ਲੈਪਟਾਪ, ਹਾਰਡ ਡਰਾਈਵ, ਪੱਖੇ, ਮਾਈਕ੍ਰੋਵੇਵ,

ਚਿੱਤਰ 7.4 ਈ-ਵੇਸਟ

ਡੀ.ਵੀ.ਡੀ. (DVD), ਪ੍ਰਿੰਟਰ, ਲੈਂਪ, ਆਦਿ। ਈ-ਵੇਸਟ ਸਾਡੇ ਵਾਤਾਵਰਣ ਲਈ ਇੱਕ ਗੰਭੀਰ ਮੁੱਦਾ ਹੈ ਕਿਉਂਕਿ ਇਹ ਧਾਤਾਂ ਤੋਂ ਹਾਨੀਕਾਰਕ ਜ਼ਹਿਰੀਲੇ ਰਸਾਇਣ ਛੱਡਦਾ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਦੀ ਰਹਿੰਦ-ਖੂੰਹਦ ਦੇ ਇੱਕ ਯੋਜਨਾਬੱਧ ਪ੍ਰਬੰਧਨ ਨੂੰ ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਨੂੰ ਵਾਤਾਵਰਣ ਤੇ ਇਸ ਦੇ ਹਾਨੀਕਾਰਕ ਜ਼ਹਿਰੀਲੇ ਪ੍ਰਭਾਵਾਂ ਨੂੰ ਰੋਕਣ ਲਈ ਇਸ ਨੂੰ ਨਸ਼ਟ ਕਰਨ ਦੇ ਇੱਕ ਸੰਪੂਰਨ ਢੰਗ ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਉਹਨਾਂ ਉਪਕਰਨਾਂ (ਈ-ਵੇਸਟ) ਨੂੰ ਰੀਸਾਈਕਲ ਅਤੇ ਰੀਯੂਜ਼ (ਮੁੜ ਵਰਤੋਂ) ਕਰਦਾ ਹੈ ਜਿਨ੍ਹਾਂ ਦੀ ਹੋਰ ਲੋੜ ਨਹੀਂ ਹੁੰਦੀ ਹੈ।

7.6.1 ਵਾਤਾਵਰਣ ਤੇ ਈ-ਵੇਸਟ ਦੇ ਪ੍ਰਭਾਵ (Effects of E-Waste on Environment):

ਜਦੋਂ ਈ-ਵੇਸਟ ਦਾ ਸਹੀ ਢੰਗ ਨਾਲ ਪ੍ਰਬੰਧਨ ਨਹੀਂ ਕੀਤਾ ਜਾਂਦਾ ਤਾਂ ਇਹ ਵਾਤਾਵਰਣ ਅਤੇ ਲੋਕਾਂ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਂਦਾ ਹੈ। ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਤੋਂ ਬਿਨਾਂ ਵਾਤਾਵਰਣ ਨੂੰ ਵੱਖ-ਵੱਖ ਪਹਿਲੂਆਂ ਤੋਂ ਪ੍ਰਭਾਵਿਤ ਕਰ ਸਕਦਾ ਹੈ। ਆਓ ਇਹਨਾਂ ਪ੍ਰਭਾਵਾਂ ਤੇ ਨੇੜਿਓਂ ਨਜ਼ਰ ਮਾਰੀਏ:

1. **ਹਵਾ ਪ੍ਰਦੂਸ਼ਣ (Air Pollution):** ਈ-ਵੇਸਟ ਦੇ ਗਲਤ ਪ੍ਰਬੰਧਨ ਕਾਰਨ ਹਵਾ ਪ੍ਰਦੂਸ਼ਣ ਇੱਕ ਪ੍ਰਮੁੱਖ ਚਿੰਤਾ ਦਾ ਵਿਸ਼ਾ ਹੈ। ਜਦੋਂ ਈ-ਵੇਸਟ ਨੂੰ ਗੈਰ-ਰਸਮੀ ਤੌਰ ਤੇ ਸਮੱਗਰੀ ਨੂੰ ਤੋੜਨਾ, ਕੱਟਣਾ ਜਾਂ ਪਿਘਲਾ ਕੇ ਨਿਪਟਾਇਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ ਧੂੜ ਦੇ ਬਾਰੀਕ ਕਣ ਜਾਂ ਜ਼ਹਿਰੀਲੀਆਂ ਗੈਸਾਂ ਛੱਡਦਾ ਹੈ ਜੋ ਹਜ਼ਾਰਾਂ ਮੀਲ ਦੀ ਯਾਤਰਾ ਕਰ ਸਕਦੇ ਹਨ ਅਤੇ ਹਵਾ ਪ੍ਰਦੂਸ਼ਣ ਦਾ ਕਾਰਨ ਬਣ ਸਕਦੇ ਹਨ। ਇਸ ਤਰ੍ਹਾਂ ਦਾ ਹਵਾ ਪ੍ਰਦੂਸ਼ਣ ਸਾਰੇ ਵਨਸਪਤੀ ਅਤੇ ਜੀਵ-ਜੰਤੂਆਂ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਂਦਾ ਹੈ।
2. **ਮਿੱਟੀ ਪ੍ਰਦੂਸ਼ਣ (Soil Pollution):** ਜਦੋਂ ਈ-ਵੇਸਟ ਨੂੰ ਨਿਯਮਤ ਜ਼ਮੀਨ ਵਿੱਚ ਖੁੱਲ੍ਹੇ ਤੌਰ ਤੇ ਨਸ਼ਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ ਉਸ ਮਿੱਟੀ ਨੂੰ ਦੂਸ਼ਿਤ ਕਰ ਦਿੰਦਾ ਹੈ। ਹੁਣ ਜਦੋਂ ਇਸ ਦੂਸ਼ਿਤ ਜ਼ਮੀਨ ਤੇ ਫ਼ਸਲ ਬੀਜੀ ਜਾਂਦੀ ਹੈ ਤਾਂ ਫ਼ਸਲ ਤੇ ਇਨ੍ਹਾਂ ਜ਼ਹਿਰਾਂ ਦਾ ਅਸਰ ਹੋਣ ਦਾ ਖਤਰਾ ਬਣ ਜਾਂਦਾ ਹੈ। ਇਹ ਕਈ ਗੰਭੀਰ ਬਿਮਾਰੀਆਂ ਦਾ ਕਾਰਨ ਬਣਦਾ ਹੈ। ਦੂਸ਼ਿਤ ਮਿੱਟੀ ਵੀ ਆਪਣੀ ਉਤਪਾਦਕਤਾ ਗੁਆ ਦਿੰਦੀ ਹੈ।
3. **ਜਲ ਪ੍ਰਦੂਸ਼ਣ (Water Pollution):** ਈ.ਵੇਸਟ ਪਾਣੀ ਦੇ ਪ੍ਰਦੂਸ਼ਣ ਦਾ ਕਾਰਨ ਵੀ ਬਣਦਾ ਹੈ। ਈ-ਵੇਸਟ ਵਿੱਚ ਆਮ ਤੌਰ ਤੇ ਪਾਰਾ, ਲਿਥੀਅਮ, ਲੈੱਡ ਅਤੇ ਬੇਰੀਅਮ ਆਦਿ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਇਸ ਲਈ ਜਦੋਂ ਈ-ਵੇਸਟ ਦਾ ਨਿਪਟਾਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਉਹ ਆਖਰਕਾਰ ਤਾਲਾਬਾਂ, ਨਦੀਆਂ, ਨਹਿਰਾਂ ਅਤੇ ਝੀਲਾਂ ਵਿੱਚ ਆਪਣਾ ਰਸਤਾ ਬਣਾਉਂਦੇ ਹਨ। ਪਾਣੀ ਦੇ ਸਰੋਤਾਂ ਵਿੱਚ ਪ੍ਰਦੂਸ਼ਕਾਂ ਦਾ ਇਹ ਵਹਾਅ ਪਾਣੀ ਨੂੰ ਦੂਸ਼ਿਤ ਕਰਦਾ ਹੈ ਜੋ ਪਾਣੀ ਵਿੱਚ ਜਾਨਵਰਾਂ, ਪੌਦਿਆਂ, ਮਨੁੱਖਾਂ ਆਦਿ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਂਦਾ ਹੈ।
4. **ਮਨੁੱਖ ਤੇ ਮਾੜੇ ਪ੍ਰਭਾਵ (Bad effects on Humans):** ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਜਾਣਦੇ ਹਾਂ ਕਿ ਈ-ਵੇਸਟ ਵਿੱਚ ਪਾਰਾ, ਲਿਥੀਅਮ, ਲੈੱਡ ਅਤੇ ਬੇਰੀਅਮ ਆਦਿ ਜ਼ਹਿਰੀਲੇ ਪਦਾਰਥ ਹੁੰਦੇ ਹਨ ਜੋ ਮਨੁੱਖਾਂ ਦੀ ਸਿਹਤ ਤੇ ਨੁਕਸਾਨਦੇਹ ਪ੍ਰਭਾਵ ਪਾਉਂਦੇ ਹਨ। ਇਸ ਉਦਯੋਗ ਦੇ ਕਾਮਿਆਂ ਨੂੰ ਥਾਇਰਾਇਡ, ਹਾਰਮੋਨਲ ਅਸੰਤੁਲਨ ਵਰਗੀਆਂ ਕਈ ਬਿਮਾਰੀਆਂ ਹੋਣ ਦੀ ਸੰਭਾਵਨਾ ਵੱਧ ਹੁੰਦੀ ਹੈ। ਸਾਹ ਦੀ ਕਮੀ ਜਾਂ ਦਮਾ, ਹਾਈਪਰਟੈਨਸ਼ਨ, ਚਮੜੀ ਦੇ ਵਿਕਾਰ ਆਦਿ।

7.6.2 ਈ-ਵੇਸਟ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਦੇ ਤਰੀਕੇ (Methods to manage E-waste):

ਅਸੀਂ ਈ-ਵੇਸਟ ਨੂੰ ਕਈ ਤਰੀਕਿਆਂ ਨਾਲ ਸੰਭਾਲ ਸਕਦੇ ਹਾਂ ਤਾਂ ਜੋ ਅਣ-ਪ੍ਰਬੰਧਿਤ ਈ-ਵੇਸਟ ਦੇ ਬੁਰੇ ਪ੍ਰਭਾਵਾਂ ਨੂੰ ਘੱਟ ਕੀਤਾ ਜਾਵੇ। ਇਹਨਾਂ ਵਿੱਚੋਂ ਕੁਝ ਪ੍ਰਭਾਵੀ ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਕਿਸਮਾਂ ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਹਨ:

1. **ਲੈਂਡਫਿਲਜ਼ (Landfills):** ਇਹ ਈ-ਵੇਸਟ ਦੇ ਨਿਪਟਾਰੇ ਦਾ ਸਭ ਤੋਂ ਵੱਧ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਤਰੀਕਾ ਹੈ। ਇਸ ਵਿਧੀ ਵਿੱਚ ਈ-ਵੇਸਟ ਨੂੰ ਦੱਬਣ ਲਈ ਮਿੱਟੀ ਵਿੱਚ ਵੱਡੀਆਂ ਖਾਈਆਂ ਬਣਾਈਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਪਰ ਈ-ਵੇਸਟ ਦਾ ਨਿਪਟਾਰਾ ਕਰਨ ਦਾ ਇਹ ਵਧੀਆ ਤਰੀਕਾ ਨਹੀਂ ਹੈ

ਚਿੱਤਰ 7.5 ਲੈਂਡਫਿਲਜ਼ (Landfills)

ਕਿਉਂਕਿ ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਨਾਲ ਧਰਤੀ ਵਿਚ ਲੈੱਡ, ਪਾਰਾ ਆਦਿ ਵਰਗੇ ਜ਼ਹਿਰੀਲੇ ਪਦਾਰਥ ਨਿਕਲਦੇ ਹਨ ਅਤੇ ਧਰਤੀ ਹੇਠਲੇ ਪਾਣੀ ਅਤੇ ਮਿੱਟੀ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਂਦੇ ਹਨ।

2. **ਸਾੜਨਾ (Incineration):** ਇਹ ਈ-ਵੇਸਟ ਦੇ ਨਿਪਟਾਰੇ ਲਈ ਆਮ ਤੌਰ ਤੇ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਤਰੀਕਾ ਹੈ। ਇਸ ਵਿਧੀ ਵਿੱਚ ਈ-ਵੇਸਟ ਨੂੰ ਉੱਚ ਤਾਪਮਾਨ ਤੇ ਵਿਸ਼ੇਸ਼ ਤੌਰ ਤੇ ਤਿਆਰ ਕੀਤੀਆਂ ਗਈ ਭੱਠੀਆਂ ਵਿੱਚ ਸਾੜਿਆ ਜਾਂਦਾ ਹੈ। ਜਿਸ ਕਾਰਨ ਈ-ਵੇਸਟ ਦੀ ਮਾਤਰਾ ਘੱਟ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸ ਵਿਧੀ ਨਾਲ ਪੈਦਾ ਹੋਈ ਊਰਜਾ ਦੀ ਵੀ ਅਲੱਗ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਪਰ ਇਹ ਇੱਕ ਚੰਗਾ ਤਰੀਕਾ ਵੀ ਨਹੀਂ ਹੈ ਕਿਉਂਕਿ ਜਦੋਂ ਈ-ਵੇਸਟ ਸੜਦਾ ਹੈ ਤਾਂ ਇਹ ਹਾਨੀਕਾਰਕ ਗੈਸਾਂ ਛੱਡਦਾ ਹੈ ਜੋ ਸਾਡੇ ਵਾਤਾਵਰਣ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਂਦਾ ਹੈ।

ਚਿੱਤਰ 7.6 ਸਾੜਨਾ (Incineration)

3. **ਐਸਿਡ ਬਾਥ (Acid Bath):** ਇਸ ਵਿਧੀ ਵਿੱਚ ਈ-ਵੇਸਟ ਨੂੰ ਸ਼ਕਤੀਸ਼ਾਲੀ ਸਲਫਿਊਰਿਕ, ਹਾਈਡ੍ਰੋਕਲੋਰਿਕ, ਨਾਈਟ੍ਰਿਕ ਐਸਿਡ ਘੋਲ ਵਿੱਚ ਭਿਗੋਇਆ ਜਾਂਦਾ ਹੈ ਜੋ ਈ-ਵੇਸਟ ਵਿੱਚੋਂ ਧਾਤ ਨੂੰ ਹਟਾ ਦਿੰਦਾ ਹੈ। ਇਸ ਤਰੀਕੇ ਨਾਲ ਪ੍ਰਾਪਤ ਕੀਤੀ ਗਈ ਧਾਤ ਨੂੰ ਹੋਰ ਉਤਪਾਦ ਬਣਾਉਣ ਲਈ ਦੁਬਾਰਾ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਵਿਧੀ ਵਿੱਚ ਵੀ ਕਮੀਆਂ ਹਨ, ਜਿਵੇਂ ਕਿ ਤੇਜ਼ਾਬ ਦੇ ਘੋਲ ਨੂੰ ਕਈ ਵਾਰ ਪਾਣੀ ਦੇ ਸਰੋਤਾਂ ਵਿੱਚ ਸੁੱਟ ਦਿੱਤਾ ਜਾਂਦਾ ਹੈ ਜੋ ਜੀਵਿਤ ਚੀਜ਼ਾਂ ਲਈ ਵਧੀਆ ਨਹੀਂ ਹੁੰਦੇ।

ਚਿੱਤਰ 7.7 ਐਸਿਡ ਬਾਥ (Acid Bath)

4. **ਮਕੈਨੀਕਲ ਰੀਸਾਈਕਲਿੰਗ (Mechanial Recycling):** ਇਹ ਸਭ ਤੋਂ ਕੁਸ਼ਲ ਤਰੀਕਾ ਹੈ ਅਤੇ ਵਾਤਾਵਰਣ ਨੇ ਅਨੁਕੂਲ ਵੀ ਹੈ। ਇਸ ਵਿਧੀ ਵਿੱਚ ਅਸੀਂ ਵਰਤੇ ਹੋਏ ਸਰਕਟ ਬੋਰਡਾਂ, ਆਈ.ਸੀ., ਮਦਰਬੋਰਡ ਆਦਿ ਨੂੰ ਈ-ਵੇਸਟ ਤੋਂ ਮੁੜ ਪ੍ਰਾਪਤ ਕਰਨ ਅਤੇ ਉਹਨਾਂ ਨੂੰ ਰੀਸਾਈਕਲ ਕਰਨ ਲਈ ਸੁੱਕੇ ਭੌਤਿਕ ਵਿਭਾਜਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਇਸ ਵਿਧੀ ਵਿੱਚ ਤਾਂਬਾ, ਸੀਸਾ ਆਦਿ ਕੀਮਤੀ ਧਾਤਾਂ ਨੂੰ ਪੀਸੀਬੀ ਰੀਸਾਈਕਲਿੰਗ ਮਸ਼ੀਨ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵਾਤਾਵਰਣ ਨੂੰ ਨੁਕਸਾਨ ਪਹੁੰਚਾਏ ਬਿਨਾਂ ਈ-ਵੇਸਟ ਤੋਂ ਵੱਖ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

ਚਿੱਤਰ 7.8 ਮਕੈਨੀਕਲ ਰੀਸਾਈਕਲਿੰਗ (Mechanical Recycling)

Test Yourself

ਸਵਾਲ: ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਦਾ ਕਿਹੜਾ ਤਰੀਕਾ ਸਭ ਤੋਂ ਵੱਧ ਕੁਸ਼ਲ ਅਤੇ ਵਾਤਾਵਰਣ ਅਨੁਕੂਲ ਹੈ?

