Downloaded from https:// www.studiestoday.com

ਕੰਪਿਊਟਰ ਸਾਇੰਸ

(ਗਿਆਰ੍ਹਵੀਂ ਸ਼੍ਰੇਣੀ ਲਈ ਪਾਠ-ਪੁਸਤਕ)

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

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ

ਸਾਹਿਬਜ਼ਾਦਾ ਅਜੀਤ ਸਿੰਘ ਨਗਰ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

© ਪੰਜਾਬ ਸਰਕਾਰ

ਐਡੀਸ਼ਨ 2016......................... 1,59,000 ਕਾਪੀਆਂ

All rights, including those of translation, reproduction
and annotation etc., are reserved by the
Punjab Government

ਚੇਤਾਵਨੀ

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

(ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਦੀਆਂ ਪਾਠ-ਪੁਸਤਕਾਂ ਬੋਰਡ ਦੇ 'ਵਾਟਰ ਮਾਰਕ' ਵਾਲੇ ਕਾਗਜ਼ 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. ਕਨਵਿਨੀਏਂਟ ਪ੍ਰਿੰਟਰਜ਼, ਜਲੰਧਰ ਦੁਆਰਾ ਛਾਪੀ ਗਈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮੁੱਖ-ਬੰਧ

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਆਪਣੀ ਸਥਾਪਨਾ ਦੇ ਸਮੇਂ ਤੋਂ ਹੀ ਸਕੂਲ ਪੱਧਰ ਦੇ ਪਾਠ-ਕ੍ਰਮਾਂ ਨੂੰ ਆਧੁਨਿਕ ਸੋਚ ਅਤੇ ਖੋਜ ਅਨੁਸਾਰ ਢਾਲਣ ਅਤੇ ਇਹਨਾਂ ਪਾਠ-ਕ੍ਰਮਾਂ ਅਨੁਸਾਰ ਪਾਠ-ਪੁਸਤਕਾਂ ਤਿਆਰ ਕਰਨ ਲਈ ਯਤਨਸ਼ੀਲ ਰਿਹਾ ਹੈ। ਰਾਸ਼ਟਰੀ ਪੱਧਰ ਤੇ ਵਰਤਮਾਨ ਸੋਚ ਨੂੰ ਮੁੱਖ ਰਖਦੇ ਹੋਏ ਬੋਰਡ ਨੇ ਪਾਠ-ਪੁਸਤਕਾਂ ਦੀ ਨਵ-ਰਚਨਾ ਦਾ ਵਿਸ਼ੇਸ਼ ਪ੍ਰੋਗਰਾਮ ਉਲੀਕਿਆ ਹੋਇਆ ਹੈ। ਇਹ ਪਾਠ-ਪੁਸਤਕ ਇਸੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਇੱਕ ਕੜੀ ਹੈ।

ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਵਿਸ਼ੇ ਦੇ ਅਧਿਐਨ ਦੀ ਲੋੜ ਦਿਨੋ-ਦਿਨ ਵੱਧਦੀ ਜਾ ਰਹੀ ਹੈ ਕਿਉਂਕਿ ਵਿਗਿਆਨ ਅਤੇ ਤਕਨੀਕੀ ਉੱਨਤੀ ਦੇ ਯੁਗ ਵਿੱਚ ਹਰ ਕਾਰਜ ਖੇਤਰ ਵਿੱਚ ਕਾਰਜ ਕੁਸ਼ਲਤਾ ਵਧਾਉਣ ਲਈ ਕੰਪਿਊਟਰ ਸਿੱਖਿਆ ਦੀ ਲੋੜ ਹੈ। ਸੂਚਨਾ ਤਕਨਾਲੌਜੀ ਦੇ ਪ੍ਰਸਾਰ ਅਤੇ ਸੰਚਾਰ ਦੀ ਤਰੱਕੀ ਨਾਲ, ਹਰ ਵਿਭਾਗ ਵਿੱਚ ਕੰਪਿਊਟਰੀਕਰਨ ਕੀਤਾ ਜਾ ਰਿਹਾ ਹੈ। ਵੱਖ-ਵੱਖ ਮਹਿਕਮਿਆਂ ਤੋਂ ਲੋੜੀਂਦੀ ਸੂਚਨਾ ਪ੍ਰਾਪਤ ਕਰਨ, E-Ticketing ਅਤੇ ਇੰਟਰਨੈੱਟ ਦੀ ਵਰਤੋਂ ਦੀ ਸੁਵਿਧਾ ਲੈਣ ਲਈ ਕੰਪਿਊਟਰ ਸਿੱਖਿਆ ਹਰ ਵਿਅਕਤੀ ਲਈ ਲਾਜ਼ਮੀ ਹੋ ਗਈ ਹੈ।

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ ਨੇ ਅਜਿਹੀਆਂ ਜ਼ਰੂਰਤਾਂ ਨੂੰ ਮੁੱਖ ਰੱਖਦੇ ਹੋਏ, ਪੰਜਾਬ ਸਰਕਾਰ ਦੇ ਦਿਸ਼ਾ-ਨਿਰਦੇਸ਼ ਅਨੁਸਾਰ ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਵਿਸ਼ਾ ਐਲੀਮੈਂਟਰੀ ਅਤੇ ਸੈਕੰਡਰੀ ਪੱਧਰ ਤੇ ਲਾਜ਼ਮੀ ਵਿਸ਼ੇ ਵਜੋਂ ਲਾਗੂ ਕੀਤਾ ਹੈ। ਇਸ ਵਿਸ਼ੇ ਦਾ ਪਾਠਕ੍ਰਮ ਕੁਝ ਸਰਕਾਰੀ ਸਕੂਲਾਂ ਵਿੱਚ ਪਿਕਟਸ ਦੁਆਰਾ ਪਹਿਲਾਂ ਹੀ ਪੜ੍ਹਾਇਆ ਜਾ ਰਿਹਾ ਸੀ। ਅਧਿਆਪਕਾਂ ਦੀ ਮੰਗ ਅਨੁਸਾਰ ਪਾਠਕ੍ਰਮ ਵਿੱਚ ਲੋੜੀਂਦੀ ਸੋਧ ਕਰਕੇ ਪੁਸਤਕ ਪੰਜਾਬੀ ਮਾਧਿਅਮ ਵਿੱਚ ਸੋਧੇ ਹੋਏ ਰੂਪ ਵਿੱਚ ਛਪਾਈ ਗਈ ਹੈ। ਇਸ ਵਿਸ਼ੇ ਸੰਬੰਧੀ ਸਾਰੀ ਜਾਣਕਾਰੀ ਇਸ ਪੁਸਤਕ ਵਿੱਚ ਉਪਲੱਬਧ ਕਰਵਾਉਣ ਦੀ ਕੋਸ਼ਿਸ਼ ਕੀਤੀ ਗਈ ਹੈ। ਆਸ ਹੈ ਇਹ ਪੁਸਤਕ ਵਿਸ਼ੇ ਦੇ ਵਿਦਿਆਰਥੀਆਂ ਅਤੇ ਅਧਿਆਪਕਾਂ ਲਈ ਉਪਯੋਗੀ ਸਿੱਧ ਹੋਵੇਗੀ। ਪੁਸਤਕ ਨੂੰ ਚੰਗੇਰਾ ਬਣਾਉਣ ਲਈ ਖੇਤਰ ਵਿੱਚੋਂ ਆਏ ਸੁਝਾਵਾਂ ਦਾ ਸਤਿਕਾਰ ਕੀਤਾ ਜਾਵੇਗਾ

ਚੇਅਰਪਰਸਨ

ਪੰਜਾਬ ਸਕੂਲ ਸਿੱਖਿਆ ਬੋਰਡ

"ਭਲਾਈ ਵਿਭਾਗ, ਪੰਜਾਬ"

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਵਿਸ਼ਾ ਸੂਚੀ

ਲੜੀ ਨੰ : ਅਧਿਆਇ ਪੰਨਾ

1. ਦਸਵੀਂ ਕਲਾਸ ਦੀ ਦੁਹਰਾਈ 1-16
   - 1.0 ਸਾਫਟਵੇਅਰ ਸੰਕਲਪ
   - 1.1 ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ :ਔਪਰੇਟਿੰਗ ਸਿਸਟਮ, ਯੁਟਿਲਿਟੀ ਸਾਫਟਵੇਅਰ, ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ
   - 1.2 ਐਕਸੈਲ : ਡਾਟਾ ਟਾਈਪ ਫਾਰਮੂਲਾ ਅਤੇ ਫੰਕਸ਼ਨਜ਼, ਐਕਸਲ ਅਤੇ ਫਾਈਨੈਂਨਸ਼ਿਯਲ ਡਾਟਾ
   - 1.3 ਐਚ.ਟੀ.ਐਮ.ਐਲ. ਦੀ ਦੁਹਰਾਈ (Review on HTML): ਵੈੱਬ ਪੇਜ਼ਿਸ, ਐਚ.ਟੀ.ਐਮ.ਐਲ. (HTML) ਫਾਈਲ
   - 1.4 ਮਾਇਕਰੋਸਾਫਟ ਅਸੈਸ : ਡਾਟਾ ਸੋਧਨਾ, ਅਸੈਸ ਡਾਟਾਬੇਸ ਦੇ ਆਬਜੈਕਟਸ
   - 1.5 ਪ੍ਰੋਗਰਾਮਿੰਗ ਕੰਨਸੈਪਟਸ ਦੀ ਦੁਹਰਾਈ (Review on programming concepts): ਪ੍ਰੋਗਰਾਮ ਡਿਵੈਲਪਮੈਂਟ ਦੇ ਵੱਖ-ਵੱਖ ਪੜਾਅ, ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੇ ਐਲੀਮੈਂਟ
2. "ਸੀ" ਭਾਸ਼ਾ ਵਿਚ ਪ੍ਰੋਗਰਾਮਿੰਗ ਲਈ ਭੂਮਿਕਾ 17-34
   - 2.0 ਭੂਮਿਕਾ
   - 2.1 ਸੀ (C) ਭਾਸ਼ਾ ਦੇ ਵਿਸ਼ੇਸ਼ ਲੱਛਣ
   - 2.2 ਸੀ (C) ਚਿੰਨ੍ਹ ਸੈੱਟ : ਐਸਕੇਪ ਚਿੰਨ੍ਹ, ਵਾਈਟ ਸਪੇਸ ਕਰੈਕਟਰ
   - 2.3 ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ ਸਟਰਕਚਰ : ਹੈਡਰ ਫਾਈਲਜ਼, ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਸਟੇਟਮੈਂਟ/ਨਿਰਦੇਸ਼, ਗਲੋਬਲ ਡਿਕਲੇਰੇਸ਼ਨਜ਼, ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ ਕੰਪਾਈਲ ਅਤੇ ਲਾਗੂ ਕਰਨ
   - 2.4 ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ
   - 2.5 ਫੰਕਸ਼ਨ : ਬਿਲਟ ਇਨ ਫੰਕਸ਼ਨਸ, ਯੂਜ਼ਰ ਪਰਿਭਾਸ਼ਤ ਫੰਕਸ਼ਨਸ
   - 2.6 ਫਾਰਮੇਟਡ ਆਈ/ਓ ਫੰਕਸ਼ਨ : ਪ੍ਰਿੰਟਐਫ ਫੰਕਸ਼ਨ (printf function), ਸਕੈਨਐਫ ਫੰਕਸ਼ਨ (scanf function)
   - 2.7 ਸੀ (C) ਪ੍ਰੋਗਰਾਮਿੰਗ ਨਾਲ ਸ਼ੁਰੂਆਤ ਕਰਨੀ : ਟਰਬੋ ਸੀ ਨੂੰ ਸਥਾਪਿਤ ਕਰਨਾ, ਪ੍ਰੋਗਰਾਮ ਦੀ ਕੰਪਾਇਲਿੰਗ ਅਤੇ ਐਗਜ਼ਿਕਿਊਟਿੰਗ
3. ਕਾਂਸਟੈਂਟਸ, ਵੇਰੀਏਬਲਜ਼ ਅਤੇ ਡਾਟਾ ਟਾਈਪਸ 35-46
   - 3.0 ਭੂਮਿਕਾ
   - 3.1 ਕਾਂਸਟੈਂਟਸ/ਸ਼ਾਬਦਿਕ : ਸੀ (C) ਕਾਂਸਟੈਂਟ ਦੀਆਂ ਟਾਈਪਸ
   - 3.2 ਸੀ (C)ਵੇਰੀਏਬਲਜ/ਆਈਡੈਂਟੀਫਾਈਰ ਦੀਆਂ ਟਾਈਪਸ : ਡਿਲੀਮਿਟਰ, ਵੇਰੀਏਬਲਜ਼ ਦਾ ਡਿਕਲੇਰੇਸ਼ਨ, ਵੇਰੀਏਬਲ ਵਿਚ ਕਾਂਸਟੈਂਟ ਸਟੋਰ ਕਰਨਾ
   - 3.3 ਡਾਟਾ ਟਾਈਪਸ : ਬਿਲਟ ਇਨ ਡਾਟਾ ਟਾਈਪਸ, ਮੇਨ ਫੰਕਸ਼ਨ ਹੈਡਰ
   - 3.4 ਟੋਕਨਜ਼ (ਆਈਡੈਂਟਿਫਾਇਰਜ਼, ਕੀ-ਵਰਡਜ਼, ਕਾਂਸਟੈਂਟ, ਓਪਰੇਟਰਸ) : ਕੀ-ਵਰਡਜ਼ ਅਤੇ ਆਈਡੈਂਟੀਫਾਇਰਜ਼, ਟਾਈਪ ਮੋਡੀਫਾਈਰ ਜਾਂ ਕੁਆਲੀਫਾਈਰ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

4. ਓਪਰੇਟਰਸ ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ 47-58
4.0 ਭੂਮਿਕਾ
4.1 ਓਪਰੇਟਰਸ ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ : ਐਕਸਪ੍ਰੇਸ਼ਨ, ਬਾਇਨਰੀ ਓਪਰੇਟਰ, ਓਪਰੇਸ਼ਨਜ਼ ਅਤੇ ਹਿਰੈਚੀਕਲ ਆਰਡਰ (Operations & Hierarchical order)
4.2 ਰਿਲੇਸ਼ਨਲ ਅਤੇ ਲੋਜੀਕਲ ਓਪਰੇਟਰ: ਲੋਜੀਕਲ ਓਪਰੇਟਰ, ਅਸਾਈਨਮੇਂਟ ਓਪਰੇਟਰ, ਇਨਕਰੀਮੈਂਟ ਐਂਡ ਡਿਕਰੀਮੈਂਟ ਓਪਰੇਟਰਸ, ਟਰਨਰੀ ਓਪਰੇਟਰ, ਕੌਮਾ ਓਪਰੇਟਰ, ਸਾਈਜ਼ਆਫ਼ ਓਪਰੇਟਰ, ਬਿਟਵਾਈਜ਼ ਓਪਰੇਟਰ
5. ਕੰਟਰੋਲ ਫਲੋ (ਭਾਗ1) 59-71
5.0 ਭੂਮਿਕਾ
5.1 ਡਿਸਿਜ਼ਨ ਮੇਕਿੰਗ ਸਟੇਟਮੈਂਟਸ : ਇਫ ਸਟੇਟਮੈਂਟ (if statement), ਇਫ ਐਲਸ (if-else)
5.2 ਸਵਿਚ ਸਟੇਟਮੈਂਟ
5.3 ਬ੍ਰੇਕ ਸਟੇਟਮੈਂਟ
5.4 ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ
6. ਕੰਟਰੋਲ ਫਲੋ (ਭਾਗ2) 72-82
6.0 ਭੂਮਿਕਾ
6.1 ਕੰਟਰੋਲ ਲੂਪ ਸਟਰਕਚਰ : ਵਾਈਲ ਸਟੇਟਮੈਂਟ (While statement), ਡੂ ਵਾਈਲ (do while), ਫਾਰ ਸਟੇਟਮੈਂਟ ਲੂਪ (For Statement loop)
7. ਐਰੇਸ (ਭਾਗ1) 83-99
7.0 ਭੂਮਿਕਾ
7.1 ਐਰੇ ਦੀ ਡਿਕਲੇਅਰਿੰਗ ਅਤੇ ਇਨਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ : ਇਨਸ਼ਿਯਲਾਈਜ਼ਿੰਗ ਐਰੇਸ
7.2 ਕੁਝ ਵਿਸ਼ੇਸ਼ ਨਿਯਮ: ਐਰੇ ਵਿਚ ਡਾਟਾ ਪ੍ਰਵੇਸ਼ ਕਰਨਾ.
7.3 ਐਰੇ ਕਾਪੀ ਕਰਨੀ
7.4 ਐਰੇ ਦੇ ਮੁੱਲਾਂ ਦੀ ਪਹੁੰਚ-ਕਰਨੀ
7.5 ਐਰੇ ਐਲੀਮੈਂਟਸ ਦਾ ਪ੍ਰਬੰਧਨ
8. ਐਰੇਸ (ਭਾਗ-2) 100-112
8.0 ਭੂਮਿਕਾ
8.1 ਦੋ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ : ਦੋ ਡਾਇਮੈਨਸ਼ਨ ਐਰੇ ਦੀ ਡਿਕਲੇਰੇਸ਼ਨ, ਦੋ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ ਦੀ ਬਣਤਰ, ਦੋ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ ਦਾ ਇਨੀਸ਼ਿਯਲਾਈਜੇਸ਼ਨ, ਦੋ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ ਐਲੀਮੈਂਟਸ ਦਾ ਇਨੀਸ਼ਿਯਲਾਈਜੇਸ਼ਨ
8.2 ਮੈਮਰੀ ਵਿਚ ਦੋ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ ਐਲੀਮੈਂਟਸ
8.3 ਮਲਟੀ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇਸ – ਕੈਰ ਟਾਈਪ : ਮਲਟੀ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇਸ ਦੇ ਐਲੀਮੈਂਟਸ ਤੱਕ ਪਹੁੰਚ, ਕੈਰ ਟਾਈਪ ਮਲਟੀ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ ਦਾ ਇਨਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ, ਕੈਰ ਵੱਰਡ ਪ੍ਰੋਸੈਸਿੰਗ ਦੇ ਦੋ ਡਾਇਮੈਨਸ਼ਨਲ ਐਰੇ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

8.4 ਇਕ-ਕੈਰ ਟਾਈਪ ਇਨਪੁੱਟ/ਆਊਟਪੁਟ

8.5 # ਡੀਫਾਈਨ ਡਾਇਰੈਕਟਿਵ (# define directive)

9. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ 113-120

9.0 ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਬਾਰੇ ਜਾਣਕਾਰੀ

9.1 ਡਾਕੂਮੈਂਟਸ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨਾ

9.2 ਪ੍ਰਿੰਟਿੰਗ ਦੇ ਤਰੀਕੇ :ਆਫ਼ਸੈਟ ਪ੍ਰਿੰਟਿੰਗ, ਲੇਜ਼ਰ ਪ੍ਰਿੰਟਿੰਗ

9.3 ਫੌਂਟਸ

9.4 ਫਰੇਮ

9.5 ਪੇਜ ਲੇਆਊਟ

9.6 ਡੈਸਕਟਾਪ ਪਬਲੀਸ਼ਿੰਗ ਅਤੇ ਵਰਡਪ੍ਰੋਸੈਸਰ ਵਿਚ ਅੰਤਰ

9.7 ਡਾਕੂਮੈਂਟ ਪਲੈਨਿੰਗ

9.8 ਮੁੱਖ ਸੂਚਨਾ ਨੂੰ ਖਾਸ ਤੌਰ ਤੇ ਦਿਖਾਇਆ ਜਾਣਾ : ਸਟਾਇਲ, ਮਾਰਜਨ, ਫੁਟਰ, ਫੌਂਟ

ਆਮ ਪ੍ਰੋਗਰਾਮਿੰਗ ਗਲਤੀਆਂ 121-123

ਸ਼ਬਦਾਵਲੀ 124-131

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੰਪਾਦਕੀ ਕਮੇਟੀ

ਤਰੁਨ ਅਗਰਵਾਲ
ਸਰਕਾਰੀ ਕੰਨਿਆ ਮਲਟੀਪਰਪਜ਼ ਸੀ. ਸੈ. ਸਕੂਲ, ਮਾਡਲ ਟਾਊਨ, ਪਟਿਆਲਾ

ਜਗਪ੍ਰੀਤ ਸਿੰਘ ਸਿੱਧੂ
ਸਰਕਾਰੀ ਸੀ. ਸੈ. ਸਕੂਲ ਨੰਦਪੁਰ ਕੇਸ਼ੋ, ਪਟਿਆਲਾ

ਗਗਨਦੀਪ ਸਿੰਘ
ਸਰਕਾਰੀ ਮਾਡਲ ਸੀ. ਸੈ. ਸਕੂਲ ਫੇਜ਼ 3 ਬੀ-1, ਮੁਹਾਲੀ

ਵਿਕਰਮ ਪਾਲ
ਸ਼ਹੀਦ ਲਾਂਸ ਨਾਇਕ ਰਣਜੋਧ ਸਿੰਘ ਸਰਕਾਰੀ ਸੀ. ਸੈ. ਸਕੂਲ, ਤਿਊੜ, ਐਸ. ਏ. ਐਸ. ਨਗਰ, ਮੁਹਾਲੀ

ਸ਼ਿਵਾਨੀ ਨੰਦਾ
ਸਰਕਾਰੀ ਸੀ. ਸੈ. ਸਕੂਲ, ਬਹਾਦਰਗੜ੍ਹ, ਪਟਿਆਲਾ

ਪ੍ਰੀਤੀ ਚੱਮ
ਸਰਕਾਰੀ ਹਾਈ ਸਕੂਲ, ਬੁਲੇਪੁਰ, ਲੁਧਿਆਣਾ

ਹਰਪ੍ਰੀਤ ਸਿੰਘ
ਸਰਕਾਰੀ ਹਾਈ ਸਕੂਲ, ਭਮੱਦੀ, ਲੁਧਿਆਣਾ

ਗੌਰਵ ਸੂਦ
ਸਰਕਾਰੀ ਸੀ. ਸੈ. ਸਕੂਲ, ਮਾਣਕ ਮਾਜਰਾ, ਲੁਧਿਆਣਾ

ਹਰਪ੍ਰੀਤ ਸਿੰਘ
ਸਰਕਾਰੀ ਸੀ. ਸੈ. ਸਕੂਲ, ਸਹੌੜਾ, ਮੋਹਾਲੀ

ਪੁਨਰ ਮੁਲਾਂਕਣ ਅਤੇ ਤਸਦੀਕ ਕਰਤਾ

ਸੀ. ਪੀ. ਕੰਬੋਜ
ਸਰਕਾਰੀ ਹਾਈ ਸਕੂਲ, ਕਿੜਿਆਂ ਵਾਲਾ, ਫਿਰੋਜ਼ਪੁਰ

ਧਰਮਜੀਤ ਸਿੰਘ
ਸਰਕਾਰੀ ਹਾਈ ਸਕੂਲ, ਗਹਿਰੀ ਦੇਵੀ ਨਗਰ, ਬਠਿੰਡਾ

ਅਨੁਵਾਦਕ
ਸੀ. ਪੀ. ਕੰਬੋਜ

ਸੋਧਕ

ਸ੍ਰੀਮਤੀ ਸਿਮਰਤ ਭੱਟੀ
ਸ.ਸ.ਸ.ਸ, ਸੋਹਾਣਾ (ਐਸ.ਏ.ਐਸ.ਨਗਰ)