ੳ. ਲੈਂਡਫਿਲਜ਼ **ਅ. ਸਾੜਨਾ**

ੲ. ਐਸਿਡ ਬਾਥ **ਸ. ਮਕੈਨੀਕਲ ਰੀਸਾਈਕਲਿੰਗ**

ਇੱਕ ਕੁਸ਼ਲ ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ ਦਾ ਇੱਕ ਹੋਰ ਪਹਿਲੂ ਵੀ ਹੈ। ਡਾਟਾ ਪ੍ਰੋਸੈਸਿੰਗ ਵਿੱਚ ਸਾਡੇ ਹਰੇਕ ਡਿਜੀਟਲ ਕੰਪੋਨੈਂਟ ਦਾ ਆਪਣਾ ਮਹੱਤਵ ਹੈ। ਜਦੋਂ ਡਾਟਾ ਨੂੰ ਮਿਟਾਏ ਬਿਨਾਂ ਡਾਟਾ ਵਾਲੇ ਕਿਸੇ ਵੀ ਹਿੱਸੇ ਨੂੰ ਸੁੱਟ ਦਿੱਤਾ ਜਾਂਦਾ ਹੈ, ਤਾਂ ਇਹ ਡਾਟੇ ਦੀ ਦੁਰਵਰਤੋਂ ਦਾ ਕਾਰਨ ਬਣ ਸਕਦਾ ਹੈ ਅਤੇ ਸਾਈਬਰ ਖ਼ਤਰਾ ਪੈਦਾ ਕਰ ਸਕਦਾ ਹੈ। ਸਾਨੂੰ ਈ-ਵੇਸਟ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਦੇ ਸਮੇਂ ਵਿਚਾਰ ਅਧੀਨ ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਸਿਧਾਂਤਾਂ 'ਤੇ ਵਿਚਾਰ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ।

7.7 ਸਾਈਬਰ ਕਾਨੂੰਨ (CYBER LAWS)

ਜਦੋਂ ਇੰਟਰਨੈਟ ਵਿਕਸਤ ਕੀਤਾ ਗਿਆ ਸੀ ਤਾਂ ਇਹ ਨਹੀਂ ਦੇਖਿਆ ਗਿਆ ਸੀ ਕਿ ਇੰਟਰਨੈਟ ਆਪਣੇ ਆਪ ਨੂੰ ਇੱਕ ਸਰਵ ਵਿਆਪਕ ਕ੍ਰਾਂਤੀ ਵਿੱਚ ਬਦਲ ਸਕਦਾ ਹੈ। ਇਸਦੀ ਅਪਰਾਧਿਤ ਗਤੀਵਿਧੀਆਂ ਲਈ ਦੁਰਵਰਤੋ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ ਅਤੇ ਜਿਸ ਲਈ ਨਿਯਮ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਅੱਜ ਸਾਈਬਰ ਸਪੇਸ ਵਿੱਚ ਬਹੁਤ ਸਾਰੀਆਂ ਪਰੇਸ਼ਾਨ ਕਰਨ ਵਾਲੀਆਂ ਚੀਜ਼ਾਂ ਹੋ ਰਹੀਆਂ ਹਨ। ਇੰਟਰਨੈਟ ਦੀ ਗੁਮਨਾਮ ਪ੍ਰਕਿਰਤੀ ਦੇ ਕਾਰਨ ਸਾਈਬਰ ਸਪੇਸ ਵਿੱਚ ਕਈ ਤਰ੍ਹਾਂ ਦੀਆਂ ਅਪਰਾਧਿਕ ਗਤੀਵਿਧੀਆਂ ਸ਼ਾਮਲ ਹੋਣਾ ਸੰਭਵ ਹੈ। ਇਸ ਲਈ ਭਾਰਤ ਵਿੱਚ ਸਾਈਬਰ ਕਾਨੂੰਨ ਸਾਡੇ ਸਾਰਿਆਂ ਲਈ ਇੱਕ ਲੋੜ ਬਣ ਗਏ ਹਨ। ਆਓ ਭਾਰਤ ਵਿੱਚ ਸਾਈਬਰ ਕਾਨੂੰਨਾਂ ਤੇ ਨੇੜਿਓਂ ਨਜ਼ਰ ਮਾਰੀਏ।

ਚਿੱਤਰ 7.9 ਸਾਈਬਰ ਕਾਨੂੰਨ (CYBER LAWS)

7.7.1 ਸੂਚਨਾ ਤਕਨਾਲੋਜੀ ਐਕਟ, 2000 (ਭਾਰਤ) (Information Technology Act, 2000-India):

ਸੂਚਨਾ ਤਕਨਾਲੋਜੀ ਐਕਟ, 2000 ਜਿਸਨੂੰ ਆਈ.ਟੀ. ਐਕਟ ਵਜੋਂ ਵੀ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ, 17 ਅਕਤੂਬਰ 2000 ਨੂੰ ਭਾਰਤੀ ਸੰਸਦ ਦੁਆਰਾ ਪ੍ਰਸਤਾਵਿਤ ਇੱਕ ਐਕਟ ਬਣਾਇਆ ਗਿਆ ਹੈ। ਇਹ ਸੂਚਨਾ ਤਕਨਾਲੋਜੀ ਐਕਟ ਸੰਯੁਕਤ ਰਾਸ਼ਟਰ ਦੇ ਇਲੈਕਟ੍ਰਾਨਿਕ ਕਾਮਰਸ 1996 (UNCITRAL ਮਾਡਲ) ਦੇ ਮਾਡਲ ਕਾਨੂੰਨ ਤੇ ਆਧਾਰਿਤ ਹੈ। ਜਿਸ ਦਾ ਸੁਝਾਅ ਸੰਯੁਕਤ ਰਾਸ਼ਟਰ ਦੀ ਜਨਰਲ ਅਸੈਂਬਲੀ ਵੱਲੋਂ 30 ਜਨਵਰੀ, 1997 ਨੂੰ ਇੱਕ ਮਤੇ ਰਾਹੀਂ ਦਿੱਤਾ ਗਿਆ ਸੀ। ਇਹ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਅਤੇ ਈ-ਕਾਮਰਸ ਨਾਲ ਨਜਿੱਠਣ ਲਈ ਭਾਰਤ ਵਿੱਚ ਸਭ ਤੋਂ ਮਹੱਵਪੂਰਨ ਕਾਨੂੰਨ ਹੈ।

ਇਸ ਐਕਟ ਦਾ ਮੁੱਖ ਉਦੇਸ਼ ਕਾਨੂੰਨੀ ਅਤੇ ਭਰੋਸੇਮੰਦ ਇਲੈਕਟ੍ਰਾਨਿਕ, ਡਿਜੀਟਲ ਅਤੇ ਆਨਲਾਈਨ ਲੈਣ-ਦੇਣ ਅਤੇ ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਨੂੰ ਘੱਟ ਕਰਨਾ ਜਾਂ ਘਟਾਉਣਾ ਹੈ। ਆਈ.ਟੀ. ਐਕਟ ਦੇ 13 ਅਧਿਆਏ ਅਤੇ 90 ਸੈਕਸ਼ਨ ਹਨ। ਆਖਰੀ ਚਾਰ ਸੈਕਸ਼ਨ, ਸੈਕਸ਼ਨ 91 - ਸੈਕਸ਼ਨ 94, ਭਾਰਤੀ ਦੰਡ ਵਿਧਾਨ 1860 ਦੇ ਸੰਸ਼ੋਧਨਾਂ ਨਾਲ ਸੰਬੰਧਿਤ ਹੈ। ਆਈ.ਟੀ. ਐਕਟ 2000 ਪੁਰਾਣੇ ਕਾਨੂੰਨਾਂ ਨੂੰ ਬਦਲਣ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰਦਾ ਹੈ ਅਤੇ ਸਾਈਬਰ ਅਪਰਾਧਾਂ ਨਾਲ ਨਜਿੱਠਣ ਦੇ ਤਰੀਕੇ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਸਾਨੂੰ ਅਜਿਹੇ ਕਾਨੂੰਨਾਂ ਦੀ ਲੋੜ ਹੈ ਤਾਂ ਜੋ ਲੋਕ ਦੁਰਵਰਤੋਂ ਦੇ ਡਰ ਤੋਂ ਬਿਨਾਂ ਕ੍ਰੈਡਿਟ ਕਾਰਡਾਂ ਰਾਹੀਂ ਨੈੱਟ ਤੇ ਖਰੀਦਦਾਰੀ ਲੈਣ-ਦੇਣ ਕਰ ਸਕਣ। ਐਕਟ ਬਹੁਤ ਲੋੜੀਂਦਾ ਕਾਨੂੰਨੀ ਢਾਂਚਾ ਪੇਸ਼ ਕਰਦਾ ਹੈ ਤਾਂ ਜੋ ਜਾਣਕਾਰੀ ਨੂੰ ਕਾਨੂੰਨੀ ਪ੍ਰਭਾਵ, ਵੈਧਤਾ ਜਾਂ ਲਾਗੂ ਕਰਨ ਤੋ ਇਨਕਾਰ ਨਾ ਕੀਤਾ ਜਾਵੇ। ਸਿਰਫ ਇਸ ਆਧਾਰ ਤੇ ਕਿ ਇਹ ਇਲੈਕਟ੍ਰਾਨਿਕ ਰਿਕਾਰਡਾਂ ਦੇ ਰੂਪ ਵਿੱਚ ਹੈ।

TEST YOURSELF

ਸਹੀ ਉੱਤਰ ਲਿਖੋ:

ਭਾਰਤ ਵਿੱਚ ਸੂਚਨਾ ਤਕਨਾਲੋਜੀ ਐਕਟ ਕਿਸ ਮਿਤੀ ਨੂੰ ਪ੍ਰਸਤਾਵਿਤ ਕੀਤਾ ਗਿਆ ਸੀ:

7.7.2 ਆਈ.ਟੀ. ਐਕਟ ਦੀਆਂ ਸਮਾਂ-ਸੂਚੀਆਂ:

ਆਈ.ਟੀ. ਐਕਟ, 2000 ਦੇ ਦੋ ਕਾਰਜਕਰਮ ਹਨ:

- **ਪਹਿਲੀ ਅਨੁਸੂਚੀ:** ਉਹਨਾਂ ਦਸਤਾਵੇਜ਼ਾਂ ਨਾਲ ਨਜਿੱਠਦਾ ਹੈ ਜਿਨ੍ਹਾਂ ਤੇ ਐਕਟ ਲਾਗੂ ਨਹੀਂ ਹੋਵੇਗਾ।
- **ਦੂਜੀ ਅਨੁਸੂਚੀ:** ਇਲੈਕਟ੍ਰਾਨਿਕ ਦਸਤਖਤ ਜਾਂ ਇਲੈਕਟ੍ਰਾਨਿਕ ਪ੍ਰਮਾਣਿਕਤਾ ਵਿਧੀ ਨਾਲ ਨਜਿੱਠਦਾ ਹੈ।

7.7.3 ਆਈ.ਟੀ. ਐਕਟ 2000 ਵਿੱਚ ਧਾਰਾਵਾਂ ਅਤੇ ਸਜ਼ਾਵਾਂ:

ਆਈ.ਟੀ. ਐਕਟ 2000 ਦੀਆਂ ਇਹਨਾਂ ਧਾਰਾਵਾਂ ਅਧੀਨ ਆਉਂਦੇ ਕੁਝ ਮਹੱਤਵਪੂਰਨ ਧਾਰਾਵਾਂ ਅਤੇ ਸਜ਼ਾਵਾਂ ਨੂੰ ਹੇਠਾਂ ਦਿੱਤਾ ਗਿਆ ਹੈ:

1. **ਧਾਰਾ 43:** ਆਈ.ਟੀ. ਐਕਟ, 2000 ਦਾ ਇਹ ਸੈਕਸ਼ਨ ਡਾਟਾ ਨੂੰ ਗਲਤ ਢੰਗ ਨਾਲ ਚਲਾਉਣ ਜਾਂ ਚੋਰੀ ਕਰਨ ਦੀਆਂ ਚਿੰਤਾਵਾਂ ਨਾਲ ਸਬੰਧਤ ਹੈ। ਇਹ ਵਿਅਕਤੀਆਂ ਦੇ ਨਾਲ-ਨਾਲ ਕਾਰਪੋਰੇਟ ਸੰਸਥਾਵਾਂ ਤੇ ਵੀ ਲਾਗੂ ਹੋ ਸਕਦਾ ਹੈ ਅਤੇ ਭੁਗਤਾਨ ਜਾਂ ਮੁਆਵਜ਼ੇ ਦੁਆਰਾ ਦੇਣਦਾਰੀ ਨੂੰ ਫਿਕਸ ਕਰ ਸਕਦਾ ਹੈ।
2. **ਧਾਰਾ 66:** ਇਹ ਧਾਰਾ ਅਸਲ ਵਿੱਚ ਹੈਕਿੰਗ, ਖਤਰਨਾਕ ਗਤੀਵਿਧੀਆਂ, ਡਿਜੀਟਲ ਧੋਖਾਧੜੀ ਆਦਿ ਨਾਲ ਸਬੰਧਤ ਹੈ ਅਤੇ ਅਪਰਾਧੀਆਂ ਨੂੰ 3 ਸਾਲ ਦੀ ਕੈਦ 5,00,000 ਰੁਪਏ ਤੱਕ ਦੇ ਜੁਰਮਾਨੇ ਜਾਂ ਦੋਵੇਂ ਹੋ ਸਕਦੇ ਹਨ। ਸਾਈਬਰ ਅੱਤਵਾਦ ਦੇ ਮਾਮਲੇ ਵਿੱਚ ਉਮਰ ਕੈਦ ਦੀ ਸਜ਼ਾ ਵੀ ਹੋ ਸਕਦੀ ਹੈ।
3. **ਧਾਰਾ 67:** ਇਹ ਧਾਰਾ ਦੱਸਦੀ ਹੈ ਕਿ ਅਸ਼ਲੀਲ ਜਾਣਕਾਰੀ ਜਾਂ ਨਿੱਜੀ ਸਮੱਗਰੀ ਨੂੰ ਪ੍ਰਕਾਸ਼ਤ ਕਰਨਾ ਜਾਂ ਅਸ਼ਲੀਲ ਸਮੱਗਰੀ ਨੂੰ ਜਨਤਕ ਤੌਰ ਤੇ ਪ੍ਰਸਾਰਿਤ ਕਰਨਾ 5 ਸਾਲ ਤੱਕ ਦੀ ਕੈਦ ਜਾਂ 10,00,000 ਰੁਪਏ ਦੇ ਜੁਰਮਾਨੇ ਜਾਂ ਦੋਵੇਂ ਹੋ ਸਕਦੇ ਹਨ।

ਯਾਦ ਰੱਖਣ ਵਾਲੇ ਨੁਕਤੇ

1. ਕਿਸੇ ਵਿਅਕਤੀ ਜਾਂ ਲੋਕਾਂ ਦੇ ਸਮੂਹਾਂ ਦੇ ਵਿਰੁੱਧ ਇਲੈਕਟਰਾਨਿਕ ਸਾਧਨਾਂ ਦੁਆਰਾ ਨੁਕਸਾਨ ਪਹੁੰਚਾਉਣ ਜਾਂ ਸਰੀਰਕ ਜਾਂ ਮਾਨਸਿਕ ਸਦਮੇ ਦਾ ਕਾਰਨ ਬਣਨ ਵਾਲੇ ਕਿਸੇ ਵੀ ਅਪਰਾਧ ਨੂੰ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਵਜੋਂ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।
2. ਇਵਸਡ੍ਰੌਪਿੰਗ ਕੁਝ ਮਹੱਤਵਪੂਰਨ ਜਾਂ ਗੁਪਤ ਜਾਣਕਾਰੀ ਇਕੱਠੀ ਕਰਨ ਲਈ ਉਹਨਾਂ ਦੀ ਸਹਿਮਤੀ ਤੋਂ ਬਿਨਾਂ ਦੂਜਿਆਂ ਦੀ ਨਿੱਜੀ ਗੱਲਬਾਤ ਜਾਂ ਸੰਚਾਰ ਨੂੰ ਗੁਪਤ ਜਾਂ ਚੋਰੀ-ਛਿਪੇ ਸੁਣਨ ਦਾ ਕੰਮ ਹੈ।
3. ਫਿਸ਼ਿੰਗ ਇੱਕ ਅਜਿਹਾ ਹਮਲਾ ਹੈ ਜੋ ਤੁਹਾਨੂੰ ਕਿਸੇ ਵੀ ਲਾਲਚ ਦੇ ਰੂਪ ਵਿੱਚ ਨਿੱਜੀ ਜਾਣਕਾਰੀ ਜਿਵੇਂ ਕਿ ਕ੍ਰੈਡਿਟ ਕਾਰਡ ਨੰਬਰ, ਬੈਂਕ ਜਾਣਕਾਰੀ ਜਾਂ ਵੈੱਬਸਾਈਟਾਂ ਤੇ ਪਾਸਵਰਡ ਨੂੰ ਪ੍ਰਗਟ ਕਰਨ ਲਈ ਤੁਹਾਡੇ ਪੈਸੇ ਜਾਂ ਤੁਹਾਡੀ ਪਛਾਣ ਨੂੰ ਚੋਰੀ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰਦਾ ਹੈ।
4. ਰੈਨਸਮਵੇਅਰ ਇੱਕ ਕਿਸਮ ਦਾ ਮਾਲਵੇਅਰ (ਨੁਕਸਾਨਦਾਇਕ ਸਾਫਟਵੇਅਰ) ਹੈ ਜੋ ਪੀੜਤ ਦੇ ਡਾਟਾ ਜਾਂ ਡਿਵਾਈਸ ਨੂੰ ਲਾਕ ਕਰਦਾ ਹੈ ਅਤੇ ਇਸ ਨੂੰ ਲਾਕ ਰੱਖਣ ਦੀ ਧਮਕੀ ਦਿੰਦਾ ਹੈ ਜਦੋਂ ਤੱਕ ਪੀੜਤ ਹਮਲਾਵਰ ਨੂੰ ਫਿਰੌਤੀ ਅਦਾ ਨਹੀਂ ਕਰਦਾ।
5. ਸਾਈਬਰ ਸਟਾਕਿੰਗ, ਸਾਈਬਰ ਅਪਰਾਧ ਦੀ ਇੱਕ ਕਿਸਮ ਹੈ ਜੋ ਕਿਸੇ ਵਿਅਕਤੀ ਨੂੰ ਪਰੇਸ਼ਾਨ ਕਰਨ ਜਾਂ ਪਿੱਛਾ ਕਰਨ ਲਈ ਇੰਟਰਨੈਟ ਅਤੇ ਤਕਨਾਲੋਜੀ ਦੀ ਵਰਤੋਂ ਕਰਦੀ ਹੈ। ਇਸ ਨੂੰ ਸਾਈਬਰ ਧੱਕੇਸ਼ਾਹੀ ਅਤੇ ਵਿਅਕਤੀਗਤ ਤੌਰ ਤੇ ਪਿੱਛਾ ਕਰਨ ਦਾ ਇੱਕ ਵਿਸਥਾਰ ਮੰਨਿਆ ਜਾ ਸਕਦਾ ਹੈ।
6. ਸਾਹਿਤਕ ਚੋਰੀ ਕਿਸੇ ਹੋਰ ਵਿਅਕਤੀ ਦੇ ਕੰਮ ਜਾਂ ਵਿਚਾਰ ਨੂੰ ਤੁਹਾਡੇ ਆਪਣੇ ਵਜੋਂ ਪੇਸ਼ ਕਰਨ ਦਾ ਕੰਮ ਹੈ। ਇਹ ਅਪਰਾਧ ਉਦੋਂ ਵਾਪਰਦਾ ਹੈ ਜਦੋਂ ਕੋਈ ਵਿਅਕਤੀ ਕਾਪੀਰਾਈਟ ਦੀ ਉਲੰਘਣਾ ਕਰਦਾ ਹੈ ਅਤੇ ਸੰਗੀਤ, ਫ਼ਿਲਮਾਂ, ਗੇਮਾਂ ਅਤੇ ਸਾਫਟਵੇਅਰ ਡਾਊਨਲੋਡ ਕਰਦਾ ਹੈ।
7. ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਜਾਂ ਇੰਟਰਨੈਟ ਸੁਰੱਖਿਆ ਆਪਣੇ ਆਪ ਨੂੰ ਕੰਪਿਊਟਰ ਅਪਰਾਧ ਤੋਂ ਬਚਾ ਰਹੀ ਹੈ ਅਤੇ ਉਪਭੋਗਤਾ ਨੂੰ ਨਿੱਜੀ ਅਤੇ ਗੁਪਤ ਜਾਣਕਾਰੀ ਤੱਕ ਸੁਰੱਖਿਆ ਉਲੰਘਣਾ ਦੇ ਜੋਖਮ ਨੂੰ ਘਟਾਉਂਦੀ ਹੈ।
8. ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਇੰਟਰਨੈੱਟ ਤੇ ਜ਼ਿੰਮੇਵਾਰ ਵਿਵਹਾਰ ਦੇ ਕੋਡ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਇਹ ਸੁਨਿਸ਼ਚਿਤ ਕੀਤਾ ਜਾਂਦਾ

ਹੈ ਕਿ ਉਪਭੋਗਤਾ ਆਪਣੇ ਆਪ ਨੂੰ ਆਨਲਾਈਨ ਕਰਨ ਲਈ ਆਪਣੀਆਂ ਜ਼ਿੰਮੇਵਾਰੀਆਂ ਨੂੰ ਸਮਝਦੇ ਹਨ।

9. ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਈਡ ਦਾ ਅਰਥ ਹੈ ਗੁਪਤਤਾ, ਇਕਸਾਰਤਾ ਅਤੇ ਉਪਲਬਧਤਾ।

10. ਈ-ਵੇਸਟ ਜਾਂ ਇਲੈਕਟ੍ਰਾਨਿਕ ਵੇਸਟ ਦਾ ਮਤਲਬ ਹੈ ਰੱਦ ਕੀਤੇ ਇਲੈਕਟ੍ਰੀਕਲ ਜਾਂ ਇਲੈਕਟ੍ਰਾਨਿਕ ਯੰਤਰ ਜਾਂ ਕੰਪੋਨੈਂਟ ਨੂੰ ਸਹੀ ਢੰਗ ਨਾਲ ਨਸ਼ਟ ਕਰਨ ਦੀ ਪਰਕਿਰਿਆ ਹੈ।

11. ਆਈ.ਟੀ. ਐਕਟ, 2000 ਦੀਆਂ ਦੋ ਧਾਰਾਵਾਂ ਹਨ: ਪਹਿਲੀ ਅਨੁਸੂਚੀ ਉਹਨਾਂ ਦਸਤਾਵੇਜ਼ਾਂ ਨਾਲ ਸੰਬੰਧਿਤ ਹੈ ਜਿਨ੍ਹਾਂ ਤੇ ਐਕਟ ਲਾਗੂ ਨਹੀਂ ਹੋਵੇਗਾ ਅਤੇ ਦੂਜੀ ਅਨੁਸੂਚੀ ਇਲੈਕਟ੍ਰਾਨਿਕ ਦਸਤਖਤ ਜਾਂ ਇਲੈਕਟ੍ਰਾਨਿਕ ਪ੍ਰਮਾਣੀਕਰਨ ਵਿਧੀ ਨਾਲ ਸੰਬੰਧਿਤ ਹੈ।

ਅਭਿਆਸ

ੳ. ਬਹੁਵਿਕਲਪੀ ਪ੍ਰਸ਼ਨ

1. ਸੂਚਨਾ ਤਕਨਾਲੋਜੀ ਐਕਟ, 2000 (ਭਾਰਤ) ਭਾਰਤ ਵਿੱਚ ਕਦੋਂ ਹੋਂਦ ਵਿੱਚ ਆਇਆ ?

ੳ) 17 ਅਕਤੂਬਰ 2000 ਅ) 27 ਨਵੰਬਰ 2003

ੲ) 1 ਜਨਵਰੀ 2006 ਸ) 17 ਅਕਤੂਬਰ 2009

2. ਈ-ਵੇਸਟ ਡਾਟਾ ਚੋਰੀ ਦਾ ਨਤੀਜਾ ਕਿਵੇਂ ਨਿਕਲਦਾ ਹੈ ?

ੳ) ਈਮੇਲ ਫਾਰਵਰਡਿੰਗ ਦੁਆਰਾ।

ਅ) ਬਿਨਾਂ ਡਾਟਾ ਕਲੀਅਰ ਕੀਤੇ ਪੁਰਾਣੇ ਇਲੈਕਟ੍ਰਾਨਿਕ ਡਿਵਾਈਸ ਨੂੰ ਵੇਸਟ ਵਿੱਚ ਬਦਲ ਕੇ।

ੲ) ਡਾਟਾ ਸਾਂਝਾ ਕਰਕੇ।

ਸ) ਵਾਈ-ਫਾਈ ਨੈੱਟਵਰਕ ਦੀ ਵਰਤੋਂ ਕਰਕੇ।

3. ਸਾਹਿਤਕ ਚੋਰੀ (Plagiarism) ਦਾ ਮਤਲਬ ਕੀ ਹੈ ?

ੳ) ਕਿਸੇ ਹੋਰ ਵਿਅਕਤੀ ਦੇ ਕੰਮ ਜਾਂ ਵਿਚਾਰ ਨੂੰ ਤੁਹਾਡੇ ਆਪਣੇ ਵਜੋਂ ਪੇਸ਼ ਕਰਨ ਦਾ ਐਕਟ।

ਅ) ਇੱਕ ਬਿਮਾਰੀ ਜੋ ਮਨੁੱਖਾਂ ਅਤੇ ਹੋਰ ਜੀਵਾਂ ਨੂੰ ਪ੍ਰਭਾਵਿਤ ਕਰਦੀ ਹੈ।

ੲ) ਇੱਕ ਛੂਤ ਵਾਲੀ ਬੈਕਟੀਰੀਆ ਦੀ ਬਿਮਾਰੀ ਜਿਸਦੀ ਵਿਸ਼ੇਸ਼ਤਾ ਬੁਖਾਰ ਨਾਲ ਹੁੰਦੀ ਹੈ।

ਸ) ਉਪਰੋਕਤ ਵਿੱਚੋਂ ਕੋਈ ਨਹੀਂ।

4. ਇਹਨਾਂ ਵਿਚੋਂ ਕਿਹੜਾ ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਇਡ ਦਾ ਹਿਸਾ ਨਹੀਂ ਹੈ।

ੳ) ਗੁਪਤਤਾ ਅ) ਇਕਸਾਰਤ ੲ) ਉਪਲਬਧਤਾ ਸ) ਫੀਸ਼ਿੰਗ

5. ਕਿਸੇ ਵੀ ਆਨਲਾਈਨ ਧੋਖਾਧੜੀ ਦੇ ਮਾਮਲੇ ਵਿੱਚ ਪੰਜਾਬ ਪੁਲਿਸ ਦੇ ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਡਿਵੀਜ਼ਨ ਕੋਲ ਕਿਸ ਟੋਲ ਫ੍ਰੀ ਨੰਬਰ ਤੇ ਸ਼ਿਕਾਇਤ ਦਰਜ ਕਰਵਾਈ ਜਾ ਸਕਦੀ ਹੈ ?

ੳ) 1911 ਅ) 1930 ੲ) 1947 ਸ) 1912

6. ਭਾਰਤ ਵਿੱਚ ਕਿਹੜਾ ਐਕਟ ਸਾਈਬਰ ਅਪਰਾਧ ਤੇ ਕੇਂਦਰਿਤ ਹੈ ?

ੳ) ਬੈਂਕਿੰਗ ਰੇਗੂਲੇਸ਼ਨ ਐਕਟ 1949 ਅ) ਆਈ.ਟੀ. ਐਕਟ 2000

ੲ) ਭਾਰਤੀ ਦੰਡਾਵਲੀ 1860 ਸ) CrPC 1973

ਅ. ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ

1. ਡਿਜੀਟਲ ਗੈਜੇਟ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਕਿਸੇ ਵਿਅਕਤੀ ਜਾਂ ਲੋਕਾਂ ਦੇ ਸਮੂਹਾਂ ਦੇ ਵਿਰੁੱਧ ਕੀਤਾ ਗਿਆ ਕੋਈ ਵੀ ਅਪਰਾਧ _____ ਵਜੋਂ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ।
2. _____ ਇੰਟਰਨੈੱਟ ਤੇ ਜ਼ਿੰਮੇਵਾਰ ਵਿਵਹਾਰ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।
3. ਲੈਂਡਫਿਲ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਈ-ਵੇਸਟ ਪ੍ਰਬੰਧਨ _____ ਨੂੰ ਪ੍ਰਦੂਸ਼ਿਤ ਕਰਦਾ ਹੈ।
4. _____ ਕੁਝ ਮਹੱਤਵਪੂਰਨ ਜਾਂ ਗੁਪਤ ਜਾਣਕਾਰੀ ਇਕੱਠੀ ਕਰਨ ਲਈ ਦੂਜਿਆਂ ਦੀ ਨਿੱਜੀ ਗਲਬਾਤ ਨੂੰ ਗੁਪਤ ਜਾਂ ਚੋਰੀ-ਛਿਪੇ ਸੁਣਨ ਦਾ ਤਰੀਕਾ ਹੈ।
5. ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਈਡ ਦਾ ਅਰਥ ਹੈ _____, ਇਕਸਾਰਤਾ ਅਤੇ ਉਪਲਬਧਤਾ।

ਏ. ਹੇਠਾਂ ਦਿੱਤੇ ਕਥਨਾਂ ਵਿੱਚੋਂ ਸਹੀ ਜਾਂ ਗਲਤ ਦੱਸੋ

1. ਸੀ.ਆਈ.ਏ ਟ੍ਰਾਈਡ (CIA Triad) ਦਾ ਅਰਥ ਹੈ ਗੁਪਤਤਾ, ਇਕਸਾਰਤਾ ਅਤੇ ਉਪਲਬਦਤਾ।
2. ਭਾਰਤ ਵਿੱਚ IT ਐਕਟ ਸਾਲ 1995 ਵਿੱਚ ਲਾਗੂ ਹੋਇਆ।
3. ਈ-ਵੇਸਟ (E-Waste) ਮਨੁੱਖੀ ਸਿਹਤ ਲਈ ਚੰਗਾ ਹੈ।
4. ਸਾਈਬਰ ਨੈਤਿਕਤਾ (Cyber Ethics) ਇਹ ਯਕੀਨੀ ਬਣਾਉਂਦੀ ਹੈ ਕਿ ਉਪਭੋਗਤਾ ਆਪਣੇ ਆਪ ਨੂੰ ਆਨਲਾਈਨ ਚਲਾਉਣ ਲਈ ਆਪਣੀਆਂ ਜ਼ਿੰਮੇਵਾਰੀਆਂ ਨੂੰ ਸਮਝਦੇ ਹਨ।
5. ਹੈਕਿੰਗ ਦੀ ਵਰਤੋਂ ਸਿਸਟਮਾਂ ਵਿੱਚ ਕਮਜ਼ੋਰੀਆਂ ਦਾ ਪਰਦਾਫਾਸ਼ ਕਰਨ ਲਈ ਵੀ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
6. ਜਾਸੂਸੀ (Spying) ਖੁਫੀਆ ਉਦੇਸ਼ਾਂ ਲਈ ਵਿਰੋਧੀਆਂ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਗਤੀਵਿਧੀਆਂ 'ਤੇ ਗੁਪਤ ਨਜ਼ਰ ਰੱਖਣ ਦਾ ਕੰਮ ਹੈ।

ਸ. ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ

1. ਸਾਈਬਰ ਅਪਰਾਧ ਦਾ ਵਰਣਨ ਕਰੋ।
2. ਹੈਕਿੰਗ ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਮਤਲਬ ਹੈ ?
3. ਉਦਾਹਰਨ ਦੇ ਨਾਲ ਫਿਸ਼ਿੰਗ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋ।
4. ਸੀ.ਆਈ.ਏ. ਟ੍ਰਾਈਡ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।
5. ਸਾਈਬਰ ਧੱਕੇਸ਼ਾਹੀ ਕੀ ਹੈ ?
6. ਸਾਹਿਤਕ ਚੋਰੀ ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਮਤਲਬ ਹੈ ?
7. ਪਛਾਣ ਦੀ ਚੋਰੀ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।
8. ਈ-ਵੇਸਟ ਦੇ ਵੱਖ-ਵੱਖ ਨੁਕਸਾਨਦੇਹ ਪ੍ਰਭਾਵਾਂ ਦਾ ਵਰਣਨ ਕਰੋ.