Downloaded from https:// www.studiestoday.com

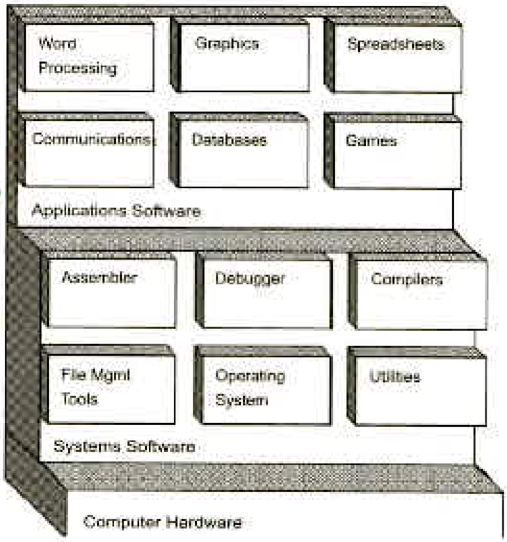
Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ 1

Review of Class X

1.0.0 ਸਾਫਟਵੇਅਰ ਸੰਕਲਪ (Review on Software Concepts)

ਪ੍ਰੋਗਰਾਮ ਹਿਦਾਇਤਾਂ ਦਾ ਇਕ ਸੈੱਟ ਹੁੰਦਾ ਹੈ ਜੋ ਦਿੱਤੀ ਗਈ ਸਮੱਸਿਆ ਨੂੰ ਹੱਲ ਕਰਨ ਲਈ ਇਕ ਵਿਸ਼ੇਸ਼ ਕ੍ਰਮ ਵਿਚ ਕੰਪਿਊਟਰ ਨੂੰ ਦਿੱਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਸਾਫਟਵੇਅਰ ਤੋਂ ਬਗੈਰ ਇਕ ਕੰਪਿਊਟਰ ਬਿਲਕੁਲ ਉਸੀ ਤਰ੍ਹਾਂ ਹੀ ਹੁੰਦਾ ਹੈ ਜਿਵੇਂ ਕਾਰ, ਬਿਨਾ ਪੈਟਰੋਲ ਦੇ।



ਸਾਫਟਵੇਅਰ ਦੀਆਂ ਵਿਸ਼ਾਲ ਸ਼੍ਰੇਣੀਆਂ ਹਨ :-

1.1 ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ : ਇਨ੍ਹਾਂ ਵਿਚ ਪ੍ਰੋਗਰਾਮਾਂ ਦੇ ਸੈੱਟ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ ਜੋ ਸੀ ਪੀ ਯੂ (CPU) ਨਾਲ ਜੁੜੇ ਅੰਦਰੂਨੀ ਕੰਮਾਂ/ਅੰਪਰੇਸ਼ਨ ਅਤੇ ਹੋਰ ਯੰਤਰਾਂ ਤੇ ਕੰਟਰੋਲ ਰੱਖਣ ਲਈ ਤਿਆਰ ਕੀਤੇ ਗਏ ਹੁੰਦੇ ਹਨ। ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ ਤੋਂ ਬਗੈਰ ਐਪਲੀਕੇਸ਼ਨ ਪੈਕੇਜਾਂ ਦੀ ਵਰਤੋਂ ਨਹੀਂ ਕੀਤੀ ਜਾ ਸਕਦੀ। ਐਮ ਐਸ ਡੋਸ ਅਤੇ ਮਾਈਕਰੋਸਾਫਟ ਵਿੰਡੋਜ਼ ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ ਜਾਂ ਉਪਰੇਟਿੰਗ ਸਾਫਟਵੇਅਰ ਦੀਆਂ ਉਦਾਹਰਨਾਂ ਹਨ।

ਟਿੱਪਣੀ : ਕੰਪਿਊਟਰ ਨਿਰਮਾਤਾ ਵੱਲੋਂ ਕੁਝ ਸਾਫਟਵੇਅਰ ਕੰਪਿਊਟਰ ਵਿਚ ਹੀ ਦਿੱਤੇ ਜਾਂਦੇ ਹਨ ਜੋ ਕੰਪਿਊਟਰ ਨੂੰ ਸਥਾਪਿਤ ਕਰਨ ਅਤੇ ਸ਼ੁਰੂ ਕਰਨ ਲਈ ਮਦਦ ਕਰਦੇ ਹਨ। ਉਦਾਹਰਨ ਵਜੋਂ ਰੈਮ ਚਿਪਸ ਅਤੇ BIOS ਵਿਚ ਲਿਖੇ ਡਾਟਾ ਨੂੰ ਪੜਨ ਲਈ।

1.1.1 ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ (Operating System)

ਇਹ ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ ਦਾ ਇਕ ਭਾਗ ਹੈ ਅਤੇ ਇਹ ਹਾਰਡਵੇਅਰ ਅਤੇ ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ ਦੇ ਵਿਚਕਾਰ ਕੰਮ ਕਰਦਾ ਹੈ। ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ ਇਕ ਪ੍ਰੋਗਰਾਮ ਹੁੰਦਾ ਹੈ ਜਿਹੜਾ ਹਾਰਡਵੇਅਰ ਦੇ ਵੱਖ-ਵੱਖ ਟੁਕੜਿਆਂ ਜਿਵੇਂ ਵੀਡੀਓ ਕਾਰਡ, ਸਾਊਂਡ ਕਾਰਡ, ਪ੍ਰਿੰਟਰ, ਮਦਰ-ਬੋਰਡ ਅਤੇ ਐਪਲੀਕੇਸ਼ਨਾਂ ਦੇ ਵਿਚਕਾਰ ਸੰਚਾਰ ਕਰਦਾ ਹੈ।

1. **ਇਹ ਕੰਪਿਊਟਰ ਹਾਰਡਵੇਅਰ ਵਲੋਂ ਇਸਦੇ ਭਾਗਾਂ ਨਾਲ ਕੰਮ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਕੰਪਿਊਟਰ ਨੂੰ ਦਸਦਾ ਹੈ ਕਿ ਕਿਵੇਂ :-**
   - ਇਹ ਸਕਰੀਨ ਤੇ ਸੂਚਨਾ ਡਿਸਪਲੇ (display) ਕਰਨੀ ਹੈ ਅਤੇ ਇਕ ਪ੍ਰਿੰਟਰ ਵਰਤਣਾ ਹੈ।
   - ਇਹ ਸੈਕੰਡਰੀ ਸਟੋਰੇਜ ਡਿਵਾਈਸ ਤੇ ਸੂਚਨਾ ਸਟੋਰ ਕਰਦਾ ਹੈ।
2. **ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ ਆਮ ਤੌਰ ਤੇ ਡਿਸਕ ਤੇ ਸਥਿਤ ਹੁੰਦਾ ਹੈ।**
   - ਇਹ ਜਾਂ ਤਾਂ ਹਾਰਡ ਡਿਸਕ ਡਰਾਈਵ, ਇਕ ਫਲਾਪੀ ਡਿਸਕ ਜਾਂ ਸੀ ਡੀ-ਰੋਮ (CD-ROM) ਡਿਸਕ ਤੇ ਹੋ ਸਕਦਾ ਹੈ
   - ਇਸ ਨੂੰ ਵਰਤਣ ਤੋਂ ਪਹਿਲਾਂ ਰੈਮ ਵਿਚ ਲੋਡ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।

ਸਿਸਟਮ ਸਾਫਟਵੇਅਰ ਜਿਹੜਾ ਪੈਰੀਫਰਲ ਯੰਤਰਾਂ ਤੇ ਕੰਟਰੋਲ ਪਾਉਂਦਾ ਹੈ, ਨੂੰ ਡਰਾਈਵਰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਉਦਾਹਰਨ ਵਜੋਂ ਇਹ ਸਾਫਟਵੇਅਰ ਇਕ ਸਿਸਟਮ ਨਾਲ ਜੁੜੀ ਸੀਮਾ ਯੰਤਰਾਂ ਨੂੰ ਚਲਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਕੁਝ ਡਰਾਈਵਰ ਆਪਣੇ ਆਪ ਵਿਚ ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ ਹੁੰਦੇ ਹਨ। ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ, ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ ਦੇ ਨਾਲ ਕੰਮ ਕਰਦਾ ਹੈ।

- ਮੁਢਲੇ ਕੰਮ ਜਿਵੇਂ ਇਕ ਦਸਤਾਵੇਜ਼ ਛਾਪਣ ਜਾਂ ਫਾਈਲ ਸੇਵ ਕਰਨ ਦਾ ਕੰਮ ਕਰਦੀ ਹੈ।
- ਆਪਰੇਟਿੰਗ ਸਿਸਟਮ ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ ਨੂੰ ਸ਼ੁਰੂ ਕਰਦਾ ਹੈ ਤਾਂ ਜੋ ਇਸ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕੇ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

1.1.2 ਯੁਟਿਲਿਟੀ ਸਾਫਟਵੇਅਰ (Utilities software)

ਯੁਟਿਲਿਟੀ ਸਾਫਟਵੇਅਰ ਨਾਲ ਤੁਸੀ ਆਪਣੇ ਕੰਪਿਊਟਰ ਤੇ ਕੁਝ ਕੰਮ ਕਰਨ ਦੇ ਕਾਬਿਲ ਹੁੰਦੇ ਹੋ। ਇਨ੍ਹਾਂ ਕੰਮਾਂ ਦੀਆਂ ਕੁਝ ਮਿਸਾਲਾਂ ਹਨ ਫਾਈਲ ਸੰਗਠਨ।

- ਕੰਪਿਊਟਰ ਨੂੰ ਚੰਗਾ ਕੰਮ ਕਰਨ ਲਈ ਅਤੇ ਸਮੱਸਿਆਵਾਂ ਨੂੰ ਦੂਰ ਰੱਖਣ ਲਈ ਵਿਸ਼ੇਸ਼ ਮੰਤਵੀ ਸਾਫਟਵੇਅਰ ਹੁੰਦਾ ਹੈ।
- ਕੁਝ ਉਪਯੋਗੀ ਪ੍ਰੋਗਰਾਮ ਐਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿਚ ਉਸਾਰੇ ਗਏ ਹਨ।
  - ਵਿੰਡੋਜ਼ ਉਪਰੇਟਿੰਗ ਸਿਸਟਮ ਵਿਚ ਸਕੈਨ-ਡਿਸਕ
  - ਡਿਸਕ ਫਾਰਮੇਟਿੰਗ ਸਾਫਟਵੇਅਰ
- ਯੂਜ਼ਰ ਪ੍ਰੋਗਰਾਮ ਦੀਆਂ ਮਿਸਾਲਾਂ
  - ਐਂਟੀ ਵਾਇਰਸ ਸਾਫਟਵੇਅਰ
  - ਡਿਸਕ ਸਾਂਭ ਸੰਭਾਲ ਸਾਫਟਵੇਅਰ
- ਫਾਈਲ ਮਨੇਜਮੈਂਟ ਪ੍ਰੋਗਰਾਮ
- ਸਿਕਉਰਟੀ (security) ਸਾਫਟਵੇਅਰ

1.1.3 ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ (Application Software)

ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ ਪ੍ਰੋਗਰਾਮਸ, ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਦੇ ਨਾਲ ਚਲਦਾ ਹੈ ਇਹ ਵਿਸ਼ੇਸ਼ ਕੰਮ ਕਰਨ ਲਈ ਕੰਪਿਊਟਰ ਵਰਤੋਂ ਵਿਚ ਮਦਦ ਕਰਦਾ ਹੈ ਜਿਵੇਂ ਇਕ ਚਿੱਠੀ ਟਾਈਪ ਕਰਨ ਲਈ ਐਮ ਐਸ ਵਰਡ-ਪ੍ਰੋਸੈਸਿੰਗ।

- ਆਮ-ਮੰਤਵੀ ਕੁਝ ਐਪਲੀਕੇਸ਼ਨ ਸਾਫਟਵੇਅਰ ਦੀਆਂ ਮਿਸਾਲਾਂ ਹਨ : ਵਰਡ-ਪ੍ਰੋਸੈਸਿੰਗ, ਡਾਟਾਬੇਸ, ਸਪਰੈਡਸ਼ੀਟ, ਡੈਸਕਟਾਪ-ਪਬਲਿਸ਼ਿੰਗ

Microsoft Office

1.2.0 ਐਕਸੈਲ (Review on Excel)

ਐਕਸੈਲ (Excel) ਇਕ ਇਲੈਕਟ੍ਰੌਨਿਕਸ ਸਪਰੈਡਸ਼ੀਟ (Spreadsheet) ਪ੍ਰੋਗਰਾਮ ਹੈ, ਜਿਸਨੂੰ ਡਾਟਾ ਸਟੋਰ, ਅਤੇ ਜੋੜ ਤੋੜ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਜਦੋ ਤੁਸੀ ਐਕਸੈਲ ਸ਼ੀਟ (Excel Sheet) ਨੂੰ ਦੇਖਦੇ ਹੋ (ਇਸ ਪੰਨੇ ਦੀ ਮਿਸਾਲ ਨੂੰ ਦੇਖੋ), ਤੁਸੀ ਇਕ ਆਇਤ (rectangular) ਟੇਬਲ ਜਾਂ ਗ੍ਰਿਡ (grid) ਨੂੰ ਦੇਖੋਗੇ। ਹੋਰੀਜੋਨਟਲ-ਰੋ (Horizontal row) ਨੂੰ ਅੰਕ (1, 2, 3) ਅਤੇ ਵਰਟਿਕਲ-ਰੋ (Vertical row) ਨੂੰ (A, B, C....) ਰਾਂਹੀ ਦਰਸਾਇਆ ਗਿਆ ਹੈ। ਕਾਲਮ 26 ਤੋ ਬਾਅਦ ਵਾਲੇ ਕਾਲਮ ਨੂੰ ਦੋ ਜਾਂ ਦੋ ਤੋ ਵੱਧ ਅੱਖਰ, ਜਿਵੇਂ AA, BB, CC ..... ਰਾਹੀ ਦਰਸਾਇਆ ਗਿਆ ਹੈ। ਜਿਥੇ ਰੋ (row) ਅਤੇ ਕਾਲਮ ਦਾ ਮੇਲ ਹੁੰਦਾ ਹੈ ਉਸਨੂੰ ਸੈੱਲ (cell) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਸੈੱਲ (cell) ਇਕ ਬੁਨੀਆਦੀ ਇਕਾਈ ਹੈ ਜਿਸ ਰਾਂਹੀ ਸਪਰੈਡਸ਼ੀਟ (spreadsheet) ਵਿਚ ਡਾਟਾ ਪ੍ਰਵੇਸ਼ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਕਿਉਂਕੀ ਐਕਸੈਲ-ਸਪਰੈਡਸ਼ੀਟ (Excel Spreadsheet) ਵਿਚ ਸੈੱਲ (cell) ਹੁੰਦੇ ਹਨ, ਹਰੇਕ ਸੈੱਲ (cell) ਤੱਕ ਪਹੁੰਚਣ ਲਈ ਸੈੱਲ reference ਜਾਂ ਐਡਰੇਸ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਸੈੱਲ (cell) reference ਲਈ ਕਾਲਮ (column), ਅੱਖਰ (letter) ਅਤੇ ਰੋ (row) ਨੰਬਰ ਦੋਵੇਂ ਇਕੱਠੇ ਹੀ ਵਰਤੇ ਜਾਂਦੇ ਹਨ ਜਿਵੇਂ ਕੀ A3, B6, A34

1.2.1 ਡਾਟਾ ਟਾਈਪ, ਫਾਰਮੂਲਾ ਅਤੇ ਫੰਕਸ਼ਨਜ਼ (Data Types, Formula & Functions)

ਇਕ ਸੈੱਲ (cell) ਵਿਚ ਕਈ ਪ੍ਰਕਾਰ ਦਾ ਡਾਟਾ ਜਿਵੇਂ ਕੀਅ ਨੰਬਰ, ਅੱਖਰ, ਜਾਂ ਫਾਰਮੂਲਾ (formula) ਆਦਿ ਸਟੋਰ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਐਕਸੈਲ (Excel) ਜਾਂ ਇਸ ਪ੍ਰਕਾਰ ਦੀਆ ਹੋਰ ਇਲੈਕਟ੍ਰੌਨਿਕ ਸਪਰੈਡਸ਼ੀਟ (Electronic Spreadsheet) ਵਿਚ ਗਣਿਤ ਫਾਰਮੂਲਾ (formula) ਉਪਲੱਬਧ ਹੁੰਦੇ ਹਨ, ਜਿਨ੍ਹਾ ਦੀ ਵਰਤੋਂ ਆਮ ਕੰਮਾਂ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ, ਅਤੇ ਇਹਨਾਂ ਨੂੰ ਫੰਕਸ਼ਨ (function) ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

1.2.2 ਐਕਸਲ ਅਤੇ ਫਾਈਨੈਂਨਸ਼ਿਯਲ ਡਾਟਾ (Excel & Financial Data)

ਸਪਰੈਡਸ਼ੀਟ (Spreadsheet) ਨੂੰ ਆਮ ਤੋਰ ਤੇ Financial ਡਾਟਾ ਸਟੋਰ ਕਰਨ ਲਈਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਪ੍ਰਕਾਰ ਦੇ ਡਾਟਾ ਉਪਰ, ਹੇਠ ਲਿਖੇ ਫਾਰਮੂਲਾ (formula) ਜਾਂ ਫੰਕਸ਼ਨ (function) ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਆਮ ਗਣਿਤਕ ਗਣਨਾ ਜਿਵੇਂ ਕਿ ਕਾਲਮਜ਼ (columns) ਜਾ ਰੋਜ਼ (rows) ਦਾ ਜੋੜ, ਲਾਭ ਜਾਂ ਹਾਨੀ ਮੁੱਲਾਂ ਨੂੰ ਲੱਭਣਾ,

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਕੁਝ ਨਿਰਧਾਰੀਤ ਡਾਟਾ ਦੀ ਔਸਤ (average) ਵੱਡੀ ਜਾ ਛੋਟੀ ਰਕਮ ਪਤਾ ਕਰਨਾ, ਕੁਝ ਹੋਰ ਕੰਮ ਜਿਨ੍ਹਾ ਲਈ ਐਕਸਲ (Excel) ਨੂੰ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ :-

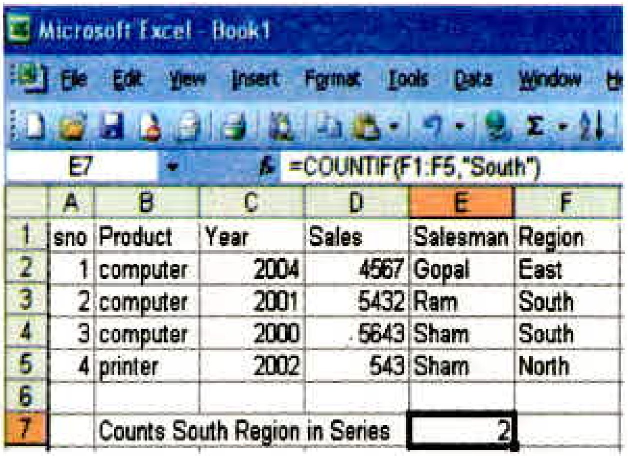
ਸਪਰੈਡਸ਼ੀਟ (Spreadsheet) ਦੁਆਰਾ ਇੱਕਠੀ ਕੀਤੀ ਗਈ ਸੂਚਨਾ ਨੂੰ ਅਸਾਨੀ ਨਾਲ ਅਸੀ ਇਲੈਕਟ੍ਰੋਨਿਕ (electronic) presentation ਵੈੱਬ ਪੇਜ (Web Page) ਜਾ ਪ੍ਰਿੰਟ ਰਿਪੋਰਟ ਵਾਂਗ ਤਿਆਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

ਐਕਸਲ ਫਾਰਮੂਲਾ (Excel formula) ਅਸਾਨ ਤੋਂ ਅਸਾਨ ਕੰਮ, ਜਿਵੇ ਕਿ ਦੋ ਨੰਬਰਾਂ ਦਾ ਜੋੜ ਜਾਂ ਬਹੁਤ ਔਖੀ ਗਣਨਾ ਜੋ ਕਿ ਕਿਸੇ ਉੱਚ ਪੱਧਰੀ ਵਪਾਰਕ ਕੰਮਾਂ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਜਦੋ ਤੁਸੀ ਫਾਰਮੂਲਾ ਦੇ ਮੂਲ ਸਿਧਾਂਤ ਨੂੰ ਲਿਖ ਲੈਂਦੇ ਹੋ ਤਾਂ ਐਕਸਲ ਤੁਹਾਡੇ ਲਈ ਸਾਰੀ ਗਣਨਾ ਕਰਦੀ ਹੈ। ਆਉ ਕੁਝ ਫੰਕਸ਼ਨਾਂ ਦੀ ਵਰਤੋਂ ਕਰੀਏ।

ਕਾਉਂਟ-ਇਫ ਫੰਕਸ਼ਨ (Countif Function)

COUNTIF ਫੰਕਸ਼ਨ ਵੀ Average ਅਤੇ SUM ਦੀ ਹੀ ਤਰ੍ਹਾਂ ਦਾ ਹੁੰਦਾ ਹੈ। ਤੁਸੀ ਪ੍ਰਵੇਸ਼ ਕਰਨ ਵਾਲੀ ਸੂਚੀ ਤਿਆਰ ਕਰਦੇ ਹੋ ਅਤੇ Excel ਨੰਬਰਾਂ ਦੇ ਮੁੱਲ ਜੋ ਕੀ ਸੂਚੀ ਵਿਚ ਹਨ, ਦੀ ਹੀ ਗਣਨਾ ਕਰਦਾ ਹੈ। COUNTIF ਫੰਕਸ਼ਨ ਕੁਝ ਵਿਸ਼ੇਸ਼ ਮੁੱਲ ਜਿਨ੍ਹਾਂ ਦੀ ਗਿਣਤੀ ਤੁਸੀ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹੋ ਨੂੰ ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦਾ ਹੈ। COUNTIF ਫੰਕਸ਼ਨ ਦਾ Syntax ਹੇਠ ਲਿਖੇ ਪ੍ਰਕਾਰ ਹੈ :-
= COUNTIF (range, criteria)

ਜਿਥੇ ਰੇਂਜ (range), ਜੋ ਕਿ ਇਕ ਦਿੱਤੀ ਗਈ ਮੁੱਲਾਂ ਦੀ ਸੂਚੀ ਹੈ, ਇਸ ਸੂਚੀ ਵਿਚੋ COUNTIF ਮੁੱਲ ਲੱਭੇਗੀ ਅਤੇ ਸ਼ਰਤ ਜੋ ਕਿ expression text ਜਾਂ ਮੁੱਲ (Value) ਹਨ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ ਕੀ ਕਿਹੜੇ ਸੈੱਲਸ (cells) ਗਿਣੇ ਜਾਣਗੇ।