ਹ. ਵੱਡੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ

1. ਸਾਈਬਰ ਕ੍ਰਾਈਮ ਕੀ ਹੈ ? ਸਾਈਬਰ ਅਪਰਾਧ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਦਾ ਵਰਣਨ ਕਰੋ।
2. ਸਾਈਬਰ ਸੁਰੱਖਿਆ ਦੀ ਵਿਆਖਿਆ ਕਰੋ। ਸਾਈਬਰ ਅਪਰਾਧ ਨਾਲ ਨਜਿੱਠਣ ਲਈ ਰੋਕਥਾਮ ਉਪਾਅ ਕੀ ਹਨ ?
3. ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰੋ। ਸਾਈਬਰ ਨੈਤਿਕਤਾ ਦੇ ਸਿਧਾਂਤ ਕੀ ਹਨ ?
4. ਈ-ਵੇਸਟ ਤੋਂ ਤੁਹਾਡਾ ਕੀ ਮਤਲਬ ਹੈ ? ਈ-ਵੇਸਟ ਦੇ ਨਿਪਟਾਰੇ ਦੇ ਵੱਖ-ਵੱਖ ਤਰੀਕਿਆਂ ਬਾਰੇ ਦੱਸੋ।

ਅੰਤਿਕਾ (APPENDIX-I)

ਪਾਈਥਨ ਸੈੱਟਅੱਪ ਕਰਨਾ (SETTING UP PYTHON)

ਇਸ ਸੈਕਸ਼ਨ ਵਿੱਚ ਅਸੀਂ ਇਹ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਾਂਗੇ ਕਿ ਕੰਪਿਊਟਰ ਉੱਤੇ ਪਾਈਥਨ ਨੂੰ ਕਿਵੇਂ ਸੈੱਟਅੱਪ ਕਰਨਾ ਹੈ ਤਾਂ ਜੋ ਅਸੀਂ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਰਨ ਅਤੇ ਐਡਿਟ ਕਰ ਸਕੀਏ। ਪਾਈਥਨ ਇਕ ਮੁਫ਼ਤ ਅਤੇ ਓਪਨ ਸੋਰਸ ਭਾਸ਼ਾ ਹੈ, ਜੋ ਸਾਰੇ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮਾਂ ਲਈ ਉਪਲਬਧ ਹੈ। ਇਸਨੂੰ python.org ਤੋਂ ਡਾਊਨਲੋਡ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

ਪਾਈਥਨ ਲਈ ਸਿਸਟਮ ਲੋੜਾਂ (System Requirements for Python):

ਕੰਪਿਊਟਰ ਵਿੱਚ Python 3 ਨੂੰ ਇੰਸਟਾਲ ਕਰਨ ਲਈ ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਸਿਸਟਮ ਲੋੜਾਂ ਹਨ:

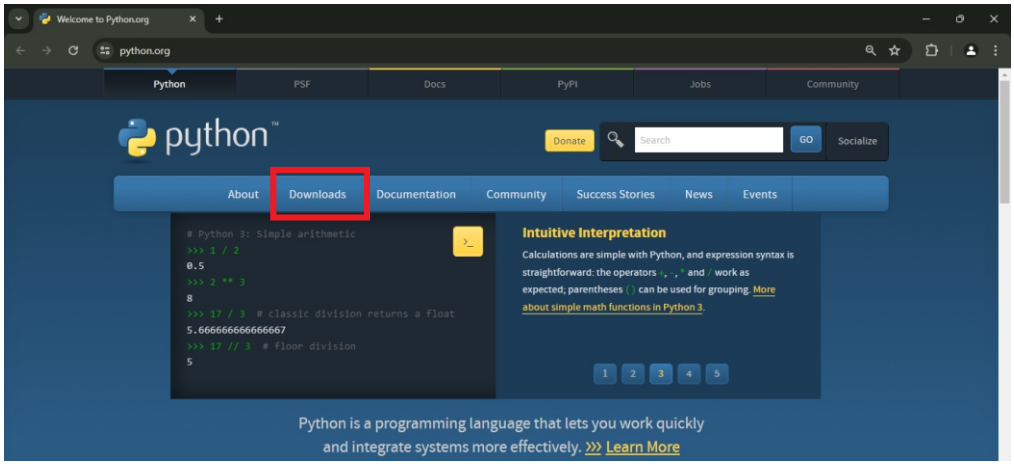
1. **ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ (Operating System):** ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਵਿਚ Linux- Ubuntu 16.04 ਜਾਂ ਇਸ ਤੋਂ ਉਪਰਲਾ ਸੰਸਕਰਣ (higher version), ਜਾਂ ਵਿੰਡੋਜ਼ 7 ਜਾਂ ਕੋਈ ਵੀ ਉਪਰਲਾ ਸੰਸਕਰਣ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।
2. ਸਿਸਟਮ ਵਿੱਚ ਘੱਟੋ-ਘੱਟ 2GB RAM (4GB Preferable) ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। ਉਹ ਐਪਲੀਕੇਸ਼ਨਾਂ ਜਿਹਨਾਂ ਨੂੰ ਮੈਮੋਰੀ ਵਿੱਚ ਵੱਡੇ ਐਰੇ/ਆਬਜੈਕਟਸ ਨੂੰ ਸਟੋਰ ਕਰਨ ਦੀ ਲੋਡ ਹੁੰਦੀ ਹੈ,ਉਹਨਾਂ ਲਈ ਵਧੇਰੇ RAM ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ, ਜਦੋਂ ਕਿ ਉਹ ਐਪਲੀਕੇਸ਼ਨਾਂ ਜਿਹਨਾਂ ਨੂੰ ਬਹੁਤ ਸਾਰੀਆਂ ਗਣਨਾਵਾਂ ਜਾਂ ਕਾਰਜਾਂ ਨੂੰ ਤੇਜ਼ੀ ਨਾਲ ਕਰਨ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ, ਉਹਨਾਂ ਲਈ ਇੱਕ ਤੇਜ਼ ਪ੍ਰੋਸੈਸਰ (Processor) ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ।

ਅਸੀਂ Python 3.7 ਜਾਂ ਇਸ ਤੋਂ ਉਪਰਲੇ ਸੰਸਕਰਣ ਅਤੇ ਸੰਬੰਧਿਤ ਪੈਕੇਜਾਂ ਨੂੰ ਇੰਸਟਾਲ ਕਰਾਂਗੇ, ਕਿਰਪਾ ਕਰਕੇ ਆਪਣੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਦੇ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਅਨੁਸਾਰ ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਗਈਆਂ ਇੰਸਟਾਲੇਸ਼ਨ ਹਦਾਇਤਾਂ ਦੀ ਪਾਲਣਾ ਕਰੋ:

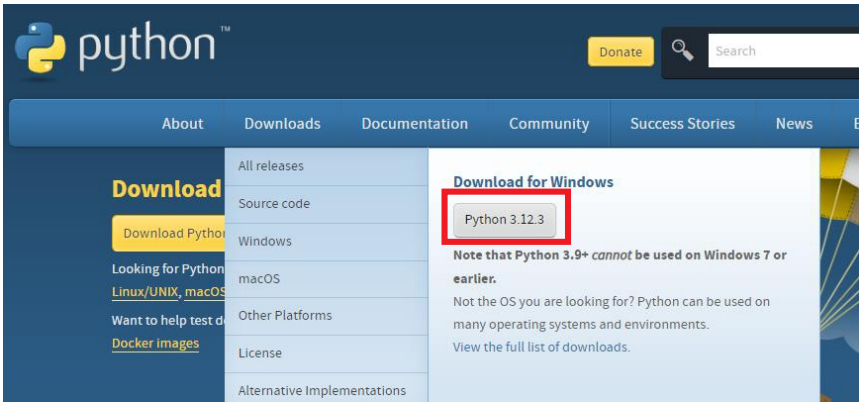
ਪਾਈਥਨ ਨੂੰ ਡਾਊਨਲੋਡ ਅਤੇ ਇੰਸਟਾਲ ਕਰਨਾ (Downloading and Installation of Python):

ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਉਪਰ Python 3 ਇੰਸਟਾਲ ਕਰਨ ਦੀ ਵਿਧੀ ਹੇਠਾਂ ਦਿੱਤੀ ਗਈ ਹੈ:

1. ਇੰਸਟਾਲੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ (Installation Procedure) ਦੀ ਪਾਲਣਾ ਕਰਨ ਲਈ ਸਾਨੂੰ ਇੰਟਰਨੈਟ ਨਾਲ ਕੁਨੈਕਟ ਹੋਣਾ ਪਵੇਗਾ।
2. ਕੋਈ ਵੀ ਵੈਬ ਬ੍ਰਾਊਜ਼ਰ ਓਪਨ ਕਰੋ ਅਤੇ https://www.python.org ਵੈਬਸਾਈਟ ਖੋਲੋ।



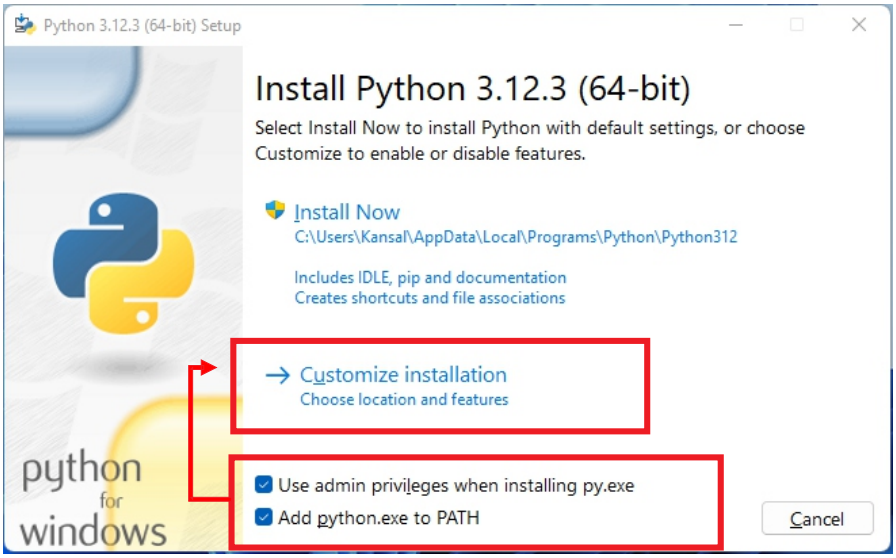
3. ਪਾਈਥਨ ਲਈ ਨਵੀਨਤਮ (latest) ਇੰਸਟਾਲਰ (Windows Installer (32-bit) ਜਾਂ Windows Installer (64-bit) (ਤੁਹਾਡੇ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੇ ਅਨੁਸਾਰ) ਨੂੰ ਡਾਊਨਲੋਡ ਕਰੋ।



Files

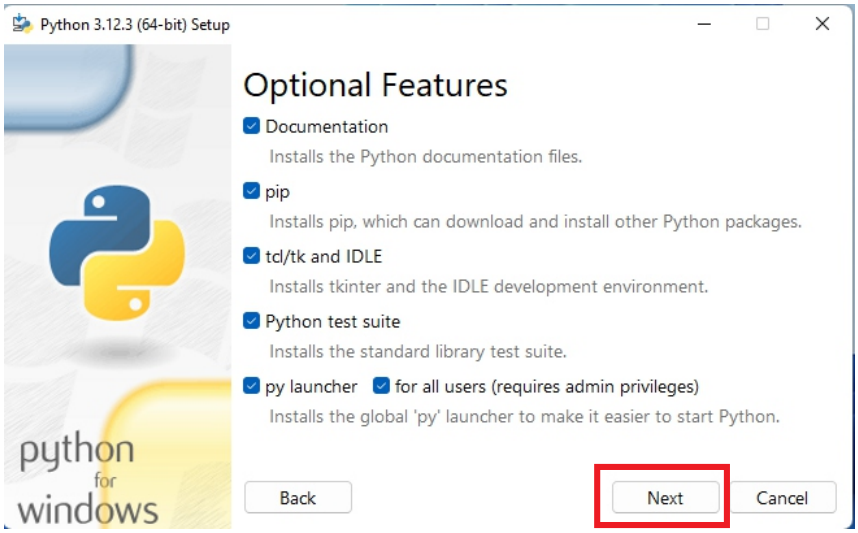
Version	Operating System	Description	MD5 Sum	File Size
Gzipped source tarball	Source release		3c5498a34d5226c9b746b1199f0bf2d9	25.9 MB
XZ compressed source tarball	Source release		8defb33f0c37aa4bdd3a38ba52abde4e	19.7 MB
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	6114a3bb9b288f23ab38dbbb959be1bf	43.6 MB
Windows installer (64-bit)	Windows	Recommended	c86949710e0471a065db970290819489	25.5 MB
Windows installer (ARM64)	Windows	Experimental	ef016521b5a147f3ded730801d36a350	24.7 MB
Windows embeddable package (64-bit)	Windows		38cce2bf5b4de76db19a31f46a0720de	10.5 MB
Windows embeddable package (32-bit)	Windows		65d873c723db66d6746e9872df5a715e	9.4 MB
Windows embeddable package (ARM64)	Windows		3229271cac55ff913aad1bb46ebc9931	9.8 MB
Windows installer (32-bit)	Windows		a95c4fbdce0b6a22ca7cfb450f57c616	24.2 MB

4. ਡਾਊਨਲੋਡ ਪੂਰਾ ਹੋਣ ਤੋਂ ਬਾਅਦ ਇੰਸਟਾਲੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ ਸ਼ੁਰੂ ਕਰਨ ਲਈ ਇੰਸਟਾਲਰ ਫਾਈਲ 'ਤੇ ਡਬਲ ਕਲਿੱਕ ਕਰੋ।

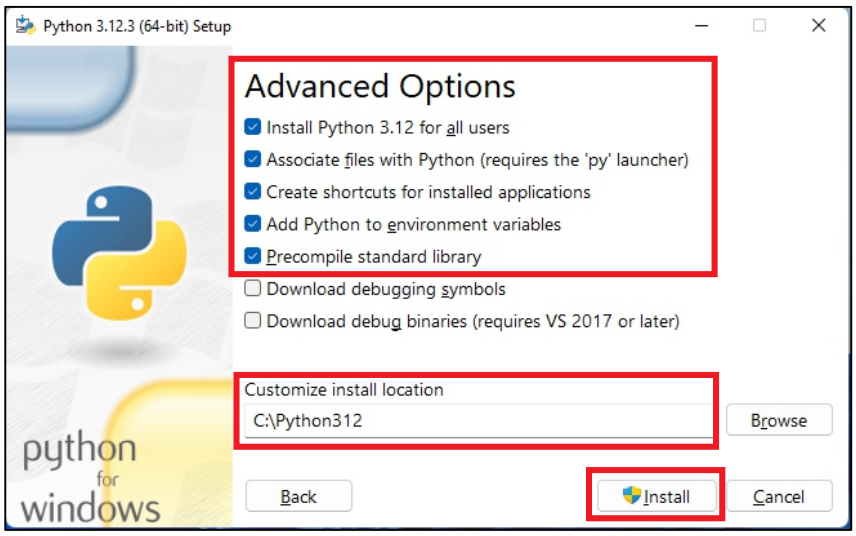


ਮਹੱਤਵਪੂਰਨ ਨੋਟ: ਜਦੋਂ ਇੰਸਟਾਲਰ ਵਿੰਡੋ ਖੁੱਲ੍ਹਦੀ ਹੈ, ਤਾਂ "Add Python 3.x.xx to PATH" ਆਪਸ਼ਨ ਵਾਲੇ ਚੈੱਕਬਾਕਸ 'ਤੇ ਕਲਿੱਕ ਕਰਨਾ ਨਾ ਭੁੱਲੋ।

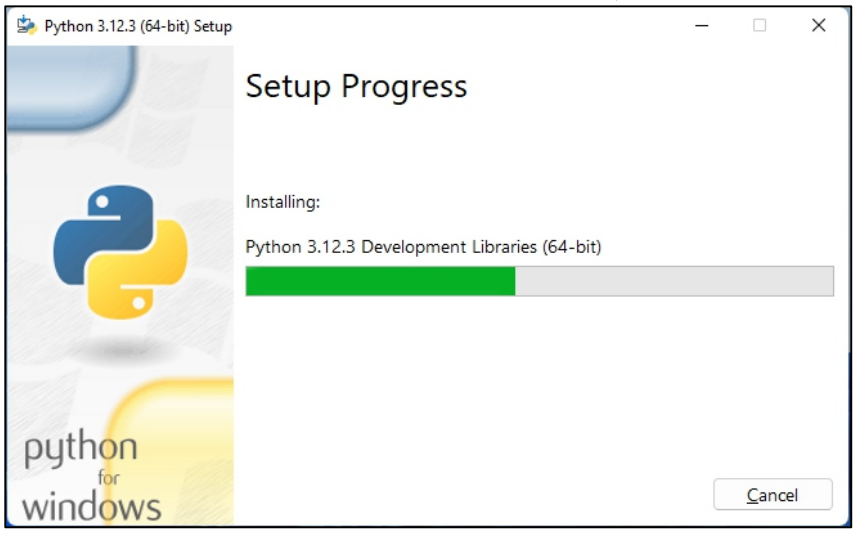
5. ਇੰਸਟਾਲਰ ਅਨੁਸਾਰ ਨਿਰਦੇਸ਼ਾਂ (Instructions) ਦੀ ਪਾਲਣਾ ਕਰੋ।



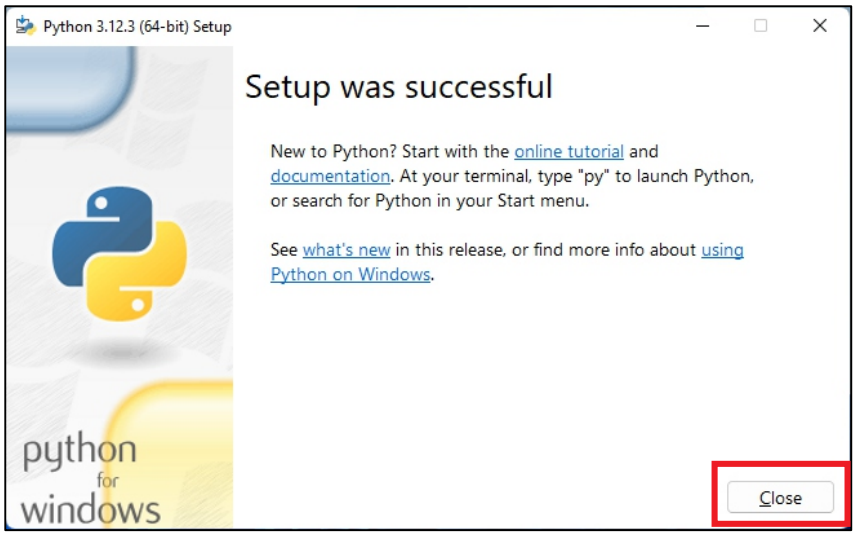
ਚਿੱਤਰ: Advanced Options ਸਿਲੈਕਟ ਕਰੋ, install location ਸੈਟ ਕਰੋ & Install ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।



ਚਿੱਤਰ: ਪਾਈਥਨ ਇੰਸਟਾਲੇਸ਼ਨ ਦਾ ਸੈਟਅੱਪ ਪ੍ਰੋਗਰੈਸ



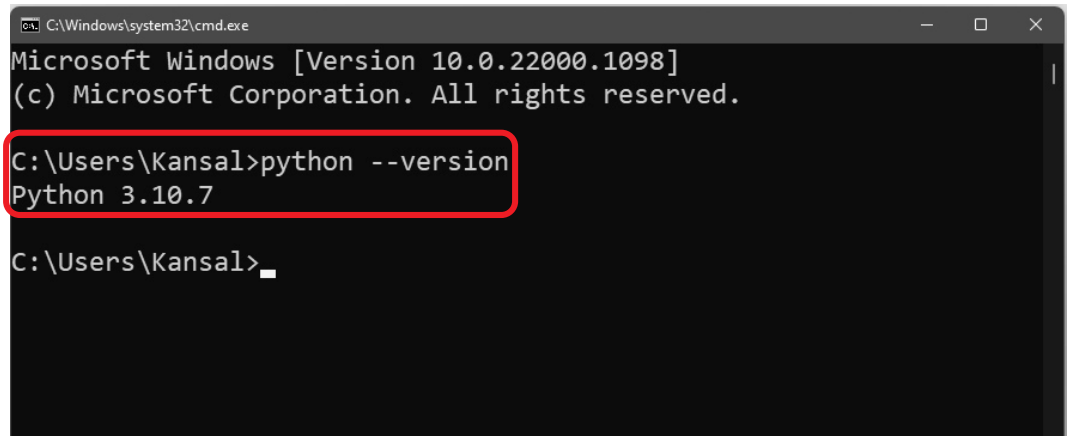
ਕਰਨਾ



ਚਿੱਤਰ: ਪਾਈਥਨ ਸੈਟਅੱਪ ਪੂਰਾ ਹੋ ਗਿਆ

ਕਿਵੇਂ ਜਾਂਚ ਕਰੀਏ ਕਿ ਪਾਈਥਨ ਸਹੀ ਢੰਗ ਨਾਲ ਇੰਸਟਾਲ ਹੋ ਗਈ ਹੈ ਜਾਂ ਨਹੀਂ (HOW TO CHECK IF PYTHON IS INSTALLED CORRECTLY):

1. CMD ਵਿੰਡੋ ਓਪਨ ਕਰੋ (ਵਿੰਡੋ ਕੀਅ + R ਦਬਾਓ –> CMD ਟਾਈਪ ਕਰੋ -> ਐਂਟਰ ਕੀਅ ਦਬਾਓ)
2. CMD ਵਿੰਡੋ ਵਿੱਚ "python --version" ਜਾਂ "python3 --version" ਕਮਾਂਡ ਟਾਈਪ ਕਰੋ ਅਤੇ ਐਂਟਰ ਕੀਅ ਦਬਾਓ।
3. ਸਾਨੂੰ ਅੱਗੇ ਦਿਖਾਏ ਚਿੱਤਰ ਅਨੁਸਾਰ ਆਉਟਪੁੱਟ ਪ੍ਰਾਪਤ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। (ਆਉਟਪੁੱਟ ਪਾਈਥਨ ਦਾ ਇੰਸਟਾਲਡ ਸੰਸਕਰਣ (Version) ਦਿਖਾਵੇਗਾ। ਅਸੀਂ ਕੰਪਿਊਟਰ ਵਿੱਚ ਪਾਈਤਨ 3.10.7 ਇੰਸਟਾਲ ਕੀਤਾ ਹੈ।)



ਪਾਇਥਨ IDEs ਨਾਲ ਜਾਣ-ਪਛਾਣ (INTRODUCTION TO PYTHON IDEs):

IDE (ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਿਵੈਲਪਮੈਂਟ ਇਨਵਾਇਰਨਮੈਂਟ) ਇੱਕ ਸਾਫਟਵੇਅਰ ਹੁੰਦਾ ਹੈ ਜੋ ਸਾਫਟਵੇਅਰ ਡਿਵੈਪਲਮੈਂਟ ਲਈ ਸਮਰਪਿਤ (dedicated) ਹੁੰਦਾ ਹੈ। ਇਹ ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮਰਾਂ ਨੂੰ ਵਿਆਪਕ ਪੱਧਰ ਤੇ ਸਾਫਟਵੇਅਰ ਡਿਵੈਪਲਮੈਂਟ ਲਈ ਕਈ ਯੋਗਤਾਵਾਂ (abilities) ਪੇਸ਼ ਕਰਦਾ ਹੈ। IDEs ਖਾਸ ਤੌਰ 'ਤੇ ਸਾਫਟਵੇਅਰ ਡਿਵੈਲਪਮੈਂਟ ਲਈ ਤਿਆਰ ਕੀਤੇ ਗਏ ਕਈ ਟੂਲਜ਼ (tools) ਨੂੰ ਇੰਟੀਗ੍ਰੇਟ ਕਰਦੇ ਹਨ। ਇਹਨਾਂ ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਹੇਠਾਂ ਦਿਤੇ ਟੂਲਜ਼ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ:

- ਸੋਰਸ ਕੋਡ ਨੂੰ ਹੈਂਡਲ ਕਰਨ ਲਈ ਇੱਕ ਐਡੀਟਰ (Editor): ਆਮ ਤੌਰ 'ਤੇ, ਇਹ ਐਡੀਟਰ ਸਿੰਟੈਕਸ ਹਾਈਲਾਈਟਿੰਗ (Syntax Highlighting) ਅਤੇ ਆਟੋ ਕੋਡ-ਕੰਪਲੀਸ਼ਨ (Auto Code-

Completion) ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ।

- ਬਿਲਡ (Build), ਐਗਜ਼ੀਕਿਊਸ਼ਨ (Execution), ਅਤੇ ਡੀਬਗਿੰਗ ਟੂਲ (Debugging Tools)

IDEs ਦੀਆਂ ਅਜਿਹੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਪ੍ਰੋਗਰਾਮਰਾਂ ਨੂੰ ਆਪਣੀ ਉਤਪਾਦਕਤਾ (Productivity) ਵਧਾਉਣ ਵਿੱਚ ਮਦਦ ਕਰਦੀਆਂ ਹਨ। ਜ਼ਿਆਦਾਤਰ IDE ਬਹੁਤ ਸਾਰੀਆਂ ਵੱਖ-ਵੱਖ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਨੂੰ ਸਪੋਰਟ ਕਰਦੇ ਹਨ ਅਤੇ ਇਹਨਾਂ ਵਿੱਚ ਕਈ ਹੋਰ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਵੀ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ।

ਪਾਈਥਨ ਸ਼ੈੱਲ (Python Shell) ਜਾਂ IDLE (ਇੰਟੀਗ੍ਰੇਟਿਡ ਡਿਵੈਲਪਮੈਂਟ ਐਂਡ ਲਰਨਿੰਗ ਇਨਵਾਇਰਮੈਂਟ) ਇੱਕ ਡਿਫਾਲਟ ਐਡੀਟਰ ਹੈ ਜੋ ਪਾਈਥਨ ਨਾਲ ਆਪਣੇ ਆਪ ਇੰਸਟਾਲ ਹੋ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਧਾਰਨ IDE/ਐਡੀਟਰ ਸ਼ੁਰੂਆਤੀ ਪੱਧਰ ਦੇ ਡਿਵੈਲਪਰਾਂ ਲਈ ਢੁਕਵਾਂ ਹੈ। IDLE ਟੂਲ ਨੂੰ Mac OS, Windows ਅਤੇ Linux ਪਲੇਟਫਾਰਮਾਂ 'ਤੇ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਹ ਮੁਫਤ ਵਿਚ (free of cost) ਪ੍ਰਦਾਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। IDLE ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਪਾਈਥਨ ਕੋਡ ਲਿਖਣਾ ਸਧਾਰਨ ਲੇਵਲ ਦੇ ਪ੍ਰੋਗਰਾਮਾਂ ਲਈ ਬਹੁਤ ਵਧੀਆ ਹੁੰਦਾ ਹੈ, ਪਰ IDLE ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੱਡੇ ਪ੍ਰੋਗਰਾਮਿੰਗ ਪ੍ਰੋਜੈਕਟਾਂ ਨੂੰ ਹੈਂਡਲ ਕਰਨਾ ਮੁਸ਼ਕਲ ਹੋ ਜਾਂਦਾ ਹੈ। ਇੱਕ ਪ੍ਰੋਫੈਸ਼ਨਲ IDE ਜਾਂ ਇੱਕ ਵਧੀਆ-ਸਮਰਪਿਤ (good-dedicated) ਕੋਡ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰਨਾ, ਕੋਡਿੰਗ ਨੂੰ ਮਜ਼ੇਦਾਰ ਬਣਾਉਂਦਾ ਹੈ।

ਪਾਈਥਨ ਲਈ ਬਹੁਤ ਸਾਰੇ IDE ਅਤੇ ਕੋਡ ਐਡੀਟਰ ਉਪਲਬਧ ਹਨ। ਵੱਖ-ਵੱਖ IDE/ਕੋਡ ਐਡੀਟਰ ਵੱਖ-ਵੱਖ ਉਦੇਸ਼ਾਂ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਅਸੀਂ ਆਪਣੀਆਂ ਜ਼ਰੂਰਤਾਂ ਅਨੁਸਾਰ ਕੋਈ ਵੀ IDE/ਕੋਡ ਐਡੀਟਰ ਚੁਣ ਸਕਦੇ ਹਾਂ। ਅੱਗੇ ਕੁੱਝ ਆਮ ਵਰਤੇ ਜਾਂਦੇ Python IDE ਅਤੇ ਕੋਡ ਐਡੀਟਰਾਂ ਦਾ ਵਰਨਣ ਕੀਤਾ ਗਿਆ ਹੈ:

- **ਪਾਇਚਾਮ (PyCharm):** ਇਹ ਪਾਈਥਨ ਲਈ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਸਭ ਤੋਂ ਵਧੀਆ, ਪੂਰੀਆਂ-ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਵਾਲਾ ਸਮਰਪਿਤ IDE (full-featured dedicated IDE) ਹੈ। ਇਹ JetBrains ਦੁਆਰਾ ਬਣਾਇਆ ਗਿਆ ਸਭ ਤੋਂ ਵੱਧ ਵਰਤਿਆ ਜਾਣ ਵਾਲਾ ਪਾਈਥਨ IDE ਹੈ। ਇਹ ਪੇਡ (Paid) (ਪ੍ਰੋਫੈਸ਼ਨਲ) ਅਤੇ ਫ੍ਰੀ (Free) ਓਪਨ-ਸੋਰਸ (ਕਮਿਊਨਿਟੀ) ਸੰਸਕਰਣਾਂ ਵਿੱਚ ਉਪਲਬਧ ਹੈ। ਅਸੀਂ ਇਸਨੂੰ Windows, Mac OS X, ਅਤੇ Linux ਪਲੇਟਫਾਰਮਾਂ 'ਤੇ ਆਸਾਨੀ ਨਾਲ ਇੰਸਟਾਲ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਹ ਪ੍ਰੋਫੈਸ਼ਨਲ ਡਿਵੈਲਪਰਾਂ ਲਈ ਢੁਕਵਾਂ ਹੈ ਅਤੇ ਵੱਡੇ ਪਾਈਥਨ ਪ੍ਰੋਜੈਕਟਾਂ ਨੂੰ ਡਿਵੈਪਲ ਕਰਨ ਦੀ ਸਹੂਲਤ ਦਿੰਦਾ ਹੈ। PyCharm ਸਿੱਧੇ ਤੌਰ 'ਤੇ (directly) ਪਾਈਥਨ ਡਿਵੈਪਲਮੈਂਟ ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ। ਅਸੀਂ ਸਿਰਫ ਇੱਕ ਨਵੀਂ ਫਾਈਲ ਖੋਲ੍ਹ ਕੇ ਉਸ ਵਿਚ ਪਾਈਥਨ ਕੋਡ ਲਿਖਣਾ ਸ਼ੁਰੂ ਕਰ ਸਕਦੇ ਹਾਂ। ਅਸੀਂ ਪਾਈਥਨ ਨੂੰ ਸਿੱਧੇ PyCharm ਦੇ ਅੰਦਰ ਚਲਾ (Execute) ਸਕਦੇ ਹਾਂ ਅਤੇ ਡੀਬੱਗ (Debug) ਕਰ ਸਕਦੇ ਹਾਂ।
- **ਵਿਜ਼ੁਅਲ ਸਟੂਡੀਓ ਕੋਡ (Visua Studio Code):** ਇਹ ਮਾਈਕ੍ਰੋਸੌਫਟ ਦੁਆਰਾ ਵਿਕਸਤ ਕੀਤਾ ਗਿਆ ਇੱਕ ਓਪਨ-ਸੋਰਸ (ਅਤੇ ਮੁਫਤ) ਕੋਡ ਐਡੀਟਰ ਹੈ। ਇਹ ਲਾਈਟਵੇਟ (lightweight) ਹੈ ਅਤੇ ਅਜਿਹੀਆਂ ਸ਼ਕਤੀਸ਼ਾਲੀ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਨਾਲ ਆਉਂਦਾ ਹੈ ਜੋ ਸਿਰਫ ਕੁਝ ਪੇਡ (Paid) IDEs ਪੇਸ਼ ਕਰਦੇ ਹਨ।
- **ਸਬਲਾਈਮ ਟੈਕਸਟ 3 (Sublime Text 3):** ਇਹ ਇੱਕ ਬਹੁਤ ਹੀ ਪ੍ਰਸਿੱਧ ਕੋਡ ਐਡੀਟਰ ਹੈ। ਇਹ ਪਾਈਥਨ ਸਮੇਤ ਕਈ ਭਾਸ਼ਾਵਾਂ ਨੂੰ ਸਪੋਰਟ ਕਰਦਾ ਹੈ। ਇਸ ਨੂੰ ਬਹੁਤ ਜ਼ਿਆਦਾ ਅਨੁਕੂਲਿਤ (Highly Customizable) ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਇਸਦੀ ਵਰਤੋਂ ਵੀ ਮੁਫਤ (free of cost) ਵਿਚ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।
- **ਜੁਪਾਈਟਰ ਨੋਟਬੁੱਕ (Jupyter Notebook):** ਇਹ ਇੱਕ ਸਰਵਰ-ਕਲਾਇੰਟ ਐਪਲੀਕੇਸ਼ਨ ਹੈ ਜੋ ਇੱਕ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਰਾਹੀਂ ਨੋਟਬੁੱਕ ਦਸਤਾਵੇਜ਼ਾਂ ਨੂੰ ਐਡਿਟ ਕਰਨ ਅਤੇ ਚਲਾਉਣ (Run) ਦੀ ਆਗਿਆ ਦਿੰਦੀ ਹੈ। ਇਹ ਡਾਟਾ ਸਾਇੰਸ ਦੇ ਖੇਤਰ ਵਿੱਚ ਵੱਡੇ ਪੱਧਰ 'ਤੇ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਵਰਤਣਾ ਆਸਾਨ ਹੈ, ਇੰਟਰਐਕਟਿਵ (Interactive) ਹੈ ਅਤੇ ਲਾਈਵ ਕੋਡ ਸ਼ੇਅਰਿੰਗ (live Code Sharing) ਅਤੇ ਵਿਜ਼ੁਅਲਾਈਜ਼ੇਸ਼ਨ (Visualization) ਦੀ ਇਜਾਜ਼ਤ ਦਿੰਦਾ ਹੈ। ਇਹ ਵੀ ਮੁਫਤ (Free of Cost) ਪ੍ਰਦਾਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।
- **ਸਪਾਈਡਰ (Spyder):** ਸਪਾਈਡਰ ਇਕ ਓਪਨ-ਸੋਰਸ IDE ਹੈ। ਇਹ ਆਮ ਤੌਰ 'ਤੇ ਸਾਇੰਟੀਫਿੱਕ ਡਿਵੈਲਪਮੈਂਟ