Microsoft Excel - Book1

E7 =COUNTIF(F1:F5,"South")

	A	B	C	D	E	F
1	sno	Product	Year	Sales	Salesman	Region
2	1	computer	2004	4567	Gopal	East
3	2	computer	2001	5432	Ram	South
4	3	computer	2000	5643	Sham	South
5	4	printer	2002	543	Sham	North
6						
7		Counts South Region in Series			2	

On the formula bar type =COUNTIF(F1:F5,"South")

Result : Counts South Region in Series 2

ਇਫ, AND, OR ਅਤੇ NOT ਫੰਕਸ਼ਨ (ਲੋਜੀਕਲ) [IF, AND, OR & NOT Function (Logical)]

ਐਕਸਲ (Excel) ਦੇ ਸਾਰੇ ਫੰਕਸ਼ਨ ਵਿਚੋਂ, IF ਫੰਕਸ਼ਨ ਸੱਭ ਤੋ ਵੱਧ ਮਹੱਤਵਪੂਰਨ ਹੈ। ਜਿਆਦਾ ਤਰ ਸਾਫਟਵੇਅਰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ IF ਸਟੇਟਮੈਂਟ ਹੀ ਹੁੰਦੀ ਹੈ ਜਿਹੜੀ ਕੰਪਿਊਟਰ ਨੂੰ ਲੋਜਿਕ ਅਤੇ ਸ਼ਕਤੀ (power) ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। IF ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਿਸੇ ਕੰਡੀਸ਼ਨ (condition) ਨੂੰ ਸਹੀ ਜਾਂ ਗਲਤ ਦੇ ਪਰਖ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਜੇ ਕੰਡੀਸ਼ਨ (condition) ਸਹੀ ਹੈ ਤਾਂ IF ਸਟੇਟਮੈਂਟ ਕਿਸੇ ਮੁੱਲ ਨੂੰ ਵਾਪਿਸ ਭੇਜਦੀ ਅਤੇ ਜੇ ਕੰਡੀਸ਼ਨ (condition) ਗਲਤ ਹੈ ਤਾ ਇਹ else ਵਿੱਚ ਮੁੱਲ ਨੂੰ ਵਾਪਿਸ ਭੇਜੇਗੀ।

IF ਸਟੇਟਮੈਂਟ ਦਾ syntax ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਹੈ :-

= IF (Condition, Value – if – True, Value – IF – False)
Condition :
Value – IF – True
Value – IF – False

IF ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਹੋਰ ਫੰਕਸ਼ਨ ਨਾਲ ਵੀ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। AND, OR ਜਾ NOT ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਸਮੂਹ ਸ਼ਰਤੀ ਟੈਸਟ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। AND, OR ਜਾ NOT ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਸੀ ਇਹ ਪਤਾ ਕਰ ਸਕਦੇ ਹਾਂ ਕੀ ਸਮੂਹ ਕੰਡੀਸ਼ਨਲ (conditional) ਸਟੇਟਮੈਂਟ ਸਹੀ ਹੈ ਜਾ ਗਲਤ।

Downloaded from https:// www.studiestoday.com

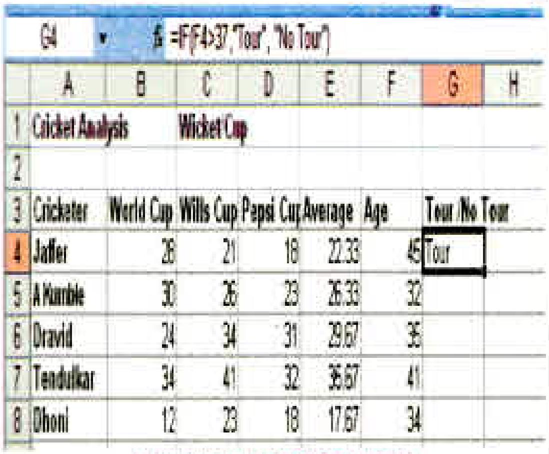
Downloaded from https:// www.studiestoday.com

IF ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋ ਕਰਦੇ ਹੋਏ ਇਕ ਕੰਡੀਸ਼ਨ test, ਹੇਠਾਂ ਲਿਖੇ steps ਅਨੁਸਾਰ ਬਣਾਓ :-

ਸੱਜੇ ਪਾਸੇ ਦਿਖਾਈ worksheet ਨੂੰ ਬਣਾਓ ਅੰਤੇ ਹੇਠਾਂ ਲਿਖੇ ਫਾਰਮੂਲਾ ਨੂੰ ਸੈੱਲ G4 ਵਿਚ ਲਿਖੋ :-

= IF (F4 > 37, "Tour", "No Tour")

ਇਸ ਫਾਰਮੂਲਾ ਨੂੰ ਸੈੱਲ G4 ਤੇ ਸੈੱਲ G5 ਤੋਂ ਲੈਕੇ ਸੈੱਲ G8 ਤੱਕ ਕਾਪੀ ਕਰੋ। ਸੈੱਲ G4 ਦੇ ਹੇਠਲੇ ਸੱਜੇ ਪਾਸੇ ਤੇ ਪੁਆਇੰਟਰ (pointer) ਰੱਖੋ ਅਤੇ ਦੇਖੋ ਕਿ ਮਾਊਸ ਕਰਸਰ (cursor) ਇਕ ਵੱਡੇ ਅਕਾਰ ਤੇ ਛੋਟੇ ਅਕਾਰ ਵਿਚ ਤਬਦੀਲ ਹੋ ਜਾਵੇਗਾ। ਹੁਣ ਮਾਊਸ ਪੁਆਇੰਟਰ (pointer) ਨੂੰ ਸੈੱਲ G8 ਤੱਕ ਲੈ ਜਾਉ।


G4 =IF(F4>37,"Tour","No Tour")

	A	B	C	D	E	F	G	H
1	Cricket Analysis		Wicket Cup					
2								
3	Cricketer	World Cup	Wills Cup	Pepsi Cup	Average	Age	Tour /No Tour	
4	Jaffer	28	21	18	22.33	45	Tour	
5	A Kumble	30	26	23	26.33	32		
6	Dravid	24	34	31	29.67	35		
7	Tendulkar	34	41	32	35.67	41		
8	Dhoni	12	23	18	17.67	34		

Worksheet with IF(Condition)

ਫਾਰਮੂਲਾ ਬਾਕਸ (box) ਵਿਚ ਫਾਰਮੂਲਾ ਨੂੰ ਦੇਖੋ। ਇਹ ਫਾਰਮੂਲਾ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਜੇਕਰ Bowler ਦੀ age 37 ਤੇ ਵੱਡੀ ਹੈ ਤਾਂ ਕੰਡੀਸ਼ਨ (condition) ਸਹੀ ਹੈ ਅਤੇ Text ਮੁੱਲ "Tour" ਵਾਪਿਸ ਆਵੇਗਾ। ਜੇਕਰ Bowler ਦੀ ਉਮਰ 37 ਸਾਲ ਤੋ ਵੱਧ ਨਹੀਂ ਹੈ ਤਾਂ ਕੰਡੀਸ਼ਨ (condition) ਗਲਤ ਹੋਵੇਗੀ ਅਤੇ ਇਹ ਕੰਡੀਸ਼ਨ (condition) Text ਮੁੱਲ "No Tour" ਵਾਪਿਸ ਭੇਜੇਗੀ।

Age ਜਾਂ average ਕੱਢਣ ਲਈ, ਹੇਠਾ ਲਿਖੇ steps ਦੀ ਮਦਦ ਦੇ ਨਾਲ ਇਕ ਸਮੂਹ ਕੰਡੀਸ਼ਨ ਬਣਾਉ, IF ਅਤੇ AND ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰ ਕੇ :-

H3 ਸੈੱਲ ਵਿਚ, Age ਜਾਂ Avg ਟਾਇਪ ਕਰੋ।
ਸੈੱਲ H4 ਵਿਚ ਹੇਠ ਲਿਖਿਆ ਫਾਰਮੂਲਾ ਟਾਇਪ ਕਰੋ :

= IF (AND (F4 > 37, E4 > = 26), "TOUR", "NO TOUR"))

ਸੈੱਲ H4 ਤੇ ਸੈੱਲ H5 ਤੋਂ ਲੈ ਕੇ ਸੈੱਲ H8 ਤਕ ਫਾਰਮੂਲਾ ਨੂੰ ਕਾਪੀ ਕਰੋ। ਸੈੱਲ H4 ਦੇ ਹੇਠਲੇ ਸੱਜੇ ਪਾਸੇ ਤੇ pointer ਰੱਖੋ ਅਤੇ ਦੇਖੋ ਕੀ ਮਾਊਸ ਕਰਸਰ (cursor) ਇਕ ਵੱਡੇ ਆਕਾਰ ਤੇ ਛੋਟੇ ਆਕਾਰ ਵਿਚ ਤਬਦੀਲ ਹੋ ਜਾਵੇਗਾ। ਹੁਣ ਮਾਊਸ ਪੁਆਇੰਟਰ (pointer) ਨੂੰ ਸੈੱਲ H8 ਤੱਕ ਲੇ ਕੇ ਜਾਓ।

SUM =IF(AND(F4>37,E4>=26),"Tour","No Tour")

	A	B	C	D	E	F	G	H
1	Cricket Analysis		Wicket Cup					
2								
3	Cricketer	World Cup	Wills Cup	Pepsi Cup	Average	Age	Tour /No Tour	Avg or Age
4	Jaffer	28	21	18	22.33	45	Tour	No Tour")
5	A Kumble	30	26	23	26.33	32	No Tour	
6	Dravid	24	34	31	29.67	35	No Tour	
7	Tendulkar	34	41	32	35.67	41	Tour	
8	Dhoni	12	23	18	17.67	34	No Tour	

Worksheet with IF(AND) Condition

ਫਾਰਮੂਲਾ ਬਾਕਸ (box) ਵਿਚ ਫਾਰਮੂਲਾ ਨੂੰ ਦੇਖੋ। ਇਹ ਫਾਰਮੂਲਾ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਜੇਕਰ Bowler ਦੀ age 37 ਤੇ ਵੱਡੀ ਹੇ ਅਤੇ Bowler ਦੀ ਔਸਤ (average) = 26 ਜਾਂ ਵੱਧ ਹੇ ਤਾਂ ਕੰਡੀਸ਼ਨ (condition) ਸਹੀ ਹੇ ਅਤੇ Text ਮੁੱਲ "Tour" ਵਾਪਿਸ ਆਵੇਗਾ। ਜੇਕਰ Bowler ਦੀ ਉਮਰ 37 age ਤੋ ਵੱਧ ਨਹੀਂ ਹੇ ਅਤੇ Bowler ਦੀ ਔਸਤ (average) = 26 ਤੋਂ ਵੱਧ ਜਾਂ ਬਰਾਬਰ ਨਹੀਂ ਤਾਂ ਕੰਡੀਸ਼ਨ (condition) ਗਲਤ ਹੋਵੇਗੀ ਅਤੇ ਇਹ ਕੰਡੀਸ਼ਨ (condition) Text ਮੁੱਲ "No Tour" ਵਾਪਿਸ ਭੇਜੇਗੀ।

VLOOKUP ਫੰਕਸ਼ਨ (VLOOKUP Function)

ਕਈ ਵਾਰੀ ਤੁਹਾਨੂੰ ਕਿਸੇ ਹੋਰ ਟੇਬਲ ਵਿਚ ਮੁੱਲ ਵੇਖਣ ਦੀ ਜਰੂਰਤ ਪੈਂਦੀ ਹੈ। ਮਿਸਾਲ ਲਈ, ਜੇਕਰ ਤੁਸੀ ਅਧਿਆਪਕ ਹੋ ਅਤੇ ਗ੍ਰੇਡ-ਸ਼ੀਟ (Grade sheet) ਬਣਾਉਂਦੇ ਹੋ, ਤਾਂ ਤੁਹਾਨੂੰ ਐਕਸਲ ਦੀ ਜਰੂਰਤ ਗ੍ਰੇਡ (grade) ਜੋ ਕਿ students ਦੇ test score average ਤੇ ਆਧਾਰਿਤ ਹੋਵੇਗੀ, ਨੂੰ ਵੇਖਣ ਲਈ ਪੈ ਸਕਦੀ ਹੈ।ਇਸ ਕੰਮ ਲਈ VLOOK UP Function ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।
VLOOK UP (LOOK-up-value ...)

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

VLOOKUP ਫਾਰਮੂਲਾ (VLOOK UP formula)

ਹੇਠਾਂ ਦਰਸਾਇਆ ਟੇਬਲ ਬਣਾਓ :-

ਇਸ ਟੇਬਲ ਨੂੰ ਸਹੀ ਕਰਨ ਦੀ ਜਰੂਰਤ ਹੈ ਤਾਂ ਜੋ ਮੁੱਲ ਜੋ ਤੁਸੀ ਲੱਭਣਾ ਚਾਹੁੰਦੇ ਹੋ ਖੱਬੇ ਪਾਸੇ ਵਾਲੇ ਕਾਲਮ (column) ਵਿਚ ਆਉਣ, ਜਿਵੇਂ ਹੇਠਾਂ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।

H5 =VLOOKUP(76,J4:K8,2)

	A	B	C	D	E	F	G	H	I	J	K
1	Annual Report				Class IX						
2											
3	Name	Maths	English	Science	Bio	Total	Average	Grade		Lookup Table	
4	Jaffer	45	45	56	67	213	53.25	F		0	F
5	A Kumble	90	56	67	90	303	75.75	C		60	D
6	Dravid	70	90	93	89	342	85.50	B		70	C
7	Tendulkar	67	78	56	67	268	67.00	D		80	B
8	Dhoni	45	56	67	78	246	61.50	F		90	A
9	Laxman	89	90	89	89	357	89.25	B			
10											

ਉਸ ਸੈੱਲ ਤੇ ਕਲਿੱਕ ਕਰੋ ਜਿਥੇ ਤੁਸੀ ਰਿਜ਼ਲਟ ਦੇਖਣਾ ਚਾਹੁੰਦੇ ਹੋ ਅਤੇ ਫਿਰ Paste ਫੰਕਸ਼ਨ ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।

All or Lookup & Reference ਸ਼੍ਰੇਣੀਆ ਵਿਚ, VLook-up ਫੰਕਸ਼ਨ ਤੇ ਕਲਿੱਕ ਕਰੋ। VLookup ਟੇਬਲ ਨਜਰ ਆਵੇਗਾ, ਜਿਵੇਂ ਕਿ ਉਪੱਰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।

Look-up-value Box ਤੇ ਕਲਿੱਕ ਕਰੋ ਅਤੇ ਸੈੱਲ ਤੇ ਕਲਿੱਕ ਕਰੋ ਜਿਸ ਵਿਚ ਮੁੱਲ ਜੋ ਤੁਸੀ ਦੇਖਣਾ ਚਾਹੁੰਦੇ ਹੋ ਸ਼ਾਮਲ ਹੋਵੇ। (ਇਥੇ ਇਹ ਮੁੱਲ Average ਹੈ)

ਟੇਬਲ ਐਰੇ ਬਾਕਸ ਤੇ ਕਲਿੱਕ ਕਰੋ ਅਤੇ look-up ਟੇਬਲ ਨੂੰ select ਕਰੋ।

Col_Index_num box ਵਿਚ Look-up ਟੇਬਲ ਦੇ ਕਿੰਨੇ ਕਾਲਮ ਹਨ ਟਾਇਪ ਕਰੋ, ਜਿਸ ਤਰ੍ਹਾਂ ਇਸ ਸਥਿਤੀ ਵਿਚ ਸਾਡੇ ਕੋਲ 2 ਕਾਲਮ ਹਨ।

Range Lookup ਵਿਚ ਤੁਸੀ ਇਹ ਫ਼ੈਸਲਾ ਕਰਨਾ ਹੈ ਕਿ ਅਸੀ ਬਿਲਕੁਲ ਸਹੀ ਮੁੱਲ ਜੋ ਕੀ ਸੈੱਲ ਵਿਚ ਹੈ (ਮਿਸਾਲ ਲਈ G4 = 53.25) ਜਾਂ ਤੁਸੀ ਲਗ ਭਗ ਮੁੱਲ ਨੂੰ ਵੇਖਣਾ ਚਾਹੁੰਦੇ ਹੋ। ਬਿਲਕੁਲ ਸਹੀ ਮੇਲ ਲਈ, ਅਸੀ Look-up- range ਵਿਚ false ਲਿਖਾਂਗੇ ਅਤੇ ਲਗ ਭਗ ਮੇਲ ਲਈ ਇਸ ਸਥਾਨ ਨੂੰ ਖਾਲੀ ਰਖਾਂਗੇ ਜਿਵੇਂ ਕਿ ਉਪਰ ਫਾਰਮੂਲਾ ਬਾਕਸ (box) ਵਿਚ ਦਰਸਾਇਆਂ ਗਿਆ ਹੈ। Look-up ਟੇਬਲ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਅਸੀ ਵਧਦੇ ਕ੍ਰਮ ਵਿਚ ਲਿਖਾਂਗੇ ਅਤੇ ਇਸਤੋਂ ਬਾਅਦ OK ਤੇ Enter ਪ੍ਰੈਸ ਕਰੋ।

ਇਸ ਤੋਂ ਬਾਅਦ, ਇਸ ਫਾਰਮੂਲੇ ਨੂੰ ਬਾਕੀ ਰਹਿੰਦੀਆਂ ਹੇਠਾਂ ਵਾਲੀਆਂ ਰੋ (rows) ਤੇ ਕਾਪੀ ਕਰੋ, ਤੇ Enter ਪ੍ਰੈਸ ਕਰੋ।

1.3.0 Review on HTML (ਹਾਈਪਰ ਟੈਕਸਟ ਮਾਰਕਪ ਲੈਂਗੁਏਜ਼)

HTML ਭਾਸ਼ਾ ਡਾਕੂਮੈਂਟ ਨੂੰ world wide web ਤੇ ਉਤਪਨ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਅਲੱਗ ਤਰ੍ਹਾਂ ਦੇ ਟੈਗਸ (Tags) ਅਤੇ ਗੁਣਾਂ ਦੀ ਵਰਤੋਂ ਨਾਲ, HTML, ਵੈੱਬ ਡਾਕੂਮੈਂਟ ਦੇ ਢਾਂਚੇ ਅਤੇ ਰੂਪਰੇਖਾ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। HTML ਡਾਕੂਮੈਂਟ ਦਾ ਸਹੀ ਢਾਂਚਾ <HTML><title><HEAD> (ਇੱਥੇ ਡਾਕੂਮੈਂਟ ਬਾਰੇ ਐਂਟਰ ਕਰੋ) ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਸ ਦੀ body, <Body>tag ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦੀ ਹੈ <Body></HTML>tag ਅਤੇ ਨਾਲ ਖਤਮ ਹੁੰਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਾਰੀ ਸੂਚਨਾ ਜਿਹੜੀ ਤੁਸੀ ਆਪਣੇ ਵੈੱਬ ਪੇਜ਼ ਵਿਚ ਸ਼ਾਮਲ ਕਰਨੀ ਚਾਹੁੰਦੇ ਹੋ, <Body> ਅਤੇ </Body> ਟੈਗਸ (Tags) ਵਿਚ ਹੁੰਦੀ ਹੈ। ਇਥੇ ਲੱਖਾਂ ਦੀ ਤਾਦਾਤ ਵਿਚ ਹੋਰ ਵੀ ਟੈਗਸ (Tags) ਹਨ ਜਿਹੜੇ ਵੈੱਬ ਪੇਜ਼ ਵਿਚ ਸੂਚਨਾ ਦੇ ਫਾਰਮੈਟ ਅਤੇ ਰੂਪਰੇਖਾ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਟੈਗਸ (Tags) ਹਾਇਪਰਟੈਕਸਟ (Hypertext) ਲਿੰਕ ਨੂੰ ਬਿਆਨ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

1.3.1 ਵੈੱਬ ਪੇਜਿਸ (Web Pages)

ਵੈੱਬ ਪੇਜਿਸ ਦੇ ਬਹੁਤ ਵਰਤਾਰੇ (uses) ਹਨ। ਇਥੇ ਕੁਝ ਮਹੱਤਵਪੂਰਨ ਤੱਥ ਹਨ ਕਿ ਵੈੱਬ ਪੇਜਿਸ ਕਿਉਂ ਫਾਇਦੇਮੰਦ ਹਨ :

- ਇਹ ਸੂਚਨਾ ਨੂੰ ਵੱਡੇ ਪੱਧਰ ਤੇ ਫੈਲਾਉਣ ਲਈ ਸਸਤਾ ਅਤੇ ਸੌਖਾ ਤਰੀਕਾ ਹੈ।
- ਤੁਹਾਡੇ ਵਪਾਰ ਨੂੰ ਵਧਾਉਣ ਦਾ ਹੋਰ ਵਸੀਲਾ।
- ਸਾਰੀ ਦੁਨਿਆ ਨੂੰ ਤੁਹਾਡੇ ਬਾਰੇ ਨਿਜੀ ਵੈਬ ਸਾਈਟ ਰਾਹੀਂ ਜਾਣੂੰ ਕਰਾਉਣਾ।

ਸ਼ਬਦਾਂ ਨੂੰ ਜਾਣਨਾ (Words to know)

ਟੈਗ (Tag) – Web Browser ਦੇ HTML ਡਾਕੂਮੈਂਟ ਦੇ ਖਾਸ ਖੇਤਰਾਂ ਦੇ ਅਰਥ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

ਟੈਗ ਇਸ ਤਰ੍ਹਾਂ ਦਿਖਦਾ ਹੈ : <tag>

ਐਲੀਮੈਂਟ (Element) : ਇਕ ਸੰਪੂਰਨ ਟੈਗ (Tag) ਹੈ। ਇਸ ਵਿਚ Opening<tag> ਅਤੇ closing</tag> ਹੁੰਦੇ ਹਨ।

ਐਟਰੀਬਿਉਟ (Attribute) : HTML ਐਲੀਮੈਂਟ (element) ਦੇ ਮੁੱਲਾਂ ਦਾ ਸੁਧਾਰ ਕਰਦਾ ਹੈ। ਐਲੀਮੈਂਟ (element) ਬਹੁਗੁਣ ਵੀ ਹੋ ਸਕਦੇ ਹਨ।

1.3.2 HTML ਫਾਈਲ ਕੀ ਹੈ (What is HTML File)

ਇਹ ਇਕ ਟੈਕਸਟ (Text) ਫਾਈਲ ਹੁੰਦੀ ਹੈ ਜਿਸ ਵਿਚ ਛੋਟੇ ਨਿਸ਼ਾਨ (Markup) ਟੈਗ ਹੁੰਦੇ ਹਨ। ਨਿਸ਼ਾਨ (Markup) ਟੈਗਸ, ਵੈੱਬ ਬਰਾਊਜ਼ਰ (Browser) ਨੂੰ ਦੱਸਦੇ ਹਨ ਕਿਵੇਂ ਪੇਜ਼ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨਾ ਹੈ।

HTML ਫਾਈਲ ਦੀ extension htm ਜਾਂ html ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ।

ਸੌਖੇ ਟੈਕਸਟ (Text) ਐਡੀਟਰ (editor) ਦੀ ਮਦਦ ਨਾਲ HTML ਫਾਈਲ ਬਣਾਈ ਜਾ ਸਕਦੀ ਹੈ।

ਜੇਕਰ ਤੁਸੀਂ ਵਿੰਡੋਜ਼ run ਕਰ ਰਹੇ ਹੋ ਤਾਂ ਨੋਟ-ਪੈਡ (Notepad) ਸਟਾਰਟ ਕਰੋ :

ਖੱਬੇ ਪਾਸੇ ਨਿਮਨਲਿਖਤ Text ਟਾਇਪ ਕਰੋ ਸੇਵ ਕਰੋ ਜਿਵੇਂ "My Page.htm" ਇੰਟਰਨੈੱਟ ਬਰੋਜ਼ਰ (internet browser) ਨੂੰ ਸ਼ੁਰੂ ਕਰੋ ਅਤੇ File menu ਵਿਚ open ਤੇ ਕਲਿੱਕ ਕਰੋ। ਤੁਹਾਨੂੰ ਡਾਈਲਾਗ ਬਾਕਸ (dialog box) ਦਿਖੇਗਾ। Select "Browse" ਅਤੇ ਤੁਹਾਡੇ ਵੱਲੋਂ ਬਣਾਈ HTML ਫਾਈਲ ਲੱਭੋ "mypage.htm" ਇਸ ਨੂੰ ਚੁਣੋ ਅਤੇ "open" ਕਲਿੱਕ ਕਰੋ।	**<html> <head> <title>Title of page</title> </head> <body> This is my first HTML page. This text is bold </body> </html>**

ਹੁਣ ਤੁਸੀਂ Dialog box ਤੇ **"C:\MyDocuments\mypage.html"** ਐਡਰੇਸ ਦੇਖ ਸਕਦੇ ਹੋ। OK ਤੇ ਕਲਿੱਕ ਕਰੋ ਅਤੇ Browser ਪੇਜ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇਗਾ।

ਮਿਸਾਲ (Example)

ਪਹਿਲਾ ਟੈਗ (Tag) ਤੁਹਾਡੇ HTML ਡਾਕੂਮੈਂਟ ਵਿਚ <html> ਹੈ। ਇਹ ਟੈਗ (Tag) ਤੁਹਾਡੇ ਬਰਾਊਜ਼ਰ (Browser) ਨੂੰ ਦੱਸਦਾ ਹੈ ਕਿ ਇਹ HTML ਡਾਕੂਮੈਂਟ ਦੀ ਸ਼ੁਰੂਆਤ ਹੈ। ਆਖਰੀ ਟੈਗ (Tag) ਤੁਹਾਡੇ ਡਾਕੂਮੈਂਟ ਵਿਚ </html> ਹੈ। ਇਹ ਟੈਗ (Tag) ਤੁਹਾਡੇ ਬਰੋਜ਼ਰ ਨੂੰ ਦੱਸਦਾ ਹੈ ਕਿ ਇਹ HTML ਡਾਕੂਮੈਂਟ ਦਾ ਅੰਤ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

<Head> ਟੈਗ (Tag) ਅਤੇ </head> ਟੈਗ ਦੇ ਵਿਚਕਾਰ ਵਾਲਾ Text Header ਸੂਚਨਾ ਹੈ। Header ਸੂਚਨਾ ਬਰੋਜ਼ਰ ਵਿੰਡੋਜ਼ ਵਿਚ ਦਰਸਾਈ ਨਹੀਂ ਗਈ।

<Title> ਟੈਗਸ (Tags) ਦੇ ਵਿਚਕਾਰਲਾ ਟੈਕਸਟ (Text) ਤੁਹਾਡੇ ਡਾਕੂਮੈਂਟ ਦਾ ਟਾਈਟਲ (Title) ਹੁੰਦਾ ਹੈ। ਟਾਈਟਲ ਬਰੋਜ਼ਰ ਦੇ Caption ਅੰਦਰ ਪ੍ਰਦਰਸ਼ਿਤ ਹੁੰਦਾ ਹੈ।

<Body> ਟੈਗਸ (Tags) ਦੇ ਵਿਚਕਾਰਲਾ ਟੈਕਸਟ ਤੁਹਾਡੇ ਬਰਾਊਜ਼ਰ ਵਿਚ ਪ੍ਰਦਰਸ਼ਿਤ ਹੋਵੇਗਾ।

**** ਅਤੇ ਟੈਗਸ (Tags) ਦੇ ਵਿਚਕਾਰਲਾ ਟੈਕਸਟ, ਬੋਲਡ ਫੌਂਟ ਵਿਚ ਪ੍ਰਦਰਸ਼ਿਤ ਹੋਵੇਗਾ।

ਜਰਾ ਦੇਖੀਏ ਟੈਗਸ (Tags) ਦੀ ਵਰਤੋਂ ਨਾਲ ਟੇਬਲ ਕਿਵੇਂ ਬਣਾਉਣਾ ਹੈ ਅਤੇ ਇਸ ਵਿਚ ਕਿਵੇਂ ਡਾਟਾ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨਾ ਹੈ।

ਟੇਬਲ ਦੀ ਪਰਿਭਾਸ਼ਾ <table> ਟੈਗ (Tag) ਨਾਲ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਹਰ ਟੇਬਲ ਰੋ (row) ਵਿਚ ਵੰਡਿਆ ਹੁੰਦਾ ਹੈ ਜਿਸਦਾ ਟੈਗ (Tag) ਹੁੰਦਾ ਹੈ <tr> ਅਤੇ ਹਰ row, cell ਵਿਚ ਵੰਡੀ ਹੁੰਦੀ ਹੈ (ਇਸਦਾ ਟੈਗ (Tag) ਹੁੰਦਾ ਹੈ <td>) ਅਤੇ <td> ਦਾ ਮਤਲੱਬ ਹੈ ਟੇਬਲ-ਡਾਟਾ (table data) ਜੋ ਕਿ ਡਾਟਾ-ਸੈੱਲ (data cell) ਦੀ ਵਿਸ਼ੇਸ਼ ਸੂਚੀ ਹੈ। ਡਾਟਾ ਸੈੱਲ ਵਿਚ text, images, list, paragraph ਅਤੇ tables ਹੋ ਸਕਦੇ ਹਨ।

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

ਬਰਾਊਜ਼ਰ (Browser) ਵਿਚ ਇਹ ਕਿਵੇਂ ਦਿਸਦੇ ਹਨ

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

ਟੇਬਲ ਅਤੇ ਬਾਰਡਰ ਐਟਰੀਬਿਊਟਸ (Tables & the Border attributes) :-

ਜੇਕਰ ਤੁਸੀਂ ਬਾਰਡਰ ਐਟਰੀਬਿਊਟਸ ਸੁਨਿਸ਼ਚਿਤ ਨਹੀਂ ਕਰਦੇ ਤਾਂ ਟੇਬਲ ਬਿਨ੍ਹਾਂ ਬਾਰਡਰ ਤੋਂ ਪ੍ਰਦਰਸ਼ਿਤ ਹੋ ਸਕਦੇ ਹਨ। ਕਦੇ ਇਹ ਲਾਭਦਾਇਕ ਹੋ ਸਕਦਾ ਹੈ ਪਰ ਜਿਆਦਾਤਰ ਸਮਾਂ ਤੁਸੀਂ ਚਾਹੁੰਦੇ ਹੋ ਕੇ ਬਾਰਡਰ ਦਿਖਾਈ ਦੇਣ। ਟੇਬਲ ਨੂੰ ਬਾਰਡਰ ਨਾਲ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ ਤੁਹਾਨੂੰ ਬਾਰਡਰ ਐਟਰੀਬਊਟ ਵਰਤਣਾ ਪਵੇਗਾ।