(scientific development) ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਐਨਾਕਾਂਡਾ ਡਿਸਟਰੀਬਿਊਸ਼ਨ (Anaconda distribution) ਦੇ ਨਾਲ ਆਉਂਦਾ ਹੈ, ਜੋ ਕਿ ਡਾਟਾ ਸਾਇੰਸ ਅਤੇ ਮਸ਼ੀਨ ਲਰਨਿੰਗ ਲਈ ਪ੍ਰਸਿੱਧ ਹੈ। ਇਸ ਦੀ ਵਰਤੋਂ ਵੀ ਮੁਫਤ ਵਿਚ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

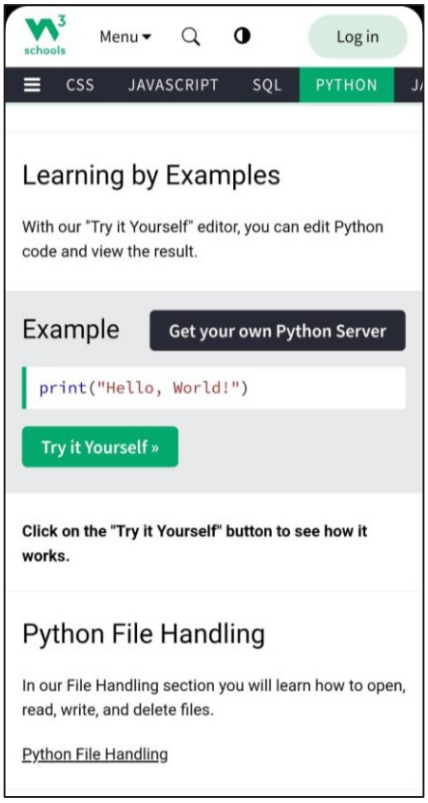
ਮੋਬਾਈਲ 'ਤੇ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਕਿਵੇਂ ਕਰੀਏ ? (HOW TO DO PYTHON PROGRAMMING ON MOBILE?):

ਇੱਕ ਐਂਡਰੌਇਡ ਡਿਵਾਈਸ ਉੱਤੇ ਪਾਈਥਨ ਕੋਡ ਨੂੰ ਚਲਾਉਣ ਦੇ ਕਈ ਤਰੀਕੇ ਹਨ: ਅਸੀਂ ਐਂਡਰਾਇਡ ਐਪਸ, ਜਿਵੇਂ ਕਿ: QPython, PyDroid, ਜਾਂ Python for Android ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਪਾਈਥਨ ਕੋਡ ਨੂੰ ਚਲਾ ਸਕਦੇ ਹਾਂ। ਇਹਨਾਂ ਐਪਸ ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਇੱਕ ਬਿਲਟ-ਇਨ ਪਾਈਥਨ ਇੰਟਰਪ੍ਰੈਟਰ ਦੇ ਨਾਲ-ਨਾਲ ਹੋਰ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਜਿਵੇਂ ਕਿ ਕੋਡ ਐਡੀਟਰ ਵੀ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ।

ਪਾਈਥਨ ਕੋਡ ਨੂੰ ਲਿਖਣ ਅਤੇ ਚਲਾਉਣ ਦਾ ਇੱਕ ਹੋਰ ਤਰੀਕਾ ਇਹ ਹੈ ਕਿ ਬਹੁਤ ਸਾਰੀਆਂ ਵੈਬਸਾਈਟਾਂ ਹਨ ਜੋ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮ ਲਈ ਆਨਲਾਈਨ ਕੰਪਾਈਲਰ/ਇੰਟਰਪ੍ਰੈਟਰ ਪ੍ਰਦਾਨ ਕਰਦੀਆਂ ਹਨ। ਸਭ ਤੋਂ ਮਸ਼ਹੂਰ ਵੈਬਸਾਈਟਾਂ ਵਿੱਚੋਂ ਇੱਕ ਹੈ: https://www.w3schools.com। ਇਹ ਸਾਈਟ ਵੱਖ-ਵੱਖ ਭਾਸ਼ਾਵਾਂ ਜਿਵੇਂ ਕਿ HTML. CSS, JAVA SCRIPT, JAVA, PYTHON ਆਦਿ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਲਈ ਕੋਡਿੰਗ ਦੀ ਸਹੂਲਤ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। ਅਸੀਂ ਇਸ ਵੈੱਬਸਾਈਟ 'ਤੇ ਆਪਣੇ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਬਹੁਤ ਆਸਾਨੀ ਨਾਲ ਕੋਡ ਕਰ ਸਕਦੇ ਹਾਂ। ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਲਈ ਆਪਣੇ ਮੋਬਾਈਲ ਡਿਵਾਈਸ 'ਤੇ ਕਿਸੇ ਵੀ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ, ਜਿਵੇਂ ਕਿ ਗੂਗਲ ਕਰੋਮ, ਨੂੰ ਖੋਲ੍ਹੋ ਅਤੇ ਹੇਠਾਂ ਦਿੱਤੇ ਲਿੰਕ ਨੂੰ ਓਪਨ ਕਰੋ:

https://www/w3schools.com/python/default.asp

ਇਹ ਤੁਹਾਡੀ ਮੋਬਾਈਲ ਸਕ੍ਰੀਨ 'ਤੇ ਹੇਠਾਂ ਦਿੱਤੇ ਵੈਬਪੇਜ਼ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇਗਾ:



ਵੈਬ ਪੇਜ਼ ਵਿਚ ਹੇਠਾਂ ਵੱਲ ਸਕਰੋਲ ਕਰੋ ਅਤੇ Try it Yourself » ਲਿੰਕ ਬਟਨ ਲੱਭੋ:

ਪਾਈਥਨ ਕੰਪਾਈਲਰ ਵਾਲਾ ਵੇਬਪੇਜ਼ ਓਪਨ ਕਰਨ ਲਈ ਇਸ ਲਿੰਗ ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰ। ਹੁਣ, ਤੁਹਾਡੇ ਮੋਬਾਈਲ ਡਿਵਾਈਸ 'ਤੇ ਹੇਠਾਂ ਦਿੱਤੀ ਸਕ੍ਰੀਨ ਦਿਖਾਈ ਦੇਵੇਗੀ:

ਆਪਣਾ ਕੋਡ ਟੈਕਸਟ ਐਡੀਟਰ ਵਿੱਚ ਟਾਈਪ ਕਰੋ, ਜਿਵੇਂ ਕਿ ਉਪਰੋਕਤ ਚਿੱਤਰ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਹੁਣ ਆਪਣੇ ਕੋਡ ਨੂੰ ਚਲਾਉਣ ਲਈ Run ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰੋ। ਆਉਟਪੁੱਟ ਨੂੰ ਆਉਟਪੁੱਟ ਸਕਰੀਨ ਵਿੱਚ ਟੈਕਸਟ ਐਡੀਟਰ ਦੇ ਹੇਠਾਂ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਜਾਵੇਗਾ ਜਿਵੇਂ ਕਿ ਉੱਪਰ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ, ਜੇਕਰ ਸਾਡੇ ਕੋਲ ਘਰ ਵਿੱਚ ਕੰਪਿਊਟਰ ਜਾਂ ਲੈਪਟਾਪ ਨਹੀਂ ਹੈ ਤਾਂ ਅਸੀਂ ਆਪਣੇ ਮੋਬਾਈਲ 'ਤੇ ਵੀ ਪਾਈਥਨ ਪ੍ਰੋਗਰਾਮਜ਼ ਦੀ ਪ੍ਰੈਕਟਿੱਸ ਕਰ ਸਕਦੇ ਹਾਂ।

ਅੰਤਿਕਾ (Appendix-II)
MYSQL SERVER ਇੰਸਟਾਲ ਕਰਨਾ

ਜਦੋਂ ਅਸੀਂ ਵਿੰਡੋ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਾਲੇ ਕੰਪਿਊਟਰ ਉੱਪਰ MySQL ਸਿੱਖਣਾ ਸ਼ੁਰੂ ਕਰਦੇ ਹਾਂ ਤਾਂ ਸਭ ਤੋਂ ਪਹਿਲਾ ਕਦਮ MySQL ਸਰਵਰ ਨੂੰ ਇੰਸਟਾਲ ਕਰਨਾ ਹੁੰਦਾ ਹੈ। ਅਸੀਂ MySQL ਕਮਿਊਨਿਟੀ ਐਡੀਸ਼ਨ ਫੋਰਮ ਤੋਂ ਇਸਨੂੰ ਡਾਊਨਲੋਡ ਕਰ ਸਕਦੇ ਹਾਂ। MySQL ਦੁਨੀਆ ਦੇ ਸਭ ਤੋਂ ਪ੍ਰਸਿੱਧ ਓਪਨ ਸੋਰਸ ਡੇਟਾਬੇਸ ਦਾ ਮੁਫਤ ਵਿੱਚ ਡਾਊਨਲੋਡ ਕਰਨ ਯੋਗ ਵਰਜ਼ਨ ਹੁੰਦਾ ਹੈ। ਇਹ GPL ਲਾਇਸੰਸ ਦੇ ਅਧੀਨ ਉਪਲਬਧ ਹੁੰਦਾ ਹੈ ਅਤੇ ਓਪਨ ਸੋਰਸ ਡਿਵੈਲਪਰਾਂ ਦੇ ਇੱਕ ਵਿਸ਼ਾਲ ਸਮੂਹ ਦੁਆਰਾ ਸਮਰਥਿਤ ਹੁੰਦਾ ਹੈ।

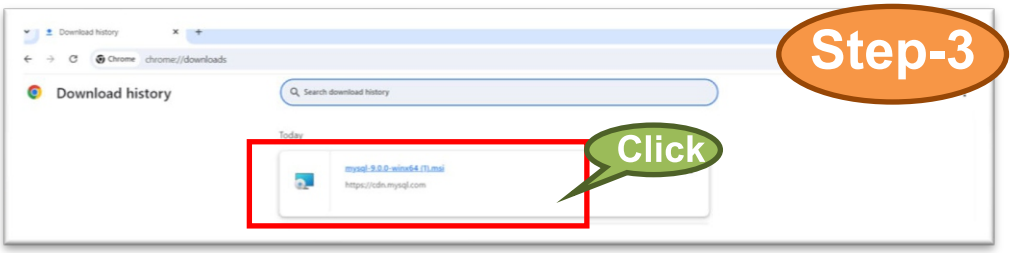
ਇੰਸਟਾਲੇਸ਼ਨ ਸ਼ੁਰੂ ਕਰਨ ਲਈ ਸਾਨੂੰ ਇਹ ਯਕੀਨੀ ਬਣਾਉਣਾ ਲਾਜ਼ਮੀ ਹੈ ਕਿ ਸਾਡੇ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਉੱਪਰ ਇੱਕ ਹਾਈ-ਸਪੀਡ ਇੰਟਰਨੈਟ ਕਨੈਕਸ਼ਨ ਮੌਜੂਦ ਹੋਵੇ।

ਸਭ ਤੋਂ ਪਹਿਲਾਂ, ਅਸੀਂ MSI (Microsoft Software Installer) ਜਾਂ ZIP (ZIP ਇੱਕ ਕਿਸਮ ਦੀ ਆਰਕਾਈਵ ਫਾਈਲ ਫਾਰਮੈਟ ਹੈ) ਫਾਈਲ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ https://dev.mysql.com/downloads/mysql/ ਲਿੰਕ ਤੇ ਪਹੁੰਚ ਕਰਾਂਗੇ ਅਤੇ ਇੱਕ ਇੰਸਟਾਰਲ ਫਾਈਲ ਡਾਊਨਲੋਡ ਕਰਾਂਗੇ। ਇਹ ਫਾਈਲ ਦੀ ਮਦਦ ਨਾਲ ਹੀ ਸਾਡੇ ਕੰਪਿਊਟਰ ਉੱਪਰ MySQL Server ਇੰਸਟਾਲ ਕੀਤਾ ਜਾ ਸਕੇਗਾ। ਦਿੱਤੇ URL (ਯੂਨੀਫਾਰਮ ਰਿਸੋਰਸ ਲੋਕੇਟਰ) ਤੇ ਪਹੁੰਚ ਕਰਨ ਤੋਂ ਬਾਅਦ ਸਾਡੇ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿੱਚ ਹੇਠਾਂ ਦਿੱਤੀ ਸਕ੍ਰੀਨ ਦਿਖਾਈ ਦੇਵੇਗੀ:

ਇੱਥੇ ਅਸੀਂ ਫਾਈਲ ਫਾਰਮੈਟ ਦੀ ਚੋਣ ਕਰਕੇ ਲੋੜਿੰਦੇ "ਡਾਊਨਲੋਡ" ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ ਜਿਸ ਫਾਰਮੈੱਟ ਵਿੱਚ ਅਸੀਂ ਆਪਣੇ ਇੰਸਟਾਲਰ ਨੂੰ ਡਾਊਨਲੋਡ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ। ਸੰਬੰਧਿਤ "ਡਾਊਨਲੋਡ" ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰਨ ਤੋਂ ਬਾਅਦ ਅਗਲੇਰੀ ਦਿੱਤੀ ਸਕ੍ਰੀਨ ਨਜ਼ਰ ਆਵੇਗੀ।

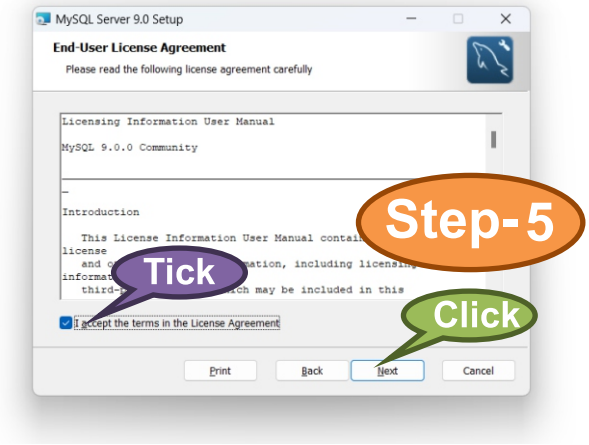
ਇੱਥੇ ਅਸੀਂ ਓਰੇਕਲ ਕੰਪਨੀ ਦੇ ਵੱਖ-ਵੱਖ ਪ੍ਰੋਡਕਟਸ ਤੱਕ ਪਹੁੰਚ ਕਰਨ ਲਈ ਇੱਕ ਮੁਫਤ Oracle ਅਕਾਉਂਟ ਬਣਾ ਸਕਦੇ ਹਾਂ। ਇਸਦਾ ਇੱਕ ਵਿਕਲਪਕ ਇਹ ਵੀ ਹੈ ਕਿ ਅਸੀਂ "No thanks, just start my download" ਕੇ ਕਲਿੱਕ ਕਰਕੇ ਡਾਊਨਲੋਡ ਸ਼ੁਰੂ ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਅਕਾਉਂਟ ਬਣਾਉਣ ਦਾ ਪੜਾਅ ਛੱਡ ਸਕਦੇ ਹਾਂ। ਇਹ ਵਿਕਲਪ ਸਮਾਂ ਬਚਾਉਣ ਵਾਲਾ ਅਤੇ ਵਧੇਰੇ ਸੁਵਿਧਾਜਨਕ ਹੈ।