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

ਟੇਬਲ ਵਿਚ ਸੁਰਖੀਆਂ <th> ਟੈਗ (Tag) ਨਾਲ ਪਰਿਭਾਸ਼ਿਤ ਹੁੰਦੀਆਂ ਹਨ।

```
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

Heading	Another Heading
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੱਮਸਿਆ (Problem): ਆਉਟਪੁਟ ਨੂੰ ਡਿਸਪਲੇ (display) ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜਿਵੇਂ :-

row 1, cell 1	row 1, cell 2
row 2, cell 1	

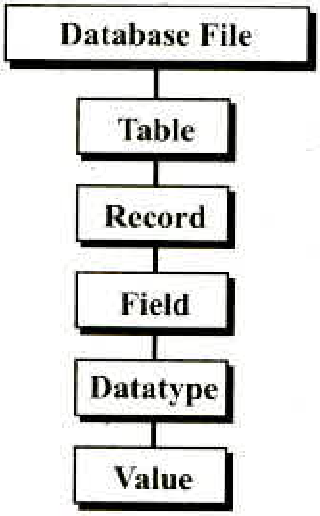
ਉੱਤਰ :- ਟੇਬਲ ਵਿਚ ਖਾਲੀ ਸੈਲੱਜ (Cells)
ਟੇਬਲ ਸੈਲਸ ਬਿਨ੍ਹਾਂ ਵਿਸ਼ਾ ਤੋਂ ਬਰੋਜ਼ਰ ਵਿਚ ਚੰਗੀ ਤਰ੍ਹਾਂ ਪ੍ਰਦਰਸ਼ਿਤ ਨਹੀਂ ਹੁੰਦੇ।

1.4.0 ਮਾਇਕਰੋਸਾਫਟ ਅਸੈਸ (Review on Microsoft Access)

ਮਾਇਕਰੋਸਾਫਟ ਅਸੈਸ (Microsoft Access Databases) ਨੂੰ ਬਣਾਉਣ ਅਤੇ ਪ੍ਰਬੰਧ ਕਰਨ ਵਿਚ ਇਕ ਸ਼ਕਤੀਸ਼ਾਲੀ ਪ੍ਰੋਗਰਾਮ ਹੈ। ਇਸ ਵਿਚ ਕਈ ਅੰਤਰ ਵਿਆਪੀ ਖਾਸੀਅਤਾਂ ਹਨ ਜੋ ਤੁਹਾਨੂੰ ਸੂਚਨਾ ਦੇ ਬਣਾਉਣ ਅਤੇ ਦਿਖਾਉਣ ਵਿਚ ਮਦਦ ਕਰਦੀਆਂ ਹਨ। Access ਦੂਜੇ ਪ੍ਰੋਗਰਾਮਸ ਜਿਵੇਂ Microsoft words ਨਾਲ ਪੇਚਦਾਰ ਅਤੇ ਭਰੋਸੇਯੋਗ ਡਾਟਾ-ਬੇਸ ਐਪਲੀਕੇਸ਼ਨ ਹੈ।

ਕੁਝ ਕੀ-ਵਰਡਸ (key words) ਇਸ ਪ੍ਰਕਿਰਿਆ ਵਿਚ ਸ਼ਾਮਲ ਹਨ ਡਾਟਾ-ਬੇਸ ਫਾਈਲ, ਟੇਬਲ, ਖੇਤਰ (field), ਡਾਟਾ ਟਾਈਪ, ਮੁੱਲ ਆਦਿ ਜਿਸ ਤਰ੍ਹਾਂ ਕਿ ਹੇਠਾਂ ਦਰਸਾਈਆ ਗਿਆ ਹੈ।

Some keywords involved in this process are: *Database File, Table, Record, Field, Data-type.* Here is the Hierarchy that Microsoft Access uses in breaking down a database.



ਡਾਟਾ-ਬੇਸ (Database File): ਇਹ ਤੁਹਾਡੀ ਮੁੱਖ ਫਾਈਲ ਹੈ ਜੋ ਸਾਰੇ ਡਾਟਾ-ਬੇਸ ਨੂੰ ਘੇਰਾ ਪਾਉਂਦੀ ਹੈ ਅਤੇ ਤੁਹਾਡੀ ਹਾਰਡ ਡਿਸਕ ਜਾਂ ਫਲੋਪੀ ਡਿਸਕ ਵਿਚ ਸੇਵ ਕਰਦੀ ਹੈ।
Example) StudentDatabase.mdb

ਟੇਬਲ (Table): ਟੇਬਲ ਇਕ ਸੁਨਿਸ਼ਚਿਤ ਭਾਵ ਦੇ Data ਦਾ ਸਮੂਹ ਹੈ। ਡਾਟਾ-ਬੇਸ ਵਿਚ ਬਹੁਤ ਟੇਬਲ ਹੋ ਸਕਦੇ ਹਨ।
Example #1) Students
Example #2) Teachers

ਖੇਤਰ (Field): ਖੇਤਰ, ਟੇਬਲ ਵਿਚ ਅਲੱਗ ਸ਼੍ਰੇਣੀਆਂ ਹਨ। ਟੇਬਲ ਆਮ ਤੌਰ ਤੇ ਬਹੁਤ ਖੇਤਰ ਵਾਲੇ ਹੁੰਦੇ ਹਨ।
Example #1) Student LastName
Example #2) Student FirstName

ਡਾਟਾ ਟਾਈਪਸ Datatypes: ਡਾਟਾ ਟਾਈਪਸ ਹਰ ਇਕ ਖੇਤਰ ਦੀਆਂ ਵਿਸ਼ੇਸ਼ਤਾਵਾ ਹੁੰਦੀਆਂ ਹਨ।
(ਖੇਤਰ ਨਾਮ) ਸਟੂਡੈਂਟ ਦਾ ਆਖਰੀ ਨਾਮ ਡਾਟਾ ਟਾਈਪ)
Text

1.4.1 ਡਾਟਾ ਸੋਧਨਾ (Manipulating Data)

ਰਿਕਾਰਡ ਵਿਚ ਨਵੀਂ ਰੋ ਜੋੜਨਾ (adding new row in record): ਨਵੀਂ ਰੋ (row) ਤੋਂ ਥੱਲੇ ਵਲ ਨੂੰ ਆਉ ਅਤੇ ਸੂਚਨਾ enter ਕਰੋ।

ਰਿਕਾਰਡ ਦਾ ਨਿਰਧਾਰਣ (Updating a record) : ਜਿਸ ਖੇਤਰ ਜਾਂ record ਨੂੰ ਨਿਰਧਾਰਤ ਕਰਨਾ ਹੈ select ਕਰੋ ਅਤੇ ਇਸ ਦਾ ਡਾਟਾ ਜਿਵੇਂ ਤੁਸੀ ਚਾਹੁੰਦੇ ਹੋ ਬਦਲੋ।

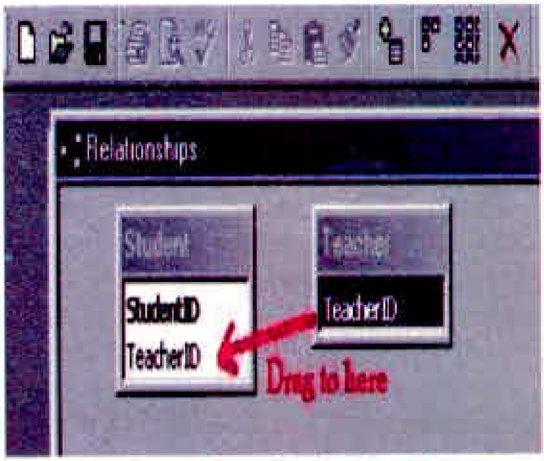
ਰਿਕਾਰਡ ਵਿਚੋ ਇਕ ਰੋ ਨੂੰ ਮਿਟਾਉਣਾ (Deleting a line in record) : ਉਸ ਰੋ (row) ਨੂੰ select ਕਰੋ ਅਤੇ Keyboard ਤੋਂ Delete ਬੱਟਨ ਦਬਾਉ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਰਿਲੇਸ਼ਨਸ਼ਿਪ (Relationship) :-

ਜੇਕਰ ਤੁਹਾਡੇ ਕੋਲ Microsoft Access ਡਾਟਾ-ਬੇਸ ਵਿਚ ਬਹੁਤ ਟੇਬਲ ਹਨ ਅਤੇ ਤੁਸੀਂ ਸੂਚਨਾ ਨੂੰ ਇਕ ਤੋਂ ਵੱਧ ਟੇਬਲ ਵਿਚੋਂ ਰੱਖਣਾ ਚਾਹੁੰਦੇ ਹੋ। ਪਹਿਲਾ ਕਦਮ ਇਸ ਪ੍ਰਕਿਰਿਆ ਵਿਚ ਆਪਣੇ ਟੇਬਲ ਵਿਚ ਸੰਬੰਧੀਕਰਨ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨਾ ਹੈ। ਇਹ ਕਰਨ ਤੋਂ ਬਾਅਦ ਤੁਸੀਂ Queries Forms ਅਤੇ Reports ਨੂੰ ਬਣਾ ਕੇ ਸੂਚਨਾ ਨੂੰ ਕਈ ਟੇਬਲਸ ਨਾਲ ਇਕਦਮ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰ ਸਕਦੇ ਹੋ। ਸੰਬੰਧੀਕਰਨ ਵਿਸ਼ੇਸ਼ ਖੇਤਰਾਂ ਵਿਚ Data ਦੇ ਮੇਲ ਨਾਲ ਕੰਮ ਕਰਦਾ ਹੈ - ਅਕਸਰ ਇਕ ਖੇਤਰ (Field) ਦੋਵੇਂ ਟੇਬਲਸ ਵਿਚ ਸਮਾਨ ਨਾਮ ਨਾਲ ਜ਼ਿਆਦਾ ਤੌਰ ਤੇ, ਇਹ ਮੇਲ-ਖੇਤਰ (matching fields) ਇਕ ਟੇਬਲ ਦੇ ਅਸਲੀ Key ਹਨ ਜਿਹੜੇ ਹਰ ਇਕ ਰਿਕਾਰਡ ਲਈ ਖਾਸ ਆਈਡੈਂਟੀਫਾਇਰਸ (Identifires) ਅਤੇ ਦੂਜੇ ਟੇਬਲ ਵਿਚ foreign key ਮੁਹੱਈਆ ਕਰਾਉਂਦੇ ਹਨ।

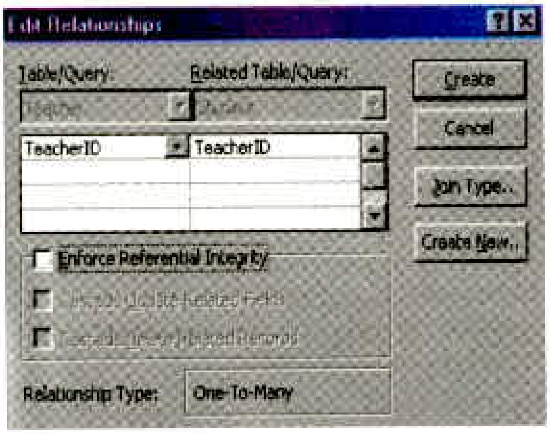


ਦੋ ਟੇਬਲਾਂ ਦਰਮਿਆਨ ਸੰਬੰਧ ਵਿਖਾਉਂਦਾ ਹੋਇਆ

ਮਿਸਾਲ :- ਅਧਿਆਪਕ (teacher), ਸਟੂਡੈਂਟਸ (students) ਨਾਲ ਜੋੜੇ ਜਾ ਸਕਦੇ ਹਨ ਅਤੇ ਇਹ ਅਧਿਆਪਕਾਂ (teachers) ਟੇਬਲ ਅਤੇ ਸਟੂਡੈਂਟਸ ਟੇਬਲ ਦੇ ਸੰਬੰਧੀਕਰਨ ਨੂੰ Teacher ID fields ਦੀ ਮਦਦ ਨਾਲ ਬਣਾਉਣ ਵਿਚ ਜ਼ਿੰਮੇਵਾਰ ਹੁੰਦੇ ਹਨ।

ਉਪਰ ਦਿਤੇ Criteria ਨੂੰ ਪੂਰਾ ਕਰਨ ਤੋਂ ਬਾਦ ਟੇਬਲਾਂ ਦੇ ਵਿਚ ਸੰਬਧ ਸਥਾਪਤ ਕਰਨ ਲਈ ਹੇਠ ਲਿਖੇ steps ਦਾ ਅਨੁਸਰਨ ਕਰੋ :-

1. ਡਾਟਾ ਬੇਸ ਵਿੰਡੋਜ਼ ਵਿਚ ਉਪਰ ਵੱਲ Tools ਤੇ ਕਲਿੱਕ ਕਰੋ।
2. ਜਿਹੜੇ ਟੇਬਲਾਂ ਨੂੰ ਤੁਸੀਂ ਆਪਸ ਵਿਚ ਲਿੰਕ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹੋ ਉਨਾਂ ਨੂੰ select ਕਰੋ ਤੇ Add ਬਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।
3. Parent ਟੇਬਲ ਦੀ ਪ੍ਰਾਇਮਰੀ (Primary) Key ਨੂੰ (Teacher table) ਕੇਸ (case) ਵਿੱਚ ਡਰੈਗ ਕਰੋ ਅਤੇ Child table ਦੀ ਉਸ Key ਤੇ ਛੱਡੋ (ਇਸ Case ਵਿਚ Student table)



ਜਦੋਂ ਸੰਬਧਤ ਭਾਗਾਂ ਦੇ check box ਨੂੰ ਜੋੜਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਮੁਖ ਟੇਬਲ ਦੀ ਪ੍ਰਾਇਮਰੀ-ਕੀ (primary key) ਦਾ ਬਾਕੀ ਮੁੱਲ, ਸੰਬਧਿਤ ਸਾਰੀਆਂ ਫੀਲਡਸ (fields) ਨੂੰ ਅਪਣੇ ਆਪ update ਕਰ ਦੇਵੇਗਾ।

ਜਿਦੋ ਸੰਬਧਤ ਭਾਗਾਂ ਦੇ ਰਿਕਾਰਡਸ (Records) ਨੂੰ ਮੁੱਖ ਟੇਬਲ ਵਿਚ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਸੰਬਧਿਤ ਸਾਰੇ ਰਿਕਾਰਡਸ (Records) ਦੂਜੇ ਟੇਬਲਸ ਵਿਚ ਆਪਣੇ ਆਪ ਡਿਲੀਟ (Delete) ਹੋ ਜਾਣਗੇ ।

Enforce Referential Integrity ਦੀ ਚੋਣ ਕਰੋ

- ਜਦੋਂ ਤੁਸੀਂ "Cascade Update Related Field Box" ਨੂੰ ਸਹੀ ਕਰਦੇ ਹੋ, ਅਤੇ parent ਟੇਬਲ ਵਿਚ ਪ੍ਰਾਇਮਰੀ (Primary) Key ਦੇ ਮੁੱਲ ਨੂੰ ਬਦਲਦੇ ਹਾਂ ਤਾਂ ਇਹ child ਟੇਬਲ ਵਿਚ ਉਸ ਮੁੱਲ ਨੂੰ ਅੰਕਿਤ ਕਰ ਦਿੰਦਾ ਹੈ।
- ਜਦੋਂ "Cascade Delete Related Records" ਦੇ check box ਨੂੰ ਸਹੀ ਕਰਦੇ ਹਾਂ ਤਾਂ ਉਹ child ਟੇਬਲ ਵਿਚ ਸੰਬੰਧਤ ਰਿਕਾਰਡਜ਼ (Records) ਨੂੰ ਡਿਲੀਟ (Delete) ਕਰ ਦਿੰਦਾ ਹੈ।
- ਕਲਿੱਕ Create ਅਤੇ ਸੇਵ Relationship

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

1.4.2 ਅਸੈਸ ਡਾਟਾਬੇਸ ਦੇ ਆਬਜੈਕਟਸ (Access Database of Objects) :

ਅਸੈਸ ਨੂੰ ਵਰਤਣ ਤੋਂ ਪਹਿਲਾਂ ਸਾਨੂੰ ਕੁਝ ਬੇਸਿਕ concepts ਨੂੰ ਸਮਝ ਲੈਣਾ ਬਹੁਤ ਜਰੂਰੀ ਹੈ। ਇਕ ਡਾਟਾ-ਬੇਸ (Database) ਕੁਝ ਵਿਸ਼ਿਆਂ ਦੀ ਜਾਣਕਾਰੀ ਦਾ ਸਮੂਹ ਹੈ। ਡਾਟਾ-ਬੇਸ ਸਾਨੂੰ, ਇਸ ਸੂਚਨਾ ਨੂੰ ਇਕ ਤਰਤੀਬ ਵਾਰ ਢੰਗ ਵਿਚ ਸਮਝਣ ਵਿਚ ਮਦਦ ਕਰੇਗਾ। ਅਸੈਸ ਵਿਚ ਡਾਟਾ-ਬੇਸ ਸਾਰੇ ਸੰਬਧਤ ਆਬਜੈਕਟਸ (objects) ਨੂੰ ਇਕੱਠਾ ਕਰਕੇ ਰਖਦਾ ਹੈ। ਅਸੈਸ ਵਿਚ ਮੁੱਖ ਤੌਰ ਤੇ ਪ੍ਰਯੋਗ ਹੋਣ ਵਾਲੇ 7 ਆਬਜੈਕਟਸ (objects) ਵਿਸ਼ੇ ਹਨ : Tables, Queries, Forms, Records, Pages, macros ਅਤੇ modules ਹਾਲਾਂ ਕਿ ਕੁਝ ਹੋਰ ਕੰਪਿਊਟਰ ਡਾਟਾ-ਬੇਸ ਪ੍ਰੋਗਰਾਮ ਵੀ ਆਬਜੈਕਟਸ (objects) ਨੂੰ ਬੁਲਾ ਸਕਦੇ ਹਨ ਜਿਹੜੇ ਕਿ ਅਮਲ ਵਿਚ ਡਾਟਾ ਰਖਦੇ ਹਨ; ਅਸੈਸ ਵਿਚ ਇਸ ਆਬਜੈਕਟ (object) ਨੂੰ ਟੇਬਲ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਅਸੈਸ ਇਕ ਸਮੇਂ ਇਕ ਹੀ ਡਾਟਾ-ਬੇਸ ਕੋਲ ਕੰਮ ਕਰ ਸਕਦੀ ਹੈ ਪਰ ਉਸ ਡਾਟਾਬੇਸ ਵਿਚ ਸੈਂਕੜੇ ਆਬਜੈਕਟਸ ਹੋ ਸਕਦੇ ਹਨ ਜਿਵੇਂ ਕਿ ਟੇਬਲ, Queries ਅਤੇ ਫਾਰਮ ਆਦਿ। ਇਹ ਸਾਰੇ ਇਕੋ ਅਸੈਸ (Access) ਦੀ ਫਾਇਲ ਵਿਚ ਰੱਖੇ ਜਾਂਦੇ ਹਨ। ਅਸੈਸ ਦੇ ਡਾਟਾ-ਬੇਸ ਦਾ ਦਿਲ ਟੇਬਲ ਹੈ। ਅਸੈਸ ਵਿਚ ਡਾਟਾ-ਬੇਸ ਦੀ ਫਾਈਲ ਨੂੰ .MDB ਦੀ extension ਨਾਲ ਸੇਵ (Save) ਕਿੱਤਾ ਜਾਂਦਾ ਹੈ ਜਿਹੜਾ ਕਿ ਡਾਟਾ-ਬੇਸ ਵਿੰਡੋਜ਼ ਵਿਚ ਪ੍ਰਦਰਸ਼ਿਤ ਹੁੰਦਾ ਹੈ।

ਟੇਬਲ (Table): ਟੇਬਲ ਡਾਟਾ ਨੂੰ ਡਾਟਾ-ਬੇਸ ਵਿਚ ਸਟੋਰ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਅੱਗੇ ਟੇਬਲ ਡਾਟਾ ਨੂੰ ਰੋ ਅਤੇ ਕਾਲਮ ਵਿਚ ਵੰਡਦਾ ਹੈ। ਜਦੋਂ ਅਸੈਸ ਵਿਚ ਡਾਟਾ-ਬੇਸ ਨੂੰ ਖੋਲ੍ਹਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਸਭ ਤੇ ਪਹਿਲਾਂ ਟੇਬਲ ਹੀ ਖੁਲਦਾ ਹੈ।

Queries: Query ਨੂੰ ਡਾਟਾ-ਬੇਸ ਵਿਚੋ ਨਿਸ਼ਚਿਤ ਸੂਚਨਾ ਨੂੰ ਲੈਣ ਵਾਸਤੇ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। Query ਇਕ ਗਰੁੱਪ ਦੇ ਆਧਾਰ ਤੇ ਵੀ ਰਿਕਾਰਡਸ (Records) ਨੂੰ ਕੱਢ ਕੇ ਲਿਆ ਸਕਦੀ ਹੈ ਜਿਹੜੇ ਇਕ ਨਿਸਚਿਤ ਸ਼ਰਤਾਂ ਨੂੰ ਪੂਰੀ ਕਰਦੇ ਹੋਣ। ਫਾਰਮਸ (Forms) ਇਹਨਾਂ ਸੂਚਨਾਵਾਂ ਨੂੰ ਸਕਰੀਨ ਦੇ ਉੱਪਰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਰਿਪੋਰਟਸ ਇਹਨਾਂ ਰਿਕਾਰਡਸ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਵਰਤਿਆਂ ਜਾਂਦੀਆ ਹਨ। Queries ਟੇਬਲਸ ਤੇ ਜਾ ਦੂਜੀਆਂ Queries ਤੇ ਅਧਾਰਿਤ ਹੁੰਦੀਆਂ ਹਨ। Queries, Records ਨੂੰ ਇਕ ਡਾਟਾ-ਬੇਸ ਵਿਚ Select, change, add ਜਾਂ Delete ਕਰਨ ਲਈ ਵਰਤਿਆਂ ਜਾਂਦੀਆ ਹਨ।

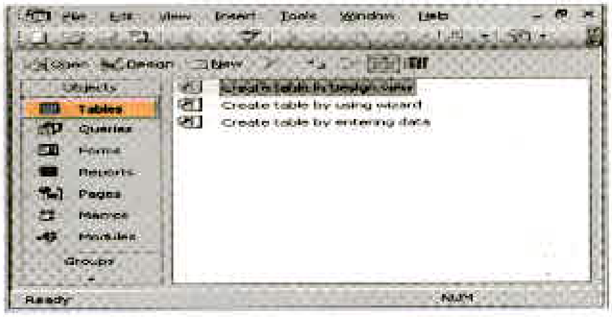
ਫਾਰਮ (Form): ਫਾਰਮਾਂ ਨੂੰ ਕਈ ਤਰਾਂ ਨਾਲ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ ਪਰ ਇਸ ਨੂੰ ਸਭ ਤੋਂ ਵੱਧ ਟੇਬਲਸ (tables) ਭਰਨ ਅਤੇ ਪ੍ਰਦਰਸਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਡਾਟਾ ਭਰਨ ਵਾਲੇ ਫਾਰਮ ਇਕ ਯੂਜ਼ਰ ਨੂੰ ਟੇਬਲਸ ਵਿਚ ਸਹੀ, ਛੇਤੀ ਤੇ ਆਸਾਨੀ ਨਾਲ ਡਾਟਾ ਭਰਨ ਵਿਚ ਮਦਦ ਕਰਦੇ ਹਨ। ਫਾਰਮ ਡਾਟਾ ਨੂੰ ਸਹੀ ਢੰਗ ਵਿਚ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਵਿਚ ਮਦਦ ਕਰਦੇ ਹਨ ਜੋ ਕਿ ਇਕ ਆਮ ਟੇਬਲ ਨਾਲ ਸੰਭਵ ਨਹੀਂ। ਤੁਸੀ ਇਕ ਫਾਰਮ ਰਾਹੀ ਟੇਬਲਸ ਵਿਚ ਡਾਟਾ ਨੂੰ add, delete ਤੇ ਵੰਡ ਸਕਦੇ ਹੋ। ਫਾਰਮਸ ਨੂੰ ਟੇਬਲਸ ਵਿਚ ਨਿਸਚਿਤ ਸੂਚਨਾ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਨ ਲਈ ਸੁਚਜੇ ਢੰਗ ਨਾਲ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ।

ਰਿਪੋਰਟ (Report): ਰਿਪੋਰਟਾਂ ਸਾਡੇ ਦੁਆਰਾ ਚੁਣੇ ਗਏ ਡਾਟਾ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਦੀਆ ਹਨ। ਰਿਪੋਰਟ ਟੇਬਲਾ ਤੋਂ ਡਾਟਾ ਲੈਂਦਿਆ ਹਨ ਜਿਹੜੀ ਕਿ Query ਤੇ ਅਧਾਰਿਤ ਹੁੰਦੀਆ ਹਨ ਤੇ ਕੇਵਲ ਉਪਯੁਕਤ ਸੂਚਨਾ ਹੀ ਪ੍ਰਦਾਨ ਕਰਦੀਆਂ ਹਨ। ਰਿਪੋਰਟਾਂ ਇਕ ਤੋਂ ਵੱਧ ਟੇਬਲਸ ਤੇ ਵੀ ਅਧਾਰਿਤ ਹੋ ਸਕਦੀਆ ਹਨ ਅਤੇ Queries ਤੇ ਵੀ ਜੋ ਡਾਟਾ ਵਿਚ ਗੁੰਝਲਦਾਰ ਸੰਬੰਧਾਂ ਨੂੰ ਦਰਸਾਉਂਦੀਆ ਹਨ। ਅਸੈਸ (Access) ਵਿਚ ਕਾਫੀ ਰਿਪੋਰਟਸ ਪਹਿਲਾ ਤੋਂ ਹੀ ਮੌਜੂਦ ਹਨ ਜੋ ਆਸਾਨੀ ਨਾਲ ਡਾਟਾ ਨੂੰ ਮਨ ਚਾਹੀ ਰੂਪ ਵਿਚ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰ ਸਕਦੀਆ ਹਨ।

ਪੇਜ (Page): Access 2000 ਵਿਚ ਪੇਜ਼ਿਸ ਨੂੰ ਨਵੇਂ ਵਿਸ਼ੇ ਵਜੋਂ ਮਿਲਾਇਆ ਗਿਆ ਸੀ ਜਿਹਨਾਂ ਨੂੰ ਡਾਟਾ ਅਸੈਸ ਪੇਜਿਜ਼ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਪੇਜਿਜ਼, HTML ਦੇ ਡਾਕੂਮੈਂਟਸ (ਦਸਤਾਵੇਜ) ਹਨ ਜੋ ਕਿ ਡਾਟਾਬੇਸ ਨਾਲ ਜੋੜੇ ਗਏ ਹਨ। ਇਹ ਦਸਤਾਵੇਜ਼ ਅਸੈਸ ਦੇ ਫਾਰਮਾਂ ਦੀ ਤਰ੍ਹਾਂ ਹੀ ਹੁੰਦੇ ਹਨ, ਪ੍ਰੰਤੂ Internet Explorer ਵਿਚ ਦੇਖੇ ਜਾ ਸਕਦੇ ਹਨ। ਇਕ ਵੱਡੀ ਭਿੰਨਤਾ ਜਿਹੜੀ ਕਿ ਡਾਟਾ ਅਸੈਸ ਪੇਜਿਸ ਤੇ ਫਾਰਮਸ ਵਿਚ ਹੈ ਉਹ ਇਹ ਹੈ ਕਿ ਪੇਜਿਸ ਨੂੰ ਇਕ ਅਲਗ ਫਾਇਲ ਵਿਚ ਸੇਵ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜਦਕਿ ਫਾਰਮਸ (Forms) ਡਾਟਾ-ਬੇਸ ਦੀ ਫਾਈਲ ਵਿਚ ਸੇਵ ਹੁੰਦੇ ਹਨ। ਇਹ ਇਸ ਕਰਕੇ ਹੈ ਕਿਉਂਕਿ ਪੇਜਿਜ਼ Internet Explorer ਵਿਚ ਖੁਲ੍ਹਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

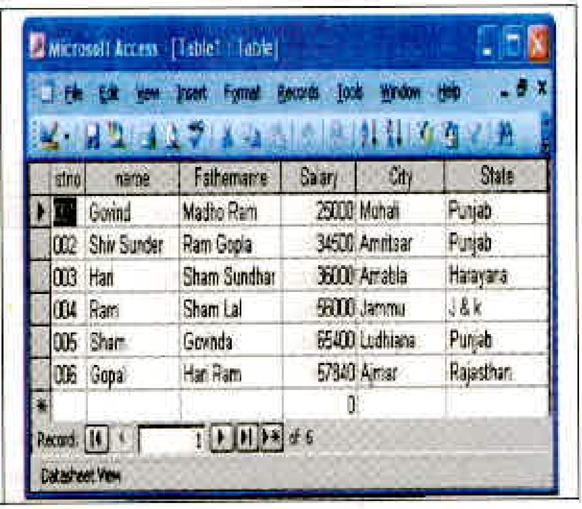


ਇਥੇ ਵਿੰਡੋ ਵਿਚ ਖੱਬੇ ਪਾਸੇ ਕਾਫੀ ਤਰਾਂ ਦੇ ਆਬਜੈਕਟਸ (Objects) ਦੇਖੇ ਜਾ ਸਕਦੇ ਹਨ ਜਿਵੇ ਕਿ ਟੇਬਲਜ਼ (Tables), Queries ਅਤੇ ਫਾਰਮਜ਼ (Forms) ਆਦਿ ਅਤੇ ਸੱਜੇ ਪਾਸੇ ਉਹ ਹੋਵੇਗਾ ਜੋ ਅਸੀਂ ਖੱਬੇ ਪਾਸੇ Select ਕਰਦੇ ਹਾਂ।ਅਸੈਸ (Access) ਇਸ ਤਰਾਂ ਦੇ ਨਵੇਂ Objects ਨੂੰ ਬਣਾਉਣ ਵਾਸਤੇ ਪ੍ਰੇਰਿਤ ਕਰਦੀ ਹੈ।

ਆਓ ਅਸੈਸ (Access) ਟੇਬਲ ਅਤੇ ਅਸੈਸ ਫਾਰਮ ਨੂੰ ਪ੍ਰਯੋਗ ਕਰਦੇ ਹੋਏ ਡਾਟਾ Entry ਕਰਨ ਵਾਲੇ ਇਕ ਫਾਰਮ ਦੀ ਸਕਰੀਨ ਬਣਾਈਏ। ਤੁਸੀ Wizard ਜਾਂ ਡਿਜ਼ਾਇਨ ਦਾ ਪ੍ਰਯੋਗ ਕਰਦੇ ਹੋਏ ਇਕ ਟੇਬਲ ਬਣਾ ਸਕਦੇ ਹੋ। ਹੇਠ ਲਿਖੇ ਕੱਦਮ (Steps) ਪ੍ਰਯੋਗ ਕੀਤੇ ਜਾਂਦੇ ਹਨ।

Step 1: ਇਕ ਡਾਟਾ-ਬੇਸ db1 ਬਣਾਓ (ਇਹ ਡਾਟਾਬੇਸ ਇਕ ਫੋਲਡਰ ਦੀ ਤਰਾਂ ਹੋਵੇਗਾ ਜਿਸ ਵਿਚ ਅਸੀ ਇਸਦੇ ਅਲਗ-2 ਭਾਗਾਂ ਨੂੰ ਸਟੋਰ ਕਰ ਸਕਦੇ ਹਾਂ)

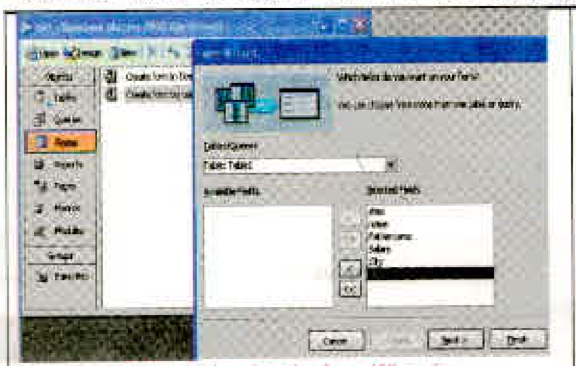
Step 2: ਟੇਬਲ 1 ਦਾ ਸਟਰਕਚਰ (Structure) ਬਣਾਓ ਅਤੇ ਜਿਸ ਵਿਚ ਹੇਠ ਦਰਸਾਏ ਰਿਕਾਰਡਸ ਹੋਣ।



stno	name	Fathername	Salary	City	State
001	Govind	Madho Ram	25000	Mohali	Punjab
002	Shiv Sunder	Ram Gopla	34500	Amritsar	Punjab
003	Hari	Sham Sundhar	36000	Ambala	Harayana
004	Ram	Sham Lal	56000	Jammu	J & k
005	Sham	Govinda	65400	Ludhiana	Punjab
006	Gopal	Hari Ram	57840	Ajmar	Rajasthan
			0		

ਹੁਣ ਇਸ ਡਾਟਾ ਨੂੰ ਪ੍ਰਯੋਗ ਕਰਦੇ ਹੋਏ ਇਕ ਡਾਟਾ Entry Screen ਬਣਾਈਏ।

Step 3: ਹੇਠਾਂ ਦਰਸਾਈ ਗਈ ਡਾਟਾਬੇਸ ਸਕਰੀਨ ਵਿਚ ਫਾਰਮ ਨੂੰ ਚੁਣੋ ਅਤੇ ਫਿਰ ਵਿਜ਼ਾਰਡ ਦਬਾਓ, ਫਿਰ ਜਿਹੜੇ ਫੀਲਡ ਮੌਜੂਦ ਹਨ, ਨੂੰ ਚੁਣੋ ਜਿਸ ਵਿਚ ਤੁਸੀ ਡੇਟੇ ਨੂੰ ਸਕਰੀਨ ਤੇ ਦਿਖਾਣਾ ਚਾਹੁੰਦੇ ਹੋ ਤੇ ਉਸ ਤੇ ਬਾਦ ਹੇਠਾਂ ਦਿੱਤੀ ਕਿਸਮ ਦੇ ਅਨੁਸਾਰ ਫਾਰਮ ਦਾ ਸਟਾਇਲ (Style) ਚੁਣੋ (ਇਥੇ ਅਸੀ ਆਮ Style ਚੁਣਿਆ ਹੈ)



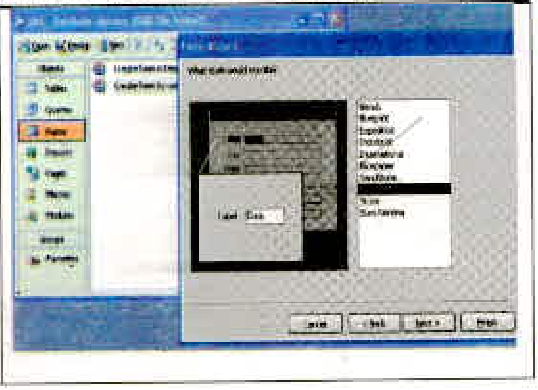


Figure: showing the form Wizards

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੈੱਟਪ 4 (Step 4) : ਉਸ ਤੋ ਬਾਦ ਫਾਰਮ ਦਾ ਵਿਜ਼ਾਰਡ (Wizard) ਤੁਹਾਨੂੰ ਫਾਰਮ ਦਾ ਸਿਰਲੇਖ (Title) ਲਿਖਣ ਲਈ ਕਹੇਗਾ। ਅਸੀ ਇਥੇ ਡਾਟਾ ਐਂਟਰੀ ਸਕਰੀਨ ਫਰਮ (firm) ਦਾ ਨਾਂ ਰਖਿਆ ਹੈ। ਫੇਰ Check ਬੱਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ। ਫਾਰਮ ਵਿਊ ਡਾਟਾ ਪਾਉਣ ਲਈ ਜਾਂ ਵੇਖਣ ਲਈ ਖੋਲ੍ਹੋ ਅਤੇ ਫਿਰ ਨਤੀਜੇ ਲਈ Finish ਬੱਟਨ ਤੇ ਕਲਿੱਕ ਕਰੋ।

1.5.0 ਪ੍ਰੋਗਰਾਮਿੰਗ ਕੰਨਸੈਪਟ ਦੀ ਦੁਹਰਾਈ

ਕੰਪਿਊਟਰ ਇਕ ਬਹੁਤ ਉਪਯੋਗੀ ਵਰਤੋਯੋਗ ਮਸ਼ੀਨ ਹੈ, ਜਿਸ ਨਾਲ ਅਸੀ ਮੁਸ਼ਕਿਲ ਤੋਂ ਮੁਸ਼ਕਿਲ ਕੰਮ ਨੂੰ ਬਹੁਤ ਤੇਜ਼, ਸਹੀ ਅਤੇ ਅਸਾਨੀ ਨਾਲ ਕਰ ਸਕਦੇ ਹਾਂ। ਕੰਪਿਊਟਰ ਦੀ ਮੱਦਦ ਨਾਲ ਅਸੀਂ ਚਿੱਤਰ ਬਣਾ ਸਕਦੇ ਹਾਂ, ਚਿੱਤਰਾਂ ਦੀ ਸੋਧ ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਭਿੰਨ-ਭਿੰਨ ਪ੍ਰਕਾਰ ਦੀਆਂ ਗੇਮਾ ਖੇਡ ਸਕਦੇ ਹਾਂ ਆਦਿ-ਆਦਿ। ਇਹ ਸਾਰੇ ਕਾਰਜ ਕਰਨ ਲਈ ਅਲੱਗ-ਅਲੱਗ ਤਰ੍ਹਾਂ ਦੇ ਪ੍ਰੋਗਰਾਮਾਂ ਦੀ ਮੱਦਦ ਲੈਣੀ ਪੈਂਦੀ ਹੈ।

ਪਹਿਲਾਂ, ਕਿਸੇ ਕੰਮ ਜਾਂ ਸਮੱਸਿਆ ਨੂੰ ਹਲ ਕਰਨ ਲਈ ਉਸ ਸਮੱਸਿਆ ਦੀ ਰੂਪ-ਰੇਖਾ ਅਤੇ ਉਸ ਨੂੰ ਸਮਝਣਾ ਜਰੂਰੀ ਹੁੰਦਾ ਹੈ। ਇਹ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਕਿਸ ਆਊਟਪੁੱਟ ਦੀ ਜਰੂਰਤ ਹੈ ਅਤੇ ਇਸ ਲਈ ਕਿਹੜੀ ਇਨਪੁੱਟ ਦਿੱਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ। ਦੂਸਰਾ, ਸਮੱਸਿਆ ਨੂੰ ਹਲ ਕਰਨ ਲਈ ਕਿਸ method ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਣੀ ਹੈ।

ਇਸ ਤੋ ਬਾਦ ਸਾਰੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਰੂਪ-ਰੇਖਾ ਤਿਆਰ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਅਤੇ ਫਿਰ ਅਸਲ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਅੰਤ ਵਿਚ ਲਿਖੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਪਰਖ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਤਾਂ ਕਿ ਇਹ ਸਹੀ ਨਤੀਜੇ ਦੇ ਸਕੇ ।

1.5.1 ਪ੍ਰੋਗਰਾਮ development ਦੇ ਵੱਖ-ਵੱਖ ਪੜਾਅ ਹੇਠ ਲਿਖੇ ਹਨ :-

- **ਸੱਮਸਿਆ ਦਾ ਵਿਸ਼ਲੇਸ਼ਣ**
- **ਸਮਾਧਾਨ ਦਾ ਵਿਕਾਸ**
- **ਸਮਾਧਾਨ ਕੋਡਿੰਗ**
- **ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਟੈਸਟ ਕਰਨਾ**

1. **Analyzing the Problem** ਦਾ ਮਤਲਬ ਹੈ ਕਿ ਪ੍ਰੋਗਰਾਮ ਦੀਆਂ ਜਰੂਰਤਾਂ ਕਿਹੜੀਆਂ ਹਨ। ਦੂਸਰੇ ਸ਼ਬਦਾਂ ਵਿਚ, ਉਪਭੋਗਤਾ ਦੀਆਂ ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕੀ-2 ਜਰੂਰਤਾਂ ਹਨ।

   ਪ੍ਰੋਗਰਾਮ ਦੀ ਇਨਪੁਟ ਤੇ ਆਊਟਪੁਟ ਤੋਂ ਇਲਾਵਾ ਇਹ ਫੈਸਲਾ ਕਰਨਾ ਵੀ ਜ਼ਰੂਰੀ ਹੈ ਕਿ ਵੇਰਿਏਬਲਸ (variables) ਅਤੇ ਉਹਨਾਂ ਦੀ ਟਾਈਪ ਕੀ ਹੋਵੇਗੀ।

   ਇਸ ਦੇ ਨਾਲ ਇਕ ਢੁਕਵੇਂ ਲੋਜਿਕਲ (logical) ਪ੍ਰੋਸੈਸ ਦਾ ਹੋਣਾ ਵੀ ਜਰੂਰੀ ਹੈ ਜੋ ਕਿ ਸਾਨੂੰ ਜਰੂਰਤਮੰਦ ਆਊਟਪੁਟ ਦੀ ਰੂਪ-ਰੇਖਾ ਪੇਸ਼ ਕਰ ਸਕੇ।

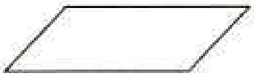
2. ਪ੍ਰੋਗਰਾਮ ਦੀ ਸੱਮਸਿਆ ਦਾ ਹੱਲ ਕਰਨ ਲਈ ਸਹੀ ਲੋਜਿਕ (logic) ਦਾ ਹੋਣਾ ਜਰੂਰੀ ਹੈ ਇਸ ਲੋਜਿਕ ਨੂੰ ਬਨਾਉਣ ਲਈ ਐਲਗੋਰਿਥਮ **(algorithm)** ਦਾ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜੋ ਕਿ ਸਰਲ ਇੰਗਲਿਸ਼ ਭਾਸ਼ਾ ਵਿਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਫਿਰ ਇਸ ਐਲਗੋਰਿਥਮ (algorithm) ਨੂੰ ਫਲੋ ਚਾਰਟ (flow chart) ਵਿਚ ਬਦਲ ਦਿਤਾ ਜਾਂਦਾ ਹੈ। ਸੂਡੋ ਕੋਡ (pseudo code) ਵੀ ਐਲਗੋਰਿਥਮ ਤੇ ਇਲਾਵਾ ਲੋਜਿਕਲ ਨੂੰ ਉਸਾਰਣ ਵਿਚ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਕਿਉਂਕਿ ਫਲੋ ਚਾਰਟ ਇਕ ਗਰਾਫਿਕਲ ਰੂਪਾਂਤਰ (graphical format) ਹੈ, ਇਸ ਵਿਚ ਵੱਖ-ਵੱਖ ਚਿੰਨ੍ਹਾਂ (symbols) ਦੇ ਦੁਆਰਾ ਕਿਸੇ ਵੀ ਕਾਰਜ ਨੂੰ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਚਿੰਨ੍ਹ ਅਗੇ ਦਰਸਾਏ ਗਏ ਹਨ :

Downloaded from https:// www.studiestoday.com

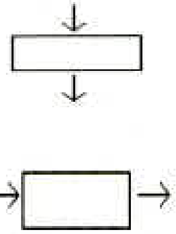
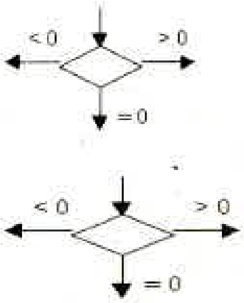
Downloaded from https:// www.studiestoday.com

ਇਨਪੁਟ / ਆਊਟਪੁਟ ਬਾਕਸ (I/O Box)

ਇਹ ਸਮਾਂਤਰ ਚਤੁਰਭੁਜ (parallelogram) ਦੇ ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ।



ਡਾਟਾ ਫਲੋ ਲਾਈਨਜ਼ (Data Flow Lines)

ਪ੍ਰੋਸੈਸਿੰਗ (Processing) ਇਹ ਆਇਤਾਕਾਰ (rectangular) ਦੁਆਰਾ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਕੋਈ ਵੀ ਡਾਟਾ ਪ੍ਰੋਸੈਸ ਕਰਨਾ ਹੁੰਦਾ ਹੈ ਤਾਂ ਇਸ ਦੇ ਵਿਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਸ **ਪ੍ਰੋਸੈਸਿੰਗ** ਚਿੰਨ੍ਹ ਵਿਚੋਂ ਕੇਵਲ ਇਕ ਫਲੋ ਲਾਈਨ ਹੀ ਬਾਹਰ ਆਉਣੀ ਚਾਹੀਦੀ ਹੈ। ਇਹ ਐਰੋਜ਼ ਦੇ ਨਾਲ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ। ਜੋ ਕਿ ਸਾਮਣੇ ਚਿੱਤਰ ਵਿਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ।	
ਡਿਸਿਜ਼ਨ ਬਾਕਸ (Decision Box) : ਇਹ ਡਾਇਮੰਡ (diamond) ਦੇ ਆਕਾਰ ਦਾ ਹੁੰਦਾ ਹੈ। ਇਸ ਵਿਚ ਕੇਵਲ ਇਕ ਹੀ ਫਲੋ ਲਾਂਈਨ ਡਿਸਿਜ਼ਨ ਚਿੰਨ੍ਹ ਅੰਦਰ ਪ੍ਰਵੇਸ਼ ਕਰਦੀ ਹੈ ਜਦ ਕਿ ਦੋ ਜਾਂ ਤਿੰਨ ਫਲੋ ਲਾਈਨਜ਼ ਹਰ ਮੁਮਕਿਨ ਉੱਤਰ ਲਈ ਇਕ ਫਲੋ ਲਾਈਨ ਇਸ ਡਿਸਿਜ਼ਨ ਚਿੰਨ੍ਹ ਤੋ ਬਾਹਰ ਨਿਕਲਦੀਆਂ ਹਨ ਜਿਵੇਂ ਕਿ ਸਾਹਮਣੇ ਚਿੱਤਰ ਵਿਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ।	
ਟਰਮੀਨਲ (Terminal) ਚਿੰਨ੍ਹ ਦੇ ਨਾਲ ਕੇਵਲ ਇਕ ਹੀ ਫਲੋ ਲਾਂਈਨ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।	

ਅਸੀ ਹੁਣ ਫਲੋ ਚਾਰਟ (flow chart) ਨੂੰ ਇਸਤੇਮਾਲ ਕਰਦੇ ਹੋਏ 3 ਅੰਕਾਂ ਵਿਚ ਵੱਡੇ ਅੰਕ ਨੂੰ ਲੱਭਾਂਗੇ ਜੋ ਕਿ ਹੇਠਾਂ ਮਿਸਾਲ ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ

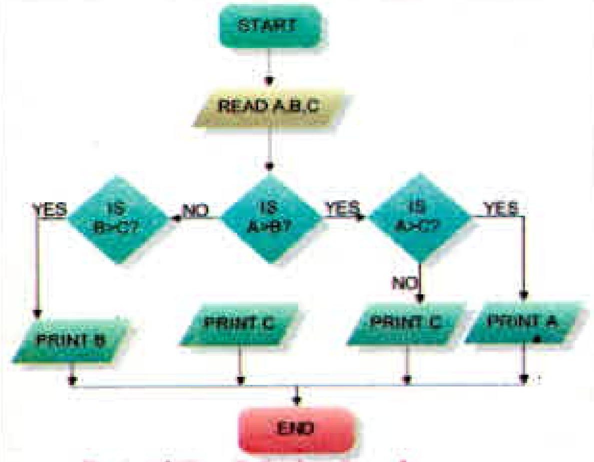


Fig a : ਤਿੰਨ ਨੰਬਰਾਂ ਵਿੱਚੋਂ ਸਭ ਤੋਂ ਵੱਡਾ ਨੰਬਰ ਲੱਭਣ ਲਈ

To find largest of three numbers

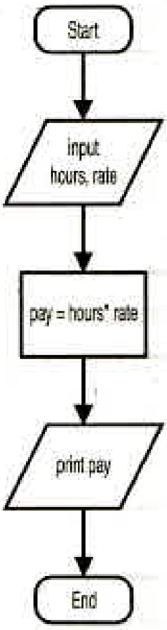
1)Start
2)Read 3 numbers: num1, num2, num3
3)if num1 > num2 then go to step 5
4)if num2 > num3 then
  print num2 is largest
  else
  print num3 is largest
  goto step 6
5)if num1 > num3 then
  print num1 is largest
  else
  print num3 is largest
6)end.

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Introductory Examples of Flowcharts and Pseudocode

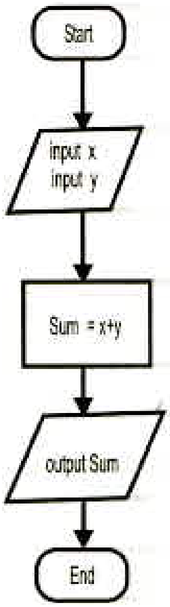
Calculate Pay - sequence