ਉਕਤ ਆਪਸ਼ਨ ਤੇ ਕਲਿੱਕ ਕਰਨ ਤੋਂ ਬਾਅਦ, ਡਾਉਨਲੋਡ ਸ਼ੁਰੂ ਹੋ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇੰਸਟਾਲਰ ਸਾਡੇ ਡਾਊਨਲੋਡ ਦੇ ਨਿਰਦਾਰਤ ਸਥਾਨ ਤੇ ਡਾਊਨਲੋਡ ਹੋ ਜਾਵੇਗਾ। ਅਸੀਂ ਕੀਬੋਰਡ ਤੋਂ Ctrl+J ਬਟਨ ਦਬਾ ਕੇ ਉਸ ਸਥਾਨ ਤੱਕ ਪਹੁੰਚ ਕਰ ਸਕਦੇ ਹਾਂ ਜਾਂ ਅਸੀਂ ਵਿੰਡੋਜ਼ ਐਕਸਪਲੋਰਰ ਤੋਂ ਡਾਊਨਲੋਡ ਸਥਾਨ ਤੱਕ ਪਹੁੰਚ ਕਰ ਸਕਦੇ ਹਾਂ। Ctrl+J ਦਬਾਉਣ ਤੋਂ ਬਾਅਦ ਹੇਠ ਦਿੱਤੀ ਸਕ੍ਰੀਨ ਦਿਖਾਈ ਦੇਵੇਗੀ।



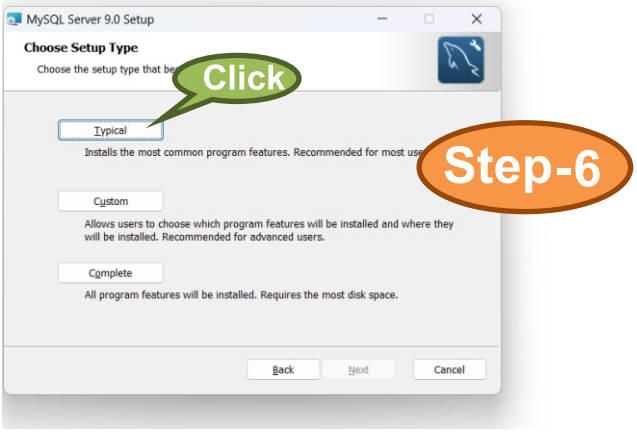
ਹੁਣ ਸਾਡੇ ਕੋਲ ਡਾਊਨਲੋਡ ਹਿਸਟਰੀ ਵਿੱਚ "mysql-9.0.0 xxxxxxxxxxx.msi" ਫਾਈਲ ਡਾਊਨਲੋਡ ਹੋ ਜਾਵੇਗੀ। ਡਾਉਨਲੋਡ ਹੋਈ ਫਾਈਲ ਦੇ ਨਾਮ ਤੇ ਕਲਿੱਕ ਕਰੋ। MySQL ਸਰਵਰ ਦੀ ਇੰਸਟਾਲੇਸ਼ਨ ਸ਼ੁਰੂ ਹੋ ਜਾਵੇਗੀ ਅਤੇ ਹੇਠ ਦਿੱਤੀ ਸਕ੍ਰੀਨ ਦਿਖਾਈ ਦੇਵੇਗੀ।

ਇਹ MySQL Server 9.0 ਸੈਟਅਪ ਵਿਜ਼ਾਰਡ (ਕਿਸੇ ਐਕਸ਼ਨ/ਟਾਸਕ ਨੂੰ ਪੂਰਾ ਕਰਨ ਦੀ ਕਦਮ ਦਰ ਕਦਮ ਪੇਸ਼ਕਸ਼) ਦੀ Welcome Screen ਹੈ। ਇਸ ਸਕਰੀਨ 'ਤੇ ਮੌਜੂਦ Next ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।



ਇਸ ਤੋਂ ਅਗਲੀ ਸਕਰੀਨ ਦਿਖਾਉਂਦੀ ਹੈ ਕਿ ਸਾਡੀ ਇੰਸਟਾਲੇਸ਼ਨ ਸ਼ੁਰੂ ਹੋਣ ਲਈ ਤਿਆਰ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਆਪਣੇ MySQL ਸਰਵਰ ਲਈ ਕਿਸੇ ਹੋਰ ਕਿਸਮ ਦਾ ਸੈੱਟਅੱਪ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ "Back" ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ। ਜੇਕਰ ਅਸੀਂ "Typical" ਸੈੱਟਅੱਪ ਕਿਸਮ ਦੇ ਨਾਲ ਜਾਰੀ ਰੱਖਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ "Install" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ।

ਇਹ ਅਗਲੀ ਸਕ੍ਰੀਨ ਸੈੱਟਅੱਪ ਦੀ ਕਿਸਮ ਦੀ ਚੋਣ ਨਾਲ ਸਬੰਧਤ ਹੈ। ਇਸ ਵਿੱਚ ਕਈ ਵਿਕਲਪ ਹਨ ਜਿਵੇਂ ਕਿ Typical, Custom ਜਾਂ Complete। ਅਸੀਂ ਇਸ ਸਕ੍ਰੀਨ ਤੋਂ "Typical" ਸੈੱਟਅੱਪ ਕਿਸਮ ਦੀ ਚੋਣ ਕਰ ਸਕਦੇ ਹਾਂ ਕਿਉਂਕਿ ਇਸ ਵਿੱਚ MySQL ਸਰਵਰ ਨੂੰ ਚਲਾਉਣ ਲਈ ਸਾਰੇ ਲੋੜੀਂਦੇ ਟੂਲ ਅਤੇ ਪੈਕੇਜ਼ ਸ਼ਾਮਲ ਹਨ।



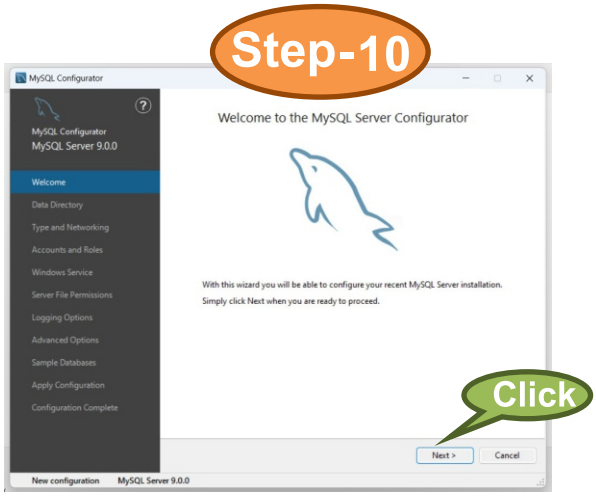
ਜਦੋਂ ਅਸੀਂ ਵਿਜ਼ਾਰਡ ਵਿੱਚ ਦਿਖਾਏ ਗਏ "Typical" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਅਗਲੀ ਸਕ੍ਰੀਨ ਦਿਖਾਈ ਦਿੰਦੀ ਹੈ।

ਇਸ ਤੋਂ ਅਗਲੀ ਸਕਰੀਨ ਦਿਖਾਉਂਦੀ ਹੈ ਕਿ ਸਾਡੀ ਇੰਸਟਾਲੇਸ਼ਨ ਸ਼ੁਰੂ ਹੋਣ ਲਈ ਤਿਆਰ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਆਪਣੇ MySQL ਸਰਵਰ ਲਈ ਕਿਸੇ ਹੋਰ ਕਿਸਮ ਦਾ ਸੈੱਟਅੱਪ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ "Back" ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ। ਜੇਕਰ ਅਸੀਂ "Typical" ਸੈੱਟਅੱਪ ਕਿਸਮ ਦੇ ਨਾਲ ਜਾਰੀ ਰੱਖਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ "Install" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ।

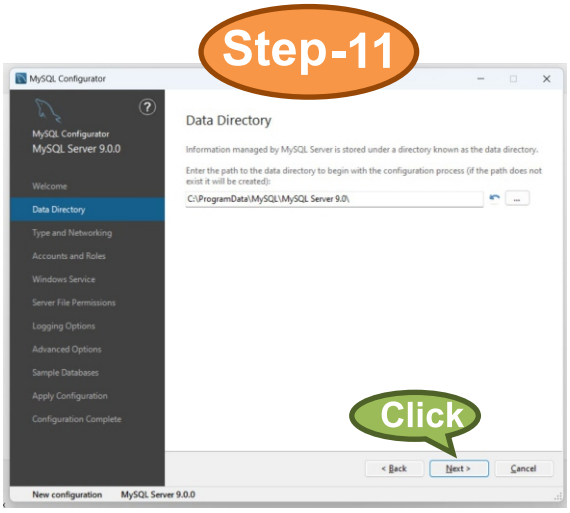
Install ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰਨ ਤੋਂ ਬਾਅਦ ਇੰਸਟਾਲੇਸ਼ਨ ਦੀ ਪ੍ਰਕਿਰਿਆ ਸ਼ੁਰੂ ਹੋ ਜਾਵੇਗੀ ਅਤੇ ਇਸ ਪ੍ਰਕਿਰਿਆ ਨੂੰ ਪ੍ਰੋਗਰੈੱਸ ਬਾਰ ਰਾਹੀਂ ਦਰਸ਼ਾਇਆ ਜਾਵੇਗਾ। ਇੰਸਟਾਲੇਸ਼ਨ ਦੇ ਪੂਰਾ ਹੋਣ ਤੋਂ ਬਾਅਦ ਮੁਕੰਮਲ ਹੋਣ ਦਾ ਸੁਨੇਹਾ ਦਿੱਤਾ ਜਾਵੇਗਾ।

ਹੁਣ MySQL Server ਸਫਲਤਾਪੂਰਵਕ ਇੰਸਟਾਲ ਹੋ ਗਿਆ ਹੈ ਪਰੰਤੂ ਇੱਕ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਕਦਮ ਅਜੇ ਬਾਕੀ ਹੈ।

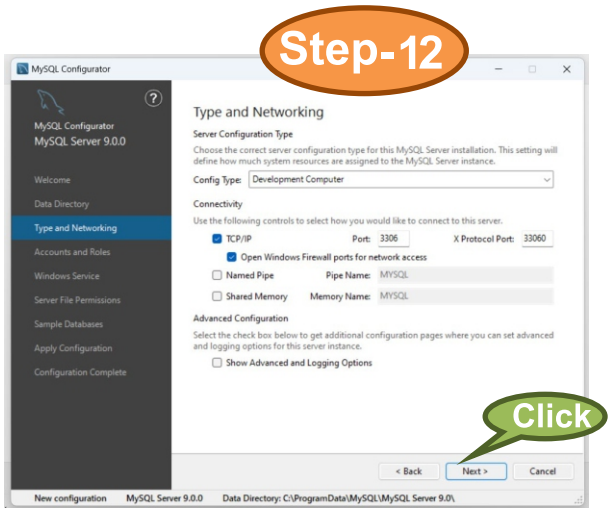
ਅਸੀਂ “Run MySQL Configurator” ਵਿਕਲਪ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਆਪਣੇ MySQl Server ਨੂੰ ਕਨਫੀਗਰ (configure) ਕਰਨਾ ਹੋਵੇਗਾ। ਇਹ ਵਿਕਲਪ ਮੂਲ ਰੂਪ ਵਿੱਚ ਸਲੈਕਟ ਹੋਇਆ ਹੁੰਦਾ ਹੈ। ਸਾਨੂੰ MySQL Server ਦੀ ਵਰਤੋਂ ਤੋਂ ਪਹਿਲਾਂ ਇਸਨੂੰ ਕਨਫੀਗਰ ਕਰਨਾ ਲਾਜ਼ਮੀ ਹੁੰਦਾ ਹੈ। ਅਸੀਂ ਇਸ ਸਕ੍ਰੀਨ ਤੋਂ “Finish” ਬਟਨ ’ਤੇ ਕਲਿੱਕ ਕਰਾਂਗੇ।



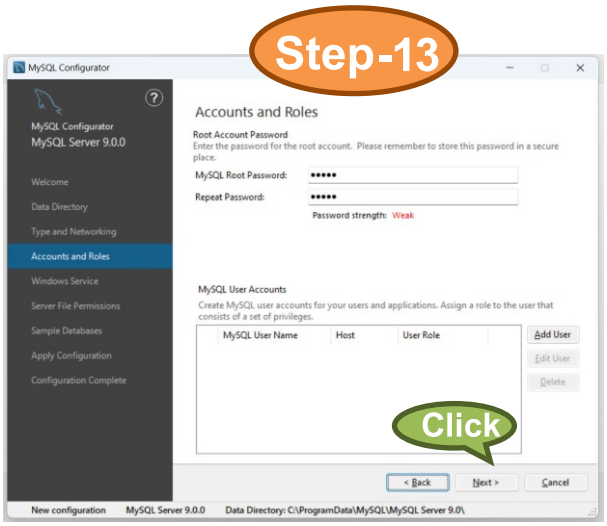
ਇਸ ਤੋਂ ਬਾਅਦ MySQL ਸਰਵਰ ਕਨਫੀਗਰੇਟਰ ਟੂਲ ਸ਼ੁਰੂ ਹੋ ਜਾਵੇਗਾ। ਡਿਸਪਲੇ ਸਕਰੀਨ ਇਸਦੀ ਇੱਕ Welcome ਸਕਰੀਨ ਹੈ। ਇਸ ਸਕਰੀਨ ਤੋਂ “Next” ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।



ਇਹ ਸਕ੍ਰੀਨ ਸਾਡੇ MySQL Server ਲਈ ਡਾਟਾ ਡਾਇਰੈਕਟਰੀ ਦੀ ਚੋਣ ਨਾਲ ਸੰਬੰਧਿਤ ਹੈ। ਡਿਫਾਲਟ ਸਟੋਰੇਜ਼ ਡਾਇਰੈਕਟੀ ਪਹਿਲਾਂ ਹੀ ਡਿਸਪਲੇ ਕੀਤੀ ਜਾਵੇਗੀ। ਅਸੀਂ ਲੋੜ ਪੈਣ ਤੇ ਇਸਨੂੰ ਬਦਲ ਸਕਦੇ ਹਾਂ ਜਾਂ ਡਿਫਾਲਟ ਡਾਇਰੈਕਟੀ ਰੱਖ ਸਕਦੇ ਹਾਂ ਅਤੇ “Next” ਬਟਨ ਦਬਾ ਸਕਦੇ ਹਾਂ।



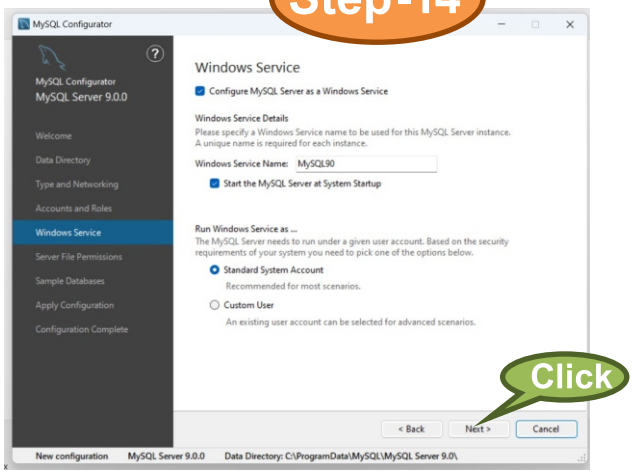
ਇਹ ਅਗਲੀ ਸਕ੍ਰੀਨ ਸਾਡੇ MySQL Server ਲਈ ਨੈੱਟਵਰਕ ਸੈੱਟਅੱਪ ਨਾਲ ਸਬੰਧਤ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਨੈੱਟਵਰਕ ਦੀ ਸੈਟਿੰਗ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ ਲੋੜੀਂਦੀਆਂ ਤਬਦੀਲੀਆਂ ਕਰ ਸਕਦੇ ਹਾਂ ਨਹੀਂ ਤਾਂ ਡਿਫਾਲਟ ਸੈਟਿੰਗਾਂ ਅਨੁਸਾਰ ਹੀ “Next” ਬਟਨ ਦਬਾਓ।