```
Begin
  input hours
  input rate
  pay = hours* rate
  print pay
End
```

Fig. b

Sum of 2Numbers - sequence




```
Begin
  input x,y
  Sum = x+y
  print sum
End
```

Fig. c

Fig. b ਵਿਚ ਤਨਖਾਹ ਦੀ ਗਣਨਾ ਕੀਤੀ ਗਈ ਹੈ ਅਤੇ Fig. c ਵਿਚ 2 ਅੰਕਾਂ ਦਾ ਜੋੜ ਕੀਤਾ ਗਿਆ ਹੈ।

3. ਪ੍ਰੋਗਰਾਮ **ਡਿਵਲਪਮੈਂਟ ਸਾਈਕਲ** ਵਿਚ ਤੀਸਰਾ ਪੜਾਵ, ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਲਿਖਣਾ ਜਾਂ ਕੋਡਿੰਗ ਕਰਨਾ ਹੁੰਦਾ ਹੈ। ਫਲੋ ਚਾਰਟ (flow chart) ਅਤੇ ਸੂਡੋ (pseudo) ਕੋਡ ਦੇ ਆਧਾਰ ਤੇ ਪ੍ਰੋਗਰਾਮਰ, ਪ੍ਰੋਗਰਾਮ ਲਿਖਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਲਿਖਣ ਤੋਂ ਪਹਿਲਾਂ ਪ੍ਰੋਗਰਾਮਰ ਇਹ ਯਕੀਨੀ ਬਣਾਉਂਦਾ ਹੈ ਕਿ ਸਬੰਧਿਤ ਕੰਪਿਊਟਰ ਉੱਪਰ ਉਚਿਤ ਸਾਫਟਵੇਅਰ ਉਪਲਬਧ ਹੈ ਜਾਂ ਨਹੀਂ। ਮਿਸਾਲ ਲਈ, 'C' ਭਾਸ਼ਾ ਵਿਚ ਪ੍ਰੋਗਰਾਮ ਲਿਖਣ ਤੋਂ ਪਹਿਲਾਂ ਪ੍ਰੋਗਰਾਮਰ ਇਹ ਜਰੂਰੀ ਬਣਾਏ ਕਿ "ਟਰਬੋ ਸੀ" (turbo C) ਜਾਂ "ਬੋਰਲੈਂਡ ਸੀ" (Borland C) ਸਾਫਟਵੇਅਰ, ਜਿਸਨੂੰ ਪ੍ਰੋਗਰਾਮ ਬਣਾਉਣ ਲਈ ਵਰਤਣਾ ਹੈ ਉਹ ਕੰਪਿਊਟਰ ਤੇ ਸਥਾਪਿਤ ਹੈ ਜਾਂ ਨਹੀਂ।

4. ਚੌਥਾ ਪੜਾਵ – ਪ੍ਰੋਗਰਾਮ ਪਰਖ (program testing)

ਪ੍ਰੋਗਰਾਮ ਬਣਾਉਣ ਤੋਂ ਬਾਅਦ ਇਹ ਯਕੀਨੀ ਬਣਾਇਆ ਜਾਂਦਾ ਹੈ ਕਿ ਪ੍ਰੋਗਰਾਮ ਗਲਤੀਆਂ ਤੋਂ ਮੁਕਤ ਹੈ ਅਤੇ ਦਿੱਤੀ ਗਈ ਸਮੱਸਿਆ ਦਾ ਹੱਲ ਕਰਨ ਲਈ ਕਾਬਲ ਹੈ ਜਾਂ ਨਹੀ। ਕਈ ਵਾਰ ਲਿਖਿਆ ਗਿਆ ਪ੍ਰੋਗਰਾਮ ਕੰਪਾਈਲ ਟਾਈਮ (compile time) ਜਾਂ ਰਨ ਟਾਈਮ (run time) ਗਲਤੀਆਂ ਨੂੰ ਨਹੀਂ ਦਰਸਾਉਂਦਾ ਪਰ ਉਹ ਫਿਰ ਵੀ ਇੱਛਕ ਆਊਟਪੁਟ ਨਹੀਂ ਦਰਸਾਉਂਦਾ। ਇਹਨਾਂ ਹਾਲਾਤਾਂ ਵਿਚ ਫਿਰ ਸਾਰੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਮੁਕੰਮਲ ਪਰਖ ਅਤੇ ਸਾਰੇ ਪੜਾਵਾਂ ਦੀ ਮੁੜ ਤੋਂ ਪੜਤਾਲ (review) ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ਤਾਂ ਕਿ ਗਲਤੀ ਜੇ ਕਰ ਕੋਈ ਹੈ ਦਾ ਪਤਾ ਲਗ ਸਕੇ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

1.5.2 ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਦੇ ਐਲੀਮੈਂਟਸ (Elements of Programing Language)

ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਕੀ ਹੈ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਦੇ ਅਲੱਗ-ਅਲੱਗ development cycle ਬਣਾਉਣ ਤੋਂ ਬਾਅਦ ਅਸੀਂ ਹੁਣ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ (Programing Language) ਦੇ ਅਲੱਗ-ਅਲੱਗ ਅੰਸ਼ਾਂ (elements) ਵੱਲ ਰੋਸ਼ਨੀ ਪਾਉਂਦੇ ਹਾਂ। ਵੈਰੀਏਬਲ (variables) ਵਿਚ ਅਸੀਂ ਸਟਰਿੰਗ (string) ਜਾਂ ਨਿਊਮੈਰਿਕ (numeric) ਡਾਟਾ ਨੂੰ ਸਟੋਰ ਕਰਦੇ ਹਾਂ ਅਤੇ ਇਸ ਡਾਟਾ ਨੂੰ ਜਾਨਣ ਲਈ ਅਸੀਂ ਵੈਰੀਏਬਲ (variable) ਦੇ ਨਾਮ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਜਦ ਇਕ ਵਾਰ ਡਾਟਾ ਵੇਰੀਏਬਲ ਵਿਚ ਸਟੋਰ ਹੋ ਜਾਂਦਾ ਹੈ ਅਤੇ ਅਸੀਂ ਇਸ ਡਾਟਾ ਨਾਲ ਕਿਸੇ ਵੀ ਤਰ੍ਹਾਂ ਦੀ ਗਣਿਤਕ ਗਣਨਾ (arithmetical calculation) ਕਰ ਸਕਦੇ ਹਾਂ। ਇਸ ਕੰਮ ਨੂੰ ਸਮਪੂਰਣ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਵਿਧੀਬੱਧ ਤਰੀਕੇ ਨਾਲ ਹਿਦਾਇਤਾਂ ਦਿੱਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਇਹ ਹਿਦਾਇਤਾਂ ਦਰਸਾਂਉਦੀਆਂ ਹਨ ਕਿ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਵਰਤੇ ਗਏ ਵੇਰੀਏਬਲਜ਼ (variables) ਦੇ ਡਾਟਾ ਨੂੰ ਕਿਸ ਤਰ੍ਹਾਂ ਜੋੜ-ਤੋੜ (manipulate) ਕਰਨਾ ਹੈ।

ਕਈ ਵਾਰ ਮੁਸ਼ਕਿਲ ਗਣਨਾਂ (calculations) ਦਾ ਹੱਲ ਲੱਭਣ ਲਈ ਇਹਨਾਂ ਹਿਦਾਇਤਾਂ ਵਿਚ ਉਪਰੇਟਰ ਦਾ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਜਿਵੇਂ ਕਿ ਰਿਲੇਸ਼ਨਲ (relational) ਜਾਂ ਲੋਜਿਕਲ (logical) ਓਪਰੇਟਰਸ ਅਤੇ ਕਈ ਵਾਰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਗਣਨਾ ਕੀਤੇ ਗਏ ਪ੍ਰਣਾਮ ਉਪਰ ਫੈਸਲਾ ਲੈਣ ਦੀ ਜਰੂਰਤ ਹੁੰਦੀ ਹੈ ਜਿਸ ਲਈ ਕੰਟਰੋਲ ਹਿਦਾਇਤਾਂ "statements" ਦਾ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਹਿਦਾਇਤਾਂ ਪ੍ਰੋਗਰਾਮ ਦੇ ਸੰਚਾਰ ਨੂੰ ਕਾਬੂ ਕਰਦੀਆਂ ਹਨ। ਇਕ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਬਹੁਤ ਸਾਰੀਆਂ ਹਿਦਾਇਤਾਂ "code lines" ਹੋ ਸਕਦੀਆਂ ਹਨ। ਇਕੋ ਤਰ੍ਹਾਂ ਦੇ ਹਿਦਾਇਤਾਂ ਦੇ ਸਮੂਹ ਨੂੰ ਜੋ ਕਿ ਇਕ ਖਾਸ ਕਾਰਜ ਕਰਦੀਆਂ ਹਨ ਅਤੇ ਫੰਕਸ਼ਨ (function) ਜਾਂ ਸਬਰੂਟੀਨਜ਼ (subroutines) ਆਖਿਆ ਜਾਂਦਾ ਹੈ। ਐਰੇਜ਼ (arrays) ਅਤੇ ਸਟਰਕਚਰ (structure) ਦੀ ਵਰਤੋਂ ਵੈਰੀਏਬਲਜ਼ (variables) ਦੇ ਸਮੂਹ ਨੂੰ ਇਕ ਨਾਮ ਦੁਆਰਾ ਦਰਸਾਇਆਂ ਜਾ ਸਕਦਾ ਹੈ। ਅਸਲ ਵਿਚ ਪ੍ਰੋਗਰਾਮ ਲਿਖਣ ਵਾਲੇ ਕਈ ਵਾਰ ਡਾਟਾ ਸਟਰਕਚਰਸ (structures) ਤੇ ਡਾਟਾ ਪ੍ਰਬੰਧ (organization) ਨੂੰ ਪ੍ਰਮੁੱਖ ਮੰਨਦੇ ਹਨ ਅਤੇ ਇਸ ਦੁਆਰਾ ਮੇਨ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਲਿਖਦੇ ਹਨ।

1.6 ਪਬਲੀਸ਼ਰ

ਪਬਲੀਸ਼ਰ ਵਿਚ ਟੈਂਪਲੇਟਸ, ਵਿਜ਼ਾਰਡਸ ਅਤੇ ਡਿਜ਼ਾਇਨ ਗੈਲਰੀ ਦੀ ਵਰਤੋਂ ਕਰਕੇ, ਪਰਚਾਰ ਅਤੇ ਮਸ਼ਹੂਰੀ ਕਰਨ ਵਾਲੇ ਇਸ਼ਤਿਹਾਰਾਂ ਨੂੰ ਬਣਾਇਆ ਜਾਂਦਾ ਹੈ। ਇਹਨਾਂ ਪਰਚਾਰ ਅਤੇ ਮਸ਼ਹੂਰੀ ਕਰਨ ਵਾਲੇ ਇਸ਼ਤੇਹਾਰਾਂ ਨੂੰ ਪਬਲੀਕੇਸ਼ਨ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਪਬਲੀਸ਼ਰ ਇਕ ਐਪਲੀਕੇਸ਼ਨ ਹੈ ਜੋ ਕਿ ਸੁੰਦਰ ਅਤੇ ਵਧੀਆ ਪਰਚਾਰ ਅਤੇ ਮਸ਼ਹੂਰੀ ਕਰਨ ਵਾਲੇ ਇਸਤੇਹਾਰਾਂ ਨੂੰ ਡਿਜ਼ਾਇਨ ਕਰਨ ਅਤੇ ਬਣਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਪੈਮਫਲੇਟ, ਲੈਟਰ-ਹੈਡ, ਬੈਨਰ, ਨਿਊਜ਼ਲੈਟਰ, ਪੋਸਟਰ, ਕੈਲੇਂਡਰ, ਨਿਮੰਤਰਣ ਆਦਿ।

1. **ਬੈਨਰ** : ਤੁਸੀਂ ਕਿਸੀ ਵੀ ਸਮਾਰੋਹ ਲਈ ਪਬਲੀਸ਼ਰ ਵਿੱਚ ਬੈਨਰ ਬਣਾ ਸਕਦੇ ਹੋ ਅਤੇ ਲੋਕਾਂ ਦੀ ਧਿਆਨ ਆਪਣੇ ਵੱਲ ਖਿੱਚ ਸਕਦੇ ਹੋ।
2. **ਨਿਊਜ਼ਲੈਟਰ** : ਤੁਸੀਂ ਲੋਕਾਂ ਨੂੰ ਆਪਣੇ ਸਮਾਰੋਹ ਬਾਰੇ ਜਾਣਕਾਰੀ ਦੇਣ ਲਈ ਨਿਊਜਲੈਟਰ ਬਣਾ ਸਕਦੇ ਹੋ। ਨਿਊਜ਼ਲੈਟਰ ਵਿੱਚ, ਤੁਸੀਂ ਸਮਾਰੋਹ ਵਿੱਚ ਹੋਣ ਵਾਲੇ ਪ੍ਰੋਗਰਾਮਾਂ ਦੇ ਸਮੇਂ ਬਾਰੇ, ਮੁੱਖ ਮੇਹਮਾਨ ਬਾਰੇ ਅਤੇ ਹੋਰ ਵੱਖਰੀਆਂ-ਵੱਖਰੀਆਂ ਸ਼ਖਸ਼ੀਅਤਾ ਬਾਰੇ ਜਾਣਕਾਰੀ ਦੇ ਸਕਦੇ ਹੋ।
3. **ਪੋਸਟ-ਕਾਰਡ** : ਪਬਲੀਸ਼ਰ ਵਿੱਚ, ਤੁਸੀਂ ਕਿਸੀ ਸਮਾਰੋਹ ਲਈ ਪੋਸਟ ਕਾਰਡ ਤਿਆਰ ਕਰ ਸਕਦੇ ਹੋ।
4. **ਲੈਟਰ-ਹੈਡ** : ਲੈਟਰ-ਹੈਡ ਇਕ ਸਟੇਸ਼ਨਰੀ ਹੈ ਜਿਸ ਵਿੱਚ ਸੰਸਥਾ ਜਾਂ ਕਿਸੀ ਵਿਅਕਤੀ ਦਾ ਨਾਂ ਅਤੇ ਪਤਾ ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਹ ਚਿੱਠੀਆਂ ਲਿੱਖਣ ਲਈ ਵਰਤੀਆ ਜਾਂਦਾ ਹੈ।
5. **ਇਨਵੇਲਪ** : ਤੁਸੀਂ ਨਿਮਨਤਰਣ ਪੱਤਰ ਅਤੇ ਚਿੱਠੀਆਂ ਭੇਜਣ ਲਈ ਇਨਵੇਲਪ ਬਣਾ ਸਕਦੇ ਹੋ।

o

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1. ਹੇਠਾਂ ਦਿਤੇ ਪ੍ਰਸ਼ਨਾਂ ਦੇ ਜਵਾਬ ਦਿਓ ?

1. Application ਸਾਫਟਵੇਅਰ ਤੇ System ਸਾਫਟਵੇਅਰ ਵਿਚ ਕੀ ਅੰਤਰ ਹੁੰਦਾ ਹੈ ?
2. MS Acess ਨੂੰ Relational ਡਾਟਾ-ਬੇਸ ਕਿਉਂ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ?
3. if ਅਤੇ countif ਐਕਸਲ ਫੰਕਸ਼ਨਸ ਵਿਚ ਕੀ ਅੰਤਰ ਹੈ ?
4. ਇਕ ਸਹੀ ਪ੍ਰੋਗਰਾਮ ਬਨਾਉਣ ਲਈ ਕਿਹੜੇ-ਕਿਹੜੇ ਪੜਾਵ ਹਨ ?
5. HTML ਵਿਚ ਟੈਗਸ (Tags) ਦਾ ਕੀ ਮਹੱਤਵ ਹੈ ?

ਪ੍ਰਸ਼ਨ 2. ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ (Fill in the blanks)

1. www ਦਾ ਮਹੱਤਵਪੂਰਨ ਦਸਤਾਵੇਜ਼ ________ ਹੁੰਦਾ ਹੈ।
2. ________ ,ਕਾਲਮ 27 ਅਤੇ ਰੋ (row) 30 ਦਾ ਐਡਰੇਸ ਹੋਵੇਗਾ ।
3. ਫਾਰਮ (form) ਨੂੰ ਬਨਾਉਣ ਦਾ ਆਸਾਨ ਤਰੀਕਾ ________ ਹੁੰਦਾ ਹੈ।
4. ਫਲੋ ਚਾਰਟਿੰਗ, ਪ੍ਰੋਗਰਾਮ development cycle ਦੇ ________ ਪੜਾਵ ਵਿਚ ਹੁੰਦਾ ਹੈ।
5. MS office ਸਾਫਟਵੇਅਰ ________ ਦੀ ਸ਼੍ਰੇਣੀ ਨਾਲ ਸਬੰਧਿਤ ਹੈ ।

ਪ੍ਰਸ਼ਨ 3. ਸਹੀ ਜਾਂ ਗਲਤ (true or false)

1. HTML ਦੇ ਟੈਗਸ (Tags) { } ਬਰੈਕਟਸ ਦੇ ਨਾਲ ਸਮਾਪਿਤ ਹੁੰਦੇ ਹਨ ।
2. ਜੇਕਰ ਤੁਸੀਂ excel ਵਿਚ ਕਿਸੇ cell ਵਿਚ 12+24 ਟਾਈਪ ਕਰਦੇ ਹੋ ਤਾਂ ਇਹ ਉਸ cell ਵਿਚ 12+24 ਹੀ ਦਰਸਾਏਗਾ ।
3. ਕੋਈ ਵੀ ਫਾਰਮ (form) ਬਨਾਉਣ ਤੋਂ ਪਹਿਲਾਂ ਟੇਬਲ ਬਨਾਉਣਾ ਜਰੂਰੀ ਹੈ ।
4. ਕਿਸੇ ਸਮਸਿਆ ਦੀ ਪਰਖ ਕਰਨ ਲਈ ਆਊਟਪੁਟ ਜਰੂਰੀ ਨਹੀ ਹੈ ।
5. Excell ਵਿਚ formula ਬਨਾਉਣ ਲਈ 5 ਤਰ੍ਹਾਂ ਦੇ ਗਣਿਤ ਓਪਰੇਟਰ ਇਸਤੇਮਾਲ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ ।

Answer to objective questions

ਪ੍ਰਸ਼ਨ	.1	.2	.3	.4	.5
2	Web Page(s)	AA30	Wizards design	Developing a solution	Application S/w
3	F	T	T	F	T

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiesdtoday.com

ਅਧਿਆਇ 2

"ਸੀ" ਭਾਸ਼ਾ ਵਿਚ ਪ੍ਰੋਗਰਾਮਿੰਗ ਲਈ ਭੂਮਿਕਾ (Introduction to programming in "C" Language)

2.0.0 ਭੂਮਿਕਾ (Introduction)

ਸੀ (C) ਭਾਸ਼ਾ ਦਾ ਡੀਜ਼ਾਈਨ 1970 ਤੋ ਪਹਿਲਾਂ ਬੈੱਲ ਲੇਬਾਰਟਰੀਸ (Bell laboratories) ਵਿਚ **ਡੈਨਿਸ ਰਿਚੀ** (Dennis Ritchie) ਵਲੋਂ ਕੀਤਾ ਗਿਆ। ਸੀ (C) ਜਿਸਨੂੰ ਬੀ (B) ਕਿਹਾ ਜਾਂਦਾ ਸੀ ਦਾ ਪੂਰਵ ਰੂਪਾਂਤਰ **ਕੇਨ ਥਾਮਪਸਨ** (Ken Thompson) ਵਲੋਂ ਲਿਖਿਆ ਗਿਆ ਸੀ ਜਿਸਨੇ ਇਸ ਨੂੰ **ਮਾਰਟਿਨ ਰਿਚਰਡਸ** (Martin Richards) ਦੀ ਬੀ ਸੀ ਪੀ ਐਲ (ਬੇਸਿਕ ਕੰਬਾਈਨਡ ਪ੍ਰੋਗਰਾਮ ਲੈਂਗਵੇਜ) ਤੋਂ ਅਪਣਾਇਆ ਸੀ। ਡੈਨਿਸ ਰਿਚੀ (Dennis Ritchie) ਦੀ ਸੀ **(C)** ਭਾਸ਼ਾ ਅਤੇ **ਬੀ ਸੀ ਪੀ ਐਲ** ਦਾ ਅਧਿਐਨ ਕੀਤਾ ਗਿਆ, ਇਸ ਵਿਚ ਸੁਧਾਰ ਲਿਆਂਦਾ ਗਿਆ ਅਤੇ ਇਸਨੂੰ ਨਾਮ ਦਿੱਤਾ **ਸੀ (C)**, ਜੋ ਕਿ ਬੀ ਸੀ ਪੀ ਐਲ ਦਾ ਦੂਜਾ ਅੱਖਰ ਹੈ।

2.1.0 ਸੀ(C) ਭਾਸ਼ਾ ਦੇ ਵਿਸ਼ੇਸ਼ ਲੱਛਣ (Distinctive Features of 'C' Language)

ਸੀ (C) ਨੂੰ ਇਕ ਭਾਸ਼ਾ ਬੁਲਾਉਣਾ ਗਲਤ ਹੋ ਜਦ ਕਿ ਸਚਮੁੱਚ ਵਿਚ ਇਹ ਇਕ ਚਿੰਨਾਤਮਕ (Symbolic) ਹਿਦਾਇਤੀ ਕੋਡ, ਜੋ ਕਮਾਂਡਸ ਦਾ ਇਕ ਸੈੱਟ ਹੈ ਜੋ ਕੰਪਿਊਟਰ ਤੇ ਕਾਰਵਾਈਆਂ ਕਰਦਾ ਹੈ। ਪ੍ਰਦਸ਼ਨ ਲਈ ਟੈਕਸਟ (text) ਲਿਖਣਾ, ਦੋ ਅੰਕ ਜੋੜਨੇ, ਡਾਟਾ ਨੂੰ ਡਿਸਕ ਤੇ ਬਦਲਣਾ - ਇਹ ਸਾਰੀਆਂ ਕਾਰਵਾਈਆਂ ਅਤੇ ਸੀ ਸਟੇਟਮੈਂਟ (statement) ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਨਗਿਣਤ ਹੋਰ ਪ੍ਰੋਗਰਾਮ ਵੀ ਹੋ ਸਕਦੇ ਹਨ ਜੋ ਸੀ (C) ਨਿਯਮਾਂ ਅਤੇ ਵਿਨਿਯਮਾਂ ਦੇ ਅਨੁਸਾਰ ਬਣਾਏ ਗਏ ਹਨ। ਦੂਜੇ ਸ਼ਬਦਾਂ ਵਿਚ ਇਸ ਨੂੰ ਵਾਕ ਰਚਨਾ (syntax) ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

- **ਸੀ (C) ਭਾਸ਼ਾ ਨੂੰ ਅਕਸਰ 'ਦਰਮਿਆਨੇ ਪੱਧਰ' ਦੀ ਭਾਸ਼ਾ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। (The C language is often described as a "middle-level" language).** ਇਹ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਉਸੇ ਸ਼ੈਲੀ ਵਿਚ ਲਿਖਣ ਦੀ ਆਗਿਆ ਦਿੰਦੀ ਹੈ ਜਿਵੇਂ ਕਿ ਬਹੁਤ ਸਾਰੇ ਆਧੁਨਿਕ ਹਾਈ ਲੇਵਲ ਭਾਸ਼ਾਵਾਂ (high level languages) ਜਿਵੇਂ ਕਿ FORTRAN, COBOL, BASIC, PLII ਅਤੇ Pascal. ਸੀ (C) ਵਿਚ ਇਹ ਸੰਭਵ ਹੈ ਕਿ ਮਸ਼ੀਨ ਨਾਲ ਬਹੁਤ ਹੀ ਲੋਅ-ਲੈਵਲ (low level) ਤੇ ਨਿਪਟਿਆ ਜਾ ਸਕਦਾ ਹੈ।
- **ਇਸਤੋਂ ਇਲਾਵਾ ਸੀ (C) ਇਕ ਹਾਈ ਲੇਵਲ (high level) ਵਿੱਧੀਵਤ (procedural) ਭਾਸ਼ਾ ਹੈ।** ਵਿਧੀਵਤ ਭਾਸ਼ਾ ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ, ਪ੍ਰੋਗਰਾਮਰ ਕੰਪਿਊਟਰ ਵਲੋਂ ਨਿਭਾਏ ਜਾਣ ਵਾਲੇ ਕਦਮਾਂ ਦੇ ਬਾਰੇ ਬਹੁਤ ਸਪੱਸ਼ਟ ਹਿਦਾਇਤਾਂ ਲਿਖਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮਰ ਨੂੰ ਡਾਟਾ ਦੇ ਅਰਥ ਸਮਝ ਆਉਣੇ ਚਾਹੀਦੇ ਹਨ ਨਿਭਾਏ ਜਾਣ ਵਾਲੇ ਕਦਮ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਆਊਟਪੁਟ। ਹਾਈ ਲੇਵਲ ਭਾਸਾ (high level language) ਨਾਲ ਪ੍ਰੋਗਰਾਮਰ ਮਸ਼ੀਨੀ ਹਿਦਾਇਤਾਂ ਲੈਣ ਤੋਂ ਮੁਕਤ ਹੋ ਜਾਂਦਾ ਹੈ।

2.2.0 ਸੀ (C) ਚਿੰਨ ਸੈੱਟ (The 'C' Character Set)

ਪ੍ਰਕ੍ਰਿਤਕ ਭਾਸ਼ਾਵਾਂ ਦੀ ਤਰ੍ਹਾਂ ਕੰਪਿਊਟਰ ਭਾਸ਼ਾਵਾਂ ਦੇ ਚੰਗੇ ਪਰਿਭਾਸ਼ਤ ਕੀਤੇ ਚਿੰਨ੍ਹ ਸੈਟ ਅਤੇ ਵਿਆਕਰਨ ਨਿਯਮ ਹੁੰਦੇ ਹਨ ਜਿਨ੍ਹਾਂ ਨੂੰ ਅਕਸਰ ਵਾਕ-ਰਚਨਾ (syntax) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਚਿੰਨ੍ਹ ਸੈਟ ਦੇ ਦੋ ਸੈਟ ਹੁੰਦੇ ਹਨ। ਉਹ ਹਨ:

Following table1 shows the valid alphabets, numbers and special symbols allowed in C

Alphabets	**A,B,........Y,Z OR a, b ......y ,z**
Digits	**0,1,2,3,4,5,6,7,8,9**
Special Symbols	**~ ' ! @ # % ^ & * ( ) - _ + = I \ { } [ ] : ; " ' < > , . ? /**

Downloaded from https:// www.studiesdtoday.com

Downloaded from https:// www.studiestoday.com

(i) ਐਸਕੇਪ ਚਿੰਨ੍ਹ (Escape characters)

ਕੀ-ਬੋਰਡ ਤੇ ਦਰਸਾਏ ਚਿੰਨ੍ਹ ਪ੍ਰਿੰਟ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ ਜਾਂ ਇਹ ਕੀ (key) ਦਬਾ ਕੇ ਸਕਰੀਨ ਤੇ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਪਰੰਤੂ ਕੁਝ ਲੱਛਣ ਜਿਵੇ ਲਾਈਨ ਫੀਡ, ਫਾਰਮ ਫੀਡ, ਟੈਬ ਆਦਿ ਨਾ ਤਾਂ ਛਾਪੇ ਜਾ ਸਕਦੇ ਹਨ ਅਤੇ ਨਾ ਹੀ ਸਿਧੇ ਪ੍ਰਦਰਸਿਤ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਸੀ (C) ਵੱਲੋ ਸਾਧਨ (mechanism) ਪ੍ਰਦਾਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈਂ ਜੋ ਅਜਿਹੇ ਚਿੰਨ੍ਹਾਂ (characters) ਨੂੰ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਜਿਹੜੇ ਦਿਖਾਈ ਨਹੀ ਦਿੰਦੇ ਜਾਂ ਐਸਕੇਪ ਚਿੰਨ੍ਹਾਂ (escape characters) ਰਾਹੀਂ ਪ੍ਰਾਪਤ ਕਰਨੇ ਮੁਸ਼ਕਿਲ ਹੁੰਦੇ ਹਨ।

Escape Character	Meaning	Escape Character	Meaning
"\n"	New line	"\r"	Carriage return
"\t"	Tab	"\\"	Back slash
"\f"	form feed	"\ ' "	Single quote
"\b"	Back Space	"\ " "	Double quote

(ii) ਵਾਈਟ ਸਪੇਸ ਕਰੈਕਟਰ (White Space Characters)

ਕੋਈ ਵੀ ਚਿੰਨ੍ਹ ਜਿਹੜੇ ਪ੍ਰਿੰਟ ਕਰਨ ਤੇ ਖਾਲੀ ਸਥਾਨ ਪੇਸ਼ ਕਰਦੇ ਹਨ, ਨੂੰ ਵਾਈਟ ਸਪੇਸ ਕਰੈਕਟਰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜਿਵੇ ਕਿ ਸਪੇਸ, ਟੈਬ, ਨਵੀਆਂ ਲਾਈਨਾਂ ਅਤੇ ਟਿੱਪਣੀਆਂ (comments) ਆਦਿ।

2.3.0 ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ ਸਟਰਕਚਰ (Structure of a C Program)

ਵੱਖ ਵੱਖ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਦੀ ਕੋਡਿੰਗ ਦੀ ਆਪਣੀ ਹੀ ਫਾਰਮੇਟ (format) ਹੁੰਦੀ ਹੈ। ਸੀ ਪ੍ਰੋਗਰਾਮ ਦੇ ਬੁਨਿਆਦੀ ਭਾਗ ਹਨ:

The Basic structure of a C program

• void main ( )
• pair of curly braces { , }
• declarations and statements
• User-defined functions.
	Preprocessor statements
	Global declarations
	void main ( )
	{
	declaration;
	statements;
	}
	User defined functions

2.3.1 ਹੈਡਰ ਫਾਈਲਜ਼ (Header files)

ਫਾਈਲਸ ਜੋ ਹੈਡਰ ਤੇ ਰੱਖੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੇ main() ਤੋਂ ਪਹਿਲਾਂ ਨੂੰ ਹੈਡਰ ਫਾਈਲਾਂ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਆਮ ਤੌਰ ਤੇ .h ਐਕਸਟੈਂਸ਼ਨ (extension) ਨਾਲ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ।

ਇਕ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕੋਈ ਰੇਖਾ ਜਿਹੜੀ # ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦੀ ਹੋਵੇ ਕੰਪਾਈਲਰ ਲਈ ਇਕ ਹਿਦਾਇਤ ਹੁੰਦੀ ਹੈ ਨਾ ਕਿ ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਇਕ ਅਸਲ ਸਟੇਟਮੈਂਟ (statement)। ਹੈਡਰ ਲਾਈਨਾਂ ਸਰੋਤ (source) ਪ੍ਰੋਗਰਾਮਾਂ ਵਿਚ # ਦੇ ਰਾਹੀਂ ਪ੍ਰਵੇਸ਼ ਕਰਦੀਆਂ ਹਨ ਜਿਸ ਵਿਚ ਸਟੇਟਮੈਂਟ (statement) ਸ਼ਾਮਲ ਹੁੰਦੀ ਹੈ। ਇਸ ਲਈ # ਸਟੇਟਮੈਂਟ (statement) ਇਕ ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਸਟੇਟਮੈਂਟ (statement) ਹੁੰਦੀ ਹੈ ਅਤੇ ਇਹ ਜਰੂਰ ਪ੍ਰੋਗਰਾਮ ਦੇ ਸ਼ੁਰੂ ਵਿਚ ਲਿਖੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ।ਹਰ ਇਕ ਹੈਡਰ ਫਾਈਲ ਵਿਚ ਕੁਝ ਸੰਖੇਪਤ ਲਾਈਬ੍ਰੇਰੀ ਫੰਕਸ਼ਨ (Library function) ਲਈ ਡਿਕਲੇਰੇਸ਼ਨ ਹੁੰਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

2.3.2 ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਸਟੇਟਮੈਂਟ/ਨਿਰਦੇਸ਼ (Preprocessor statements/directives)

ਕੰਪਾਈਲਿੰਗ (compiling) ਦੇ ਸਮੇਂ ਕੁਝ ਪ੍ਰਕਿਰਿਆਵਾਂ ਸੀ (C) ਵਿਚ ਕੀਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਉਹ ਕਮਾਂਡ ਜਿਹੜੀਆਂ ਅਜਿਹੀਆਂ ਪ੍ਰਕਿਰਿਆਵਾਂ ਕਰਦੀਆਂ ਹਨ ਨੂੰ ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਨਿਰਦੇਸ਼ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਵਿਕਲਪ (optional) ਵਜੋਂ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਮੌਜੂਦ ਹੁੰਦੀਆਂ ਹਨ। ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਕੰਪਾਈਲ (Compile) ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਦੇ ਨਿਰਦੇਸ਼ ਲਾਗੂ ਕੀਤੇ ਜਾਂਦੇ ਹਨ. ਇਹ ਨਿਮਨਲਿਖਤ ਕਾਰਵਾਈਆਂ ਕਰਨ ਲਈ ਲਾਭਦਾਇਕ ਹੁੰਦੇ ਹਨ:

- ਮੁੱਲਾਂ ਦਾ ਪ੍ਰਤਿਸਥਾਪਣ — Substitution of values
- ਫਾਈਲਾਂ ਦੀ ਸਮੂਲੀਅਤ — Including files
- ਕੰਡੀਸ਼ਨਲ ਕੰਪਾਈਲੇਸ਼ਨ — Conditional compilation

ਸੀ (C) ਨਾਲ ਤਿੰਨ ਪ੍ਰਕਾਰ ਦੇ ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਨਿਰਦੇਸ਼ (directives) ਉਪਲਬਧ ਹੁੰਦੇ ਹਨ। ਉਹ ਹਨ # include ਮੈਕਰੋ ਪ੍ਰਤਿਸਥਾਪਣ (# define) ਅਤੇ ਕੰਡੀਸ਼ਨਲ (# if) ਨਿਰਦੇਸ਼ (directives). ਇਹ ਸਟੇਟਮੈਂਟ (statement) ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਨੂੰ ਹੈਡਰ ਫਾਈਲਾਂ ਅਤੇ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਚਿੰਨ੍ਹਾਤਮਕ ਨਿਰੰਤਰਤਾ ਸ਼ਾਮਲ ਕਰਨ ਲਈ ਨਿਰਦੇਸ਼ ਦਿੰਦੇ ਹਨ। ਜਿਵੇਂ

# include<stdio.h>	: for the standard I/O functions
#include<math.h>	: for certain mathematical functions
#include <string.h>	: for string manipulation functions.
# include "test.h"	: for file inclusion of header file test
# define NULL 0	: for defining symbolic constant, NULL = 0.

2.3.3 ਗਲੋਬਲ ਡਿਕਲੇਰੇਸ਼ਨਸ (Global declarations):

ਵੇਰੀਏਬਲਸ (variables) ਜਾਂ ਫੰਕਸ਼ਨਸ (functions) ਜਿਨ੍ਹਾਂ ਦੀ ਹੱਦ ਮੇਨ ਫੰਕਸ਼ਨ ਅਤੇ ਹੋਰ ਯੂਜ਼ਰ ਡਿਫਾਇੰਡ ਪ੍ਰੋਗਰਾਮਸ (user defined programs) ਵੱਜੋਂ ਜਾਣੀ ਜਾਂਦੀ ਹੈ ਨੂੰ ਗਲੋਬਲ ਵੇਰੀਏਬਲਸ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਉਨ੍ਹਾਂ ਦੀ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਨੂੰ **ਗਲੋਬਲ** ਡਿਕਲੇਰੇਸ਼ਨਸ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

main( ): ਜਿਵੇਂ ਕਿ ਨਾਮ ਤੋਂ ਹੀ ਪ੍ਰਤੀਤ ਹੁੰਦਾ ਹੈ ਇਹ ਹਰੇਕ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ main ਫੰਕਸ਼ਨ ਹੁੰਦਾ ਹੈ। ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੀ ਸ਼ੁਰੂਆਤ main ( ) ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦੀ ਹੈ। ਇਹ ਲੋਅਰ-ਕੇਸ ਅੱਖਰਾਂ ਦੇ ਨਾਲ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਸੈਮੀਕੋਲਨ ਦੁਆਰਾ ਖਤਮ ਨਹੀਂ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ। ਇਸਨੂੰ ਹੋਰ ਲਾਇਬ੍ਰੇਰੀ ਫੰਕਸ਼ਨ ਅਤੇ ਯੂਜ਼ਰ ਡਿਫਾਇੰਡ ਫੰਕਸ਼ਨਸ (user defined functions) ਨੂੰ ਬੁਲਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਹਰ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕੇਵਲ ਇਕ ਹੀ main ( ) ਫੰਕਸ਼ਨ ਹੁੰਦਾ ਹੈ।

Global variables are defined above main( ) in the following way:-

```
short number, sum=0;      } ਗਲੋਬਲ
int bignumber, bigsum;    } ਵੇਰੀਏਬਲ
char letter;              }

void main ( )
  {
        ------------------------
  }
```

It is also possible to pre-initializes global variables using the = operator for assignment

ਬਰੇਸਿਸ (Braces): ਹਰ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਲੋਂ ਘੁੰਗਰਾਲੇ ਚਿੰਨ੍ਹਾਂ ({,}) ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਖੱਬੀ ਬਰੇਸਿਸ ਆਰੰਭ ਦਾ ਪ੍ਰਤੀਕ ਹੁੰਦੀ ਹੈ ਅਤੇ ਸੱਜੀ ਬਰੇਸਿਸ ਯੂਜ਼ਰ ਪਰਿਭਾਸ਼ਤ ਫੰਕਸ਼ਨ (function) ਜਾਂ main ਦਾ ਅੰਤ ਹੁੰਦੀ ਹੈ।

ਡਿਕਲੇਰੇਸ਼ਨ (Declarartions): ਇਹ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ ਭਾਗ ਹੁੰਦਾ ਹੈ ਜਿਥੇ ਸਮੂਹ ਵੇਰੀਏਬਲ (variable), ਐਰੇ (array), ਫੰਕਸ਼ਨ (function) ਆਦਿ ਬੁਨਿਆਦੀ ਡਾਟਾ ਟਾਈਪਸ ਦੇ ਨਾਲ ਸ਼ੁਰੂ ਕੀਤੇ ਜਾਂਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਟੇਟਮੈਂਟ (Statement): ਇਹ ਕੰਪਿਊਟਰ ਲਈ ਹਿਦਾਇਤਾਂ ਹਨ ਜੋ ਵਿਸ਼ੇਸ਼ ਕੰਮਾਂ ਨੂੰ ਕਰਨ ਲਈ ਕਿਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਇਹ ਆਈ/ਓ ਸਟੇਟਮੈਂਟ (I/O statement), ਗਣਿਤ ਦੇ ਸਟੇਟਮੈਂਟ (statement) ਅਤੇ ਹੋਰ ਸਟੇਟਮੈਂਟ (statement) ਹੋ ਸਕਦੇ ਹਨ। ਟਿੱਪਣੀ ਕੀਤੇ ਜਾਣ ਵਾਲੇ ਸਟੇਟਮੈਂਟਸ (statements) ਨੂੰ /* ਅਤੇ */ ਦੇ ਅੰਦਰ ਲਿਖਿਆ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ ਜਿਹਨਾਂ ਨੂੰ ਕੰਪਾਈਲ (compile) ਅਤੇ ਲਾਗੂ ਨਹੀ ਕੀਤਾ ਜਾਂਦਾ ।

A sample example of C program is given below:

```
# include <stdio.h>
void main( )
{
printf ( "Welcome to C \n") ;
}
```

- ਪਹਿਲੀ ਲਾਈਨ ਕੰਪਾਈਲਰ (compiler) ਨੂੰ ਦਸਦੀ ਹੈ ਕਿ ਡਾਟਾ ਪੜ੍ਹਣ ਅਤੇ ਪ੍ਰਿੰਟਿੰਗ ਲਈ ਸਟੈਂਡਰਡ ਆਈ/ਓ (I/O) ਹੈਡਰ ਫਾਈਲ ਨੂੰ ਸ਼ਾਮਲ ਕੀਤਾ ਜਾਵੇ ।
- ਦੂਜੀ ਲਾਈਨ ( ) main ਹੈ, ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ main ਫੰਕਸ਼ਨ (function)। ਮਿਸਾਲ ਲਈ ਪ੍ਰੋਗਰਾਮ ਅੰਦਰ ਦਿੱਤੀ ਸਟੇਟਮੈਂਟ **printf ("Welcome to C\n");**

ਜਦੋਂ ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਲਿਆ ਜਾਂਦਾ ਹੈ main( ) ਪ੍ਰਿੰਟ( ) ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਬੁਲਾਉਂਦੀ ਹੈ ਤਾਂ ਜੋ ਇਸ ਵਿਚ <stdio.h> ਸ਼ਾਮਲ ਕੀਤਾ ਜਾਵੇ। ਪ੍ਰਿੰਟ( ) ਸਕਰੀਨ ਤੇ "Welcome to C" ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ।

2.3.4 ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦਾ ਕੰਪਾਈਲ ਅਤੇ ਲਾਗੂ ਕਰਨ (Compiling and execution of C Program):

ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੇ ਕੰਪਾਈਲ (compile) ਦਾ ਭਾਵ ਹੈ ਇਸਨੂੰ ਮਸ਼ੀਨ ਭਾਸ਼ਾ ਵਿਚ ਬਦਲਣਾ। ਇਸ ਕੰਮ ਲਈ ਸੀ (C) ਕੰਪਾਈਲਰ (compiler) ਸ਼ਾਮਲ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਕੰਪਾਈਲ (compile) ਕਰਨ ਲਈ ਇਕ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਨਾਲ ਟਾਈਪ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।

ਸੀ (C) ਕੰਪਾਈਲਰ ਐਡੀਟਰ ਦੇ ਨਾਲ ਜਾਂ ਉਨ੍ਹਾਂ ਤੋਂ ਬਿਨਾ ਉਪਲਬਧ ਹੁੰਦੇ ਹਨ। ਉਹ ਵਾਤਾਵਰਣ ਜਿਥੇ ਤੁਹਾਨੂੰ ਕੰਪਾਈਲਰ, ਐਡੀਟਰ, ਡਿਬਗਿੰਗ ਉਪਕਰਣ, ਲਿੰਕਿੰਗ ਸਹੂਲਤਾਂ, ਟਰੇਸਿੰਗ ਅਤੇ ਟੈਸਟਿੰਗ ਉਪ-ਕਰਣ ਉਪਲੱਬਧ ਹੋਣ ਨੂੰ ਸੰਗਠਿਤ ਵਿਕਾਸ ਵਾਤਾਵਰਣ (Integrated Development Environment / IDE) ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ : Turbo C (TC), Borland C ਆਦਿ ।

ਇਕ ਬੁਨਿਆਦੀ ਪ੍ਰੋਗਰਾਮ ਦੇ ਸਫਲਤਾ ਪੂਰਵਕ ਲਾਗੂ ਹੋਣ ਲਈ ਲੋੜੀਦੇ ਪੰਜ ਕਦਮ :

1.	ਪ੍ਰੋਗਰਾਮ ਫਾਈਲ ਬਣਾਉਣੀ	Creating a program file
2.	ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸੇਵ ਕਰਨਾ	saving the program
3.	ਕੰਪਾਈਲੇਸ਼ਨ	Compilation.
4.	ਲਿੰਕਿੰਗ ਸਿਸਟਮ ਲਾਇਬ੍ਰੇਰੀ ਫੰਕਸ਼ਨ	Linking system library function
5.	ਰਨਿੰਗ (ਲਾਗੂ-ਕਰਨ) ਪ੍ਰੋਗਰਾਮ	Running (executing) Program

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੇ ਲਾਗੂ ਕਰਨ ਅਤੇ ਕੰਪਾਈਲੇਸ਼ਨ ਵਿਚ ਵਰਤੀ ਜਾਂਦੀ ਵਿਧੀ ਇਕ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ ਤੋਂ ਦੂਜੀ ਲਈ ਭਿੰਨ ਹੁੰਦੀ ਹੈ। DOS ਪਲੇਟਫਾਰਮ (ਵਿੰਡੋ OS ਅਧੀਨ) ਤੇ ਜੇਕਰ ਤੁਸੀਂ C/C ++ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰ ਰਹੇ ਹੋ ਤਾਂ ਤੁਹਾਨੂੰ ਹੇਠ ਦਿਤੇ ਗਏ ਕਦਮ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੇ ਪ੍ਰਵੇਸ਼ ਕਰਨ ਅਤੇ ਕੰਪਾਈਲੇਸ਼ਨ (Compilation) ਵਿਚ ਮਦਦ ਦੇਣਗੇ ।

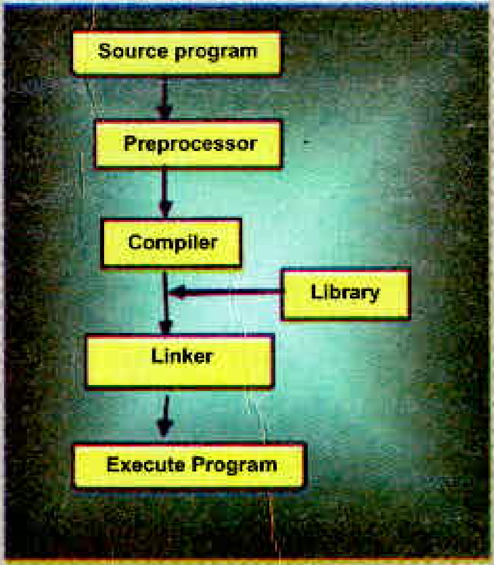
2.4.0 ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ (Use of Editor):

- ਫਿਰ TC tutor ਐਡੀਟਰ ਲਾਗੂ ਅਤੇ ਲੋਡ ਲਈ tc ਟਾਈਪ ਕਰੋ।
- F3 ਕੀ ਦਬਾਉ ਅਤੇ ਫਾਈਲ ਨਾਮ (ਪ੍ਰੋਗਰਾਮ ਦਾ ਨਾਮ ਜਿਵੇਂ ਕਿ **abc.c**) ਦਿਉ ਅਤੇ ਫਿਰ ਐਂਟਰ ਕੀ ਦਬਾਓ।
- ਹੁਣ ਐਡੀਟਰ ਖੁੱਲ੍ਹ ਗਿਆ ਹੈ ਅਤੇ ਟਾਈਪ ਕਰੋ ਸੀ (C) ਦਾ ਪ੍ਰੋਗਰਾਮ। ਇਸਨੂੰ ਸੇਵ ਕਰਨ ਲਈ F2 ਕੀ ਦਬਾਓ ।
- ਪ੍ਰੋਗਰਾਮ ਦੇ ਕੰਪਾਈਲੇਸ਼ਨ (Compilation) ਲਈ ALT+F9 ਕੀਜ਼ (keys) ਨੂੰ ਇਕਠਿਆਂ ਦਬਾਓ ਜਾਂ ਮੀਨੂੰ ਬਾਰ ਚੋਂ ਸਿਲੈਕਟ ਕੰਪਾਈਲ ਆਪਸ਼ਨ ਅਤੇ ਡਰਾਪ ਡਾਊਨ ਮੀਨੂੰ ਚੋਂ build all options ਚੁਣੋ। ਫਿਰ ਪ੍ਰੋਗਰਾਮ ਦੀ ਇਕ obj (object) ਫਾਈਲ ਬਣਾਉਣ ਲਈ ਐਂਟਰ (enter) ਕੀ ਨੂੰ ਦਬਾਓ ।
- ਪ੍ਰੋਗਰਾਮ ਚਲਾਉਣ ਲਈ ਨਾਲੋ ਨਾਲ ALT+R ਕੀਜ ਨੂੰ ਦਬਾਉ ਜਾਂ ਡਰਾਪ ਡਾਊਨ ਮੀਨੂੰ ਤੋਂ ਚਲਾਓ Run ਵਿਕਲਪ ਨੂੰ ਚੁਣੋ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਲਾਗੂ ਕਰਨ ਲਈ ਐਂਟਰ (enter) ਕੀ ਦਬਾਓ।
- ਸਕਰੀਨ ਤੇ ਆਊਟਪੁਟ (output) ਦਿਖਾਉਣ ਲਈ ALT+F5 ਦਬਾਓ।
- ਐਡਿਟ ਸਕਰੀਨ ਤੇ ਵਾਪਿਸ ਜਾਉਣ ਲਈ ਕਿਸੇ ਵੀ ਕੀ ਨੂੰ ਦਬਾਓ।

Stages of Compilation

&

Linking



2.5.0 ਫੰਕਸ਼ਨ (Function):

C ਭਾਸ਼ਾ ਦੇ ਵਿਚ ਫੰਕਸ਼ਨ ਉਪ-ਪ੍ਰੋਗਰਾਮ ਹੁੰਦੇ ਹਨ ਜਿਹੜੇ ਕਿਸੇ ਨਿਸ਼ਚਿਤ ਕਾਰਜਾਂ ਨੂੰ ਕਰਨ ਲਈ ਲਿਖੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਇਹ ਸੰਪੂਰਨ ਅਤੇ ਸੁਤੰਤਰ ਹੁੰਦੇ ਹਨ, ਕਿਓਂਕਿ ਇਹਨਾ ਦੀ ਆਪਣੀ ਹੀ ਵੇਰੀਏਬਲਸ ਦੀ ਡਿਕਲੇਰੇਸ਼ਨ, ਸਟੇਟਮੈਂਟਸ ਹੁੰਦੀਆਂ ਹਨ ਜਿਹਨਾਂ ਦੀ ਸ਼ੁਰੂਆਤ ਅਤੇ ਸਮਾਪਤੀ ਫੰਕਸ਼ਨ ਦੇ ਵਿਚ ਹੀ ਹੁੰਦੀ ਹੈ। ਸੀ ਦੇ ਵਿਚ ਫੰਕਸ਼ਨ ਦੋ ਤਰ੍ਹਾਂ ਦੇ ਹੁੰਦੇ ਹਨ :

ਬਿਲਟ ਇਨ ਫੰਕਸ਼ਨਜ਼ / ਲਾਈਬਰੇਰੀ ਫੰਕਸ਼ਨਜ਼	**(Built In Functions/library functions)**
ਯੂਜ਼ਰ ਪਰਿਭਾਸ਼ਤ ਫੰਕਸ਼ਨਜ਼	**(User Defined Functions)**

2.5.1 ਬਿਲਟ ਇਨ ਫੰਕਸ਼ਨਸ (Built In Functions)

ਇਹ ਫੰਕਸ਼ਨ ਸੀ ਭਾਸ਼ਾ ਵਿਚ ਹਮੇਸ਼ਾ ਉਪਲਬਦ ਹੁੰਦੇ ਹਨ ਜਿਹੜੇ ਕਿ ਕਿਸੇ ਵੀ ਵੇਲੇ ਪ੍ਰੋਗਰਾਮ ਵਿਚ call ਕਿਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਉਪਲਬਦ **ਫੰਕਸ਼ਨਜ਼ ਦੀ ਵਰਤੋਂ** ਲਈ, ਜਿਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋ ਕਰਨੀ ਹੈ ਦਾ ਨਾਮ ਲਿਖਣ ਤੋਂ ਬਾਅਦ ਬਰੈਕਟਾਂ ਦੇ ਅੰਦਰ ਆਰਗੂਮੈਂਟ ਨੂੰ ਲਿਖ ਕੇ call ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

2.5.2 ਯੂਜ਼ਰ ਪਰਿਭਾਸ਼ਤ ਫੰਕਸ਼ਨਸ (User Defined Functions)

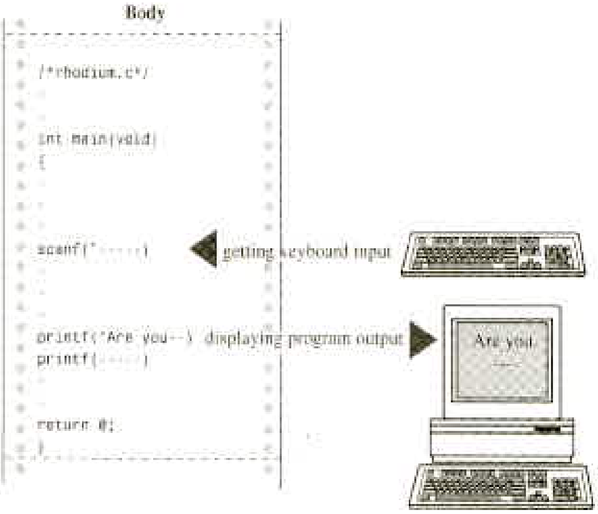
ਇਹ ਉਪ-ਪ੍ਰੋਗਰਾਮ ਹੁੰਦੇ ਹਨ। ਆਮ ਤੌਰ ਤੇ ਉਪ-ਪ੍ਰੋਗਰਾਮ ਇਕ ਫੰਕਸ਼ਨ (function) ਹੁੰਦਾ ਹੈ ਅਤੇ ਉਨ੍ਹਾਂ ਵਿਚ ਵਿਸ਼ੇਸ਼ ਫੰਕਸ਼ਨ (function) ਕਰਨ ਲਈ ਸਟੇਟਮੈਂਟਸ (statements) ਦਾ ਇਕ ਸੈੱਟ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ। ਇਹ ਯੂਜ਼ਰਸ ਵਲੋਂ ਲਿਖੇ ਜਾਂਦੇ ਹਨ ਇਸ ਲਈ ਇਹਨਾਂ ਨੂੰ ਯੂਜ਼ਰ ਡਿਫਾਇੰਡ ਫੰਕੰਸ਼ਨ (user defind function) ਕਹਿੰਦੇ ਹਨ। ਇਹ main( ) ਫੰਕਸ਼ਨ (function) ਦੇ ਪਹਿਲਾਂ ਜਾਂ ਬਾਅਦ ਵਿਚ ਲਿਖੇ ਜਾ ਸਕਦੇ ਹਨ।

2.6.0 ਫਾਰਮੈਟਡ ਆਈ /ਓ ਫੰਕਸ਼ਨ : scanf( ) ਅਤੇ printf( ) ਫੰਕਸ਼ਨ (Formatted I/O Functions : [scanf( ) & printf( ) functions])

ਫਾਰਮੈਟਿਡ ਆਈ/ਓ (I/O) ਸ਼੍ਰੇਣੀ ਦੀ ਇਨਪੁਟ (input) ਲਈ scanf( ) ਫੰਕਸ਼ਨ (function) ਅਤੇ ਆਉਟਪੁਟ (output) ਲਈ printf( ) ਫੰਕਸ਼ਨ (function) ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

2.6.1 printf Function

ਇਹ ਸੀ (C) ਦਾ ਲਾਇਬ੍ਰੇਰੀ ਫੰਕਸ਼ਨ (Function) ਹੁੰਦਾ ਹੈ ਜੋ ਸਾਨੂੰ ਨਿਸ਼ਚਿਤ ਫਾਰਮੇਟ (format) ਵਿਚ ਆਉਟਪੁਟ ਡਾਟਾ ਦਿੰਦਾ ਹੈ। printf() ਫੰਕਸ਼ਨ ਜੋ ਨਿਸ਼ਚਿਤ ਰੂਪ-ਰੇਖਾ (format) ਅਤੇ ਹੋਰ ਦਲੀਲਾਂ (arguments) ਦੇ ਅਨੁਸਾਰ ਆਉਟਪੁਟ ਡਾਟਾ ਮੋਨੀਟਰ ਦੀ ਸਕਰੀਨ ਤੇ ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ ਜਿਵੇ ਕਿ ਚਿੱਤਰ ਵਿਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ ----- (Are you) -------



ਸਟਰਿੰਗ ਫਾਰਮੇਟ (String format) ਦੇ ਟਾਈਪਸ (types) ਦੇ ਹੁੰਦੇ ਹਨ (1) ਚਿੰਨ ਜਿਹੜੇ ਸਕਰੀਨ ਤੇ ਪ੍ਰਿੰਟ ਹੋਣਗੇ ਅਤੇ (2) ਫਾਰਮੇਟ ਕਮਾਂਡ (format command) ਜਿਹੜੇ ਇਹ ਦਰਸਾਉਂਦੇ ਹਨ ਕਿ printf() ਲਈ ਹੋਰ ਕੰਮ ਕਿਵੇ ਪ੍ਰਦਰਸ਼ਿਤ (display) ਕੀਤੇ ਜਾਂਦੇ ਹਨ।ਬੁਨਿਆਦੀ ਤੋਰ ਤੇ ਤੁਸੀ ਇਕ ਸਟਰਿੰਗ ਫਾਰਮੇਟ (String format) ਨਿਸ਼ਚਿਤ ਕਰਦੇ ਹੋ । ਜਿਸ ਵਿਚ ਇਕ text ਹੁੰਦਾ ਹੈ ਅਤੇ ਨਾਲ ਹੀ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ ਹੁੰਦੇ ਹਨ ਜੋ printf() ਦੇ ਹੋਰਨਾਂ ਕੰਮਾ ਨੂੰ ਵੀ ਨਕਸ਼ਾਂ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ। ਇਸ ਦੀ ਵਾਕ ਰਚਨਾਂ (syntax) ਇਸ ਪ੍ਰਕਾਰ ਹੈ

printf("format string", arg1, arg2, arg3…., argn);

ਜਿਥੇ ਫਾਰਮੇਟ ਸਟਰਿੰਗ ਫੰਕਸ਼ਨ (Function) ਨੂੰ ਦਸਦਾ ਹੈ ਕਿਥੇ ਅਤੇ ਕਿਸ ਰੂਪ ਵਿਚ ਰਿਜਲਟ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤੇ ਜਾਣੇ ਹਨ ਅਤੇ ਪ੍ਰਗਟਾਓ (expression) ਜਿਹਨਾਂ ਦਾ ਅਸੀ ਮੁੱਲਾਂਕਣ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।

<format string > could be,	**%f** for displaying **floating point values** **%d** for displaying **integer values** **%c** for displaying **character values**

arg1, arg2, arg3…., argn may be constants , variables ,expressions etc.

ਮਿਸਾਲ 1: ਤੁਸੀ ਸਕਰੀਨ