ਇਹ ਸਕ੍ਰਿਨ ਇੰਸਟਾਲੇਸ਼ਨ ਅਤੇ ਕਨਫੀਗਰੇਸ਼ਨ ਦਾ ਸਭ ਤੋਂ ਮਹੱਤਵਪੂਰਨ ਹਿੱਸਾ ਹੈ।

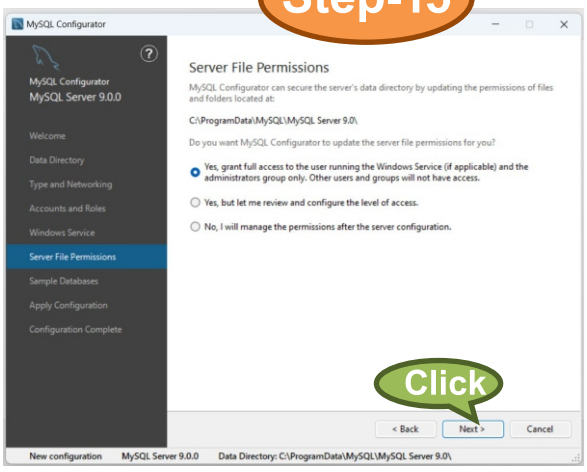
ਇਸ ਕਦਮ ਤੇ ਅਸੀਂ ਆਪਣੇ MySQL ਲਈ ਰੂਟ (Root) ਪਾਸਵਰਡ ਸੈੱਟ ਕਰਾਂਗੇ। ਇਸ ਪਾਸਵਰਡ ਦੀ ਹਰ ਵਾਰ ਲੋੜ ਪਵੇਗੀ ਜਦੋਂ ਅਸੀਂ ਆਪਣਾ MySQL ਕਲਾਇੰਟ ਖੋਲ੍ਹਦੇ ਹਾਂ। ਸਾਨੂੰ "MySQL Root Password" ਅਤੇ "Repeat Password" ਨਾਮਕ ਦੋਨਾਂ ਟੈਕਸਟ ਬਾਕਸਾਂ ਵਿੱਚ ਕੋਈ ਵੀ ਪਾਸਵਰਡ ਦਰਜ ਕਰਨਾ ਹੋਵੇਗਾ।
ਦੋਵੇਂ ਟੈਕਸਟ ਬਾਕਸਾਂ ਵਿੱਚ ਇੱਕੋ ਪਾਸਵਰਡ ਦਰਜ ਕਰਨ ਤੋਂ ਬਾਅਦ ਅਸੀਂ "Next" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ।

Step-14



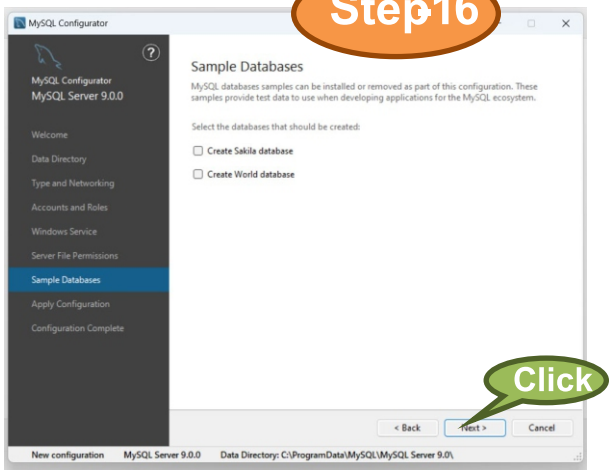
ਇੰਸਟਾਲੇਸ਼ਨ ਦੀ ਅਗਲੀ ਸਕ੍ਰੀਨ MySQL Server ਨੂੰ ਚਲਾਉਣ ਲਈ ਲੋੜੀਂਦੀ ਵਿੰਡੋਜ਼ ਸਰਵਿਸਜ਼ ਕਨਫਿਗਰੇਸ਼ਨ ਨਾਲ ਸੰਬੰਧਿਤ ਹੈ। ਅਸੀਂ ਡਿਪਾਲਟ ਸੈਟਿੰਗ ਰੱਖ ਸਕਦੇ ਹਾਂ ਅਤੇ ਇਸ ਸਕ੍ਰੀਨ ਤੋਂ "Next" ਬਟਨ ਦਬਾ ਸਕਦੇ ਹਾਂ।

Step-15



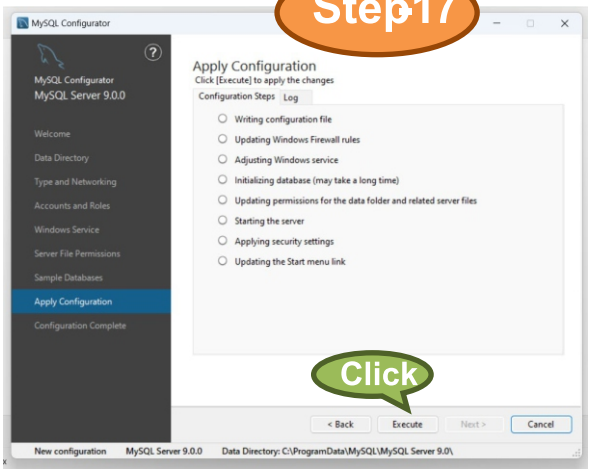
ਹੁਣ ਸਰਵਰ ਫਾਈਲ ਪਰਮਿਸ਼ਨ ਮੋਡੀਊਲ ਦਿਖਾਇਆ ਜਾਵੇਗਾ। ਸਾਨੂੰ ਸਾਡੀਆਂ MySQL ਫਾਈਲਾਂ ਤੱਕ ਪਹੁੰਚ ਕਰਨ ਲਈ ਅਨੁਮਤੀ ਦੇਣੀ ਪਵੇਗੀ। ਲੋੜੀਂਦੀਆਂ ਫਾਈਲਾਂ ਤੱਕ ਪੂਰੀ ਪਹੁੰਚ ਦੇਣ ਲਈ ਪਹਿਲਾ ਵਿਕਲਪ ਚੁਣੋ ਅਤੇ "Next" ਬਟਨ ਦਬਾਓ।

Step16

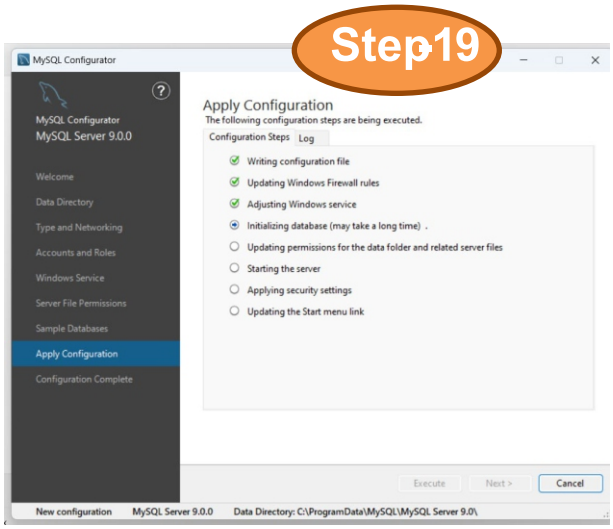


MySQL ਕਨਫਿਗਰੇਟਰ ਦਾ ਅਗਲਾ ਕਦਮ ਨਮੂਨਾ ਡੇਟਾਬੇਸ ਬਣਾਉਣ ਨਾਲ ਸਬੰਧਤ ਹੈ ਜੋ ਪਹਿਲਾਂ ਹੀ MySQL ਕਨਫਿਗਰੇਟਰ ਵਿੱਚ ਨਿਰਧਾਰਤ ਕੀਤੇ ਗਏ ਹਨ। ਅਸੀਂ ਕਿਸੇ ਵੀ ਨਮੂਨਾ ਡੇਟਾਬੇਸ ਦੀ ਰਚਨਾ ਨੂੰ ਛੱਡ ਸਕਦੇ ਹਾਂ ਅਤੇ "Next" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ।

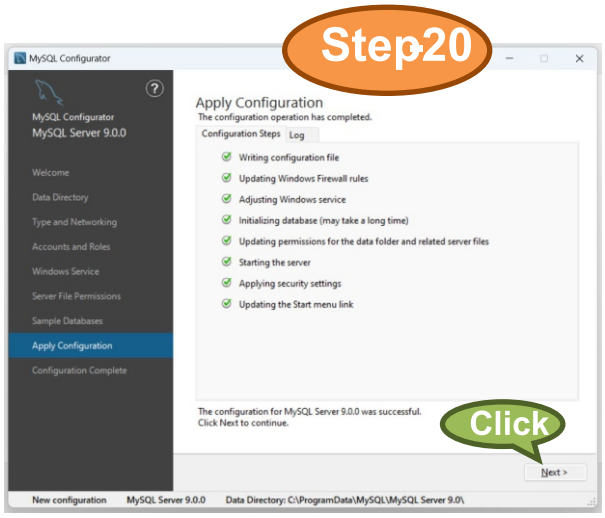
Step17



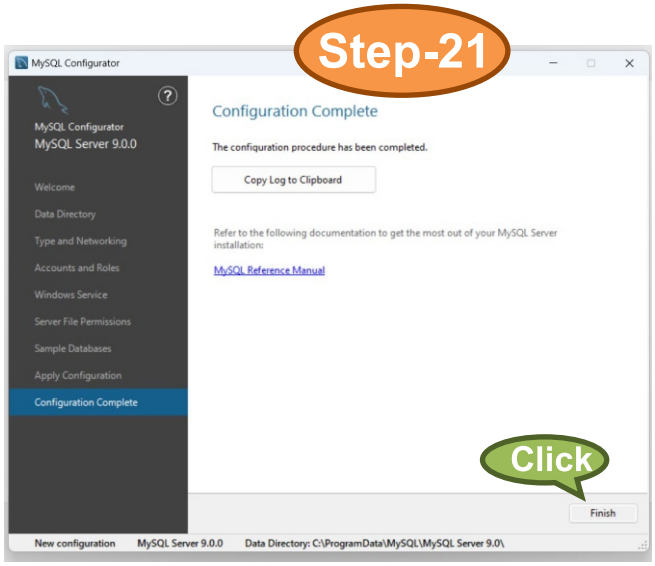
ਹੁਣ ਪਿਛਲੇ ਪੜਾਵਾਂ ਦੌਰਾਨ ਸਾਰੇ ਚੁਣੇ ਗਏ ਇੰਸਟਾਲੇਸ਼ਨ ਵਿਕਲਪਾਂ ਨੂੰ ਲਾਗੂ ਕਰਨ ਦਾ ਸਮਾਂ ਹੈ। ਜੇਕਰ ਅਸੀਂ ਸਾਰੇ ਚੁਣੇ ਹੋਏ ਵਿਕਲਪਾਂ ਨਾਲ ਅੱਗੇ ਵਧਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ "Execute" ਬਟਨ 'ਕੇ ਕਲਿੱਕ ਕਰੋ।



ਹੁਣ ਇੰਸਟਾਲੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ ਸ਼ੁਰੂ ਹੋ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸਨੂੰ ਪੂਰਾ ਹੋਣ ਵਿੱਚ ਕੁਝ ਸਮਾਂ ਲੱਗ ਸਕਦਾ ਹੈ। ਥੋੜੀ ਦੇਰ ਇੰਤਜ਼ਾਰ ਕਰੋ ਅਤੇ ਸਾਰੀਆਂ ਪ੍ਰਕਿਰਿਆਵਾਂ ਦੇ ਪੂਰਾ ਹੋਣ ਦੀ ਉਡੀਕ ਕਰੋ। ਇਹ ਹਰੇਕ ਵਿਕਲਪ ਦੇ ਵਿਰੁੱਧ ਟਿੱਕ ਲਗਾ ਕੇ ਇਸ ਪ੍ਰਕਿਰਿਆਂ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇਗਾ।



ਜਦੋਂ ਸਾਰੀ ਇੰਸਟਾਲੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ ਪੂਰੀ ਹੋ ਜਾਂਦੀ ਹੈ, ਅਸੀਂ ਇਸ ਇੰਸਟਾਲੇਸ਼ਨ ਪ੍ਰਕਿਰਿਆ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ "Next" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰ ਸਕਦੇ ਹਾਂ।

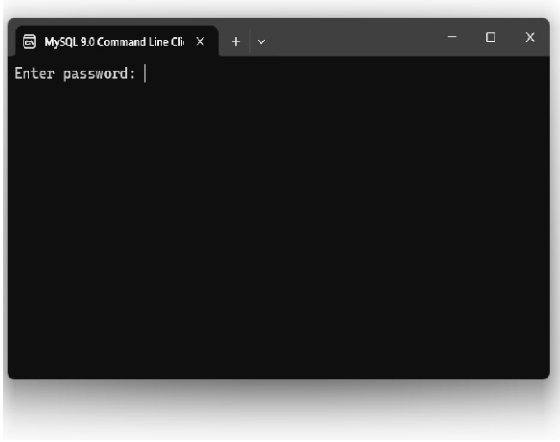


ਇਸ ਅੰਤਮ ਪ੍ਰਕਿਰਿਆ ਤੋਂ ਬਾਅਦ ਸਕਰੀਨ ਤੇ ਇੰਸਟਾਲੇਸ਼ਨ ਪੂਰਾ ਹੋ ਜਾਣ ਦਾ ਸੁਨੇਹਾ ਦਿਖਾਇਆ ਜਾਵੇਗਾ। ਵਿਜ਼ਾਰਡ ਨੂੰ ਬੰਦ ਕਰਨ ਲਈ "Finish" ਬਟਨ 'ਤੇ ਕਲਿੱਕ ਕਰੋ।

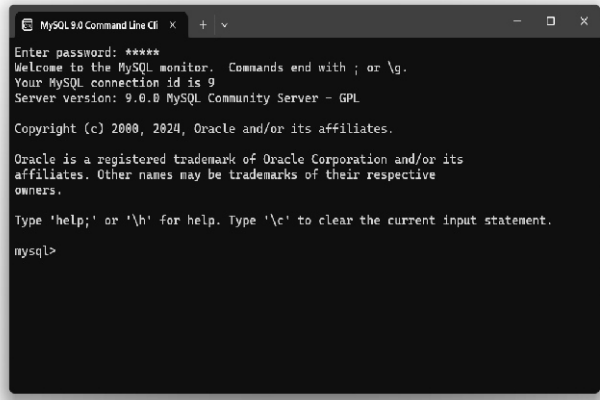
ਹੁਣ MySQL ਦੀ ਇੰਸਟਾਲੇਸ਼ਨ ਅਤੇ ਕਨਫੀਗਰੇਸ਼ਨ ਪੂਰੀ ਹੋ ਗਈ ਹੈ। ਅਸੀਂ ਇਸ ਪ੍ਰਕਿਰਿਆ ਨੂੰ ਪੂਰਾ ਕਰਨ ਲਈ ਇੱਕ ਵਾਰ ਸਿਸਟਮ ਨੂੰ Restart ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਹੇਠਾਂ ਦਿੱਤੇ ਕਦਮਾਂ ਦੀ ਵਰਤੋਂ ਕਰਕੇ MySQL Server ਦੀ ਸਹੀ ਤਰੀਕੇ ਨਾਲ ਵਰਤੋਂ ਦੀ ਜਾਂਚ ਕਰ ਸਕਦੇ ਹਾਂ। ਜਿਸਦੇ ਕਦਮ ਹੇਠਾਂ ਦਿਤੇ ਗਏ ਹਨ।

Start-> MySQL 9.0 Command Line Client-> ਪ੍ਰਗਟ ਹੋਏ ਪ੍ਰੋਗਰਾਮ ਤੇ ਕਲਿੱਕ ਕਰੋ।

ਇਸ ਉਪਰੰਤ ਹੇਠਾਂ ਦਿਤੀਆਂ ਸਕ੍ਰਿਨਾਂ ਨਜ਼ਰ ਆਉਣਗੀਆਂ



ਇਹ ਸਕ੍ਰੀਨ ਪਾਸਵਰਡ ਦੀ ਮੰਗ ਕਰ ਰਹੀ ਹੈ। ਇਹ ਪਾਸਵਰਡ ਉਹੀ ਪਾਸਵਰਡ ਹੈ ਜੋ ਅਸੀਂ MySQL Configuration ਦੌਰਾਨ ਰੂਟ ਪਾਸਵਰਡ ਸੈੱਟ ਕੀਤਾ ਹੈ। ਪਾਸਵਰਡ ਦਰਜ ਕਰਨ ਤੋਂ ਬਾਅਦ MySQl prompt ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਜਾਵੇਗਾ ਅਤੇ ਸਾਡਾ MySQL ਕਲਾਇੰਟ ਦਰਸਾਏ ਅਨੁਸਾਰ ਵਰਤਣ ਲਈ ਤਿਆਰ ਹੋਵੇਗਾ।



ਉਪਰੋਕਤ ਸਕਰੀਨ ਦਿਖਾਉਂਦੀ ਹੈ ਕਿ MySQL ਸਰਵਰ ਹੁਣ ਇੰਸਟਾਲ ਹੈ ਅਤੇ ਵਰਤਣ ਲਈ ਤਿਆਰ ਹੈ।

ਹੁਣ ਇੰਸਟਾਲੇਸ਼ਨ ਪੂਰੀ ਹੋ ਗਈ ਹੈ ਅਤੇ ਅਸੀਂ MySQL ਸਰਵਰ ਨਾਲ ਸਬੰਧਤ ਸਾਰੀਆਂ ਪ੍ਰੈਕਟੀਕਲ ਗਤੀਵਿਧੀਆਂ ਦਾ ਅਭਿਆਸ ਕਰਨ ਲਈ ਤਿਆਰ ਹਾਂ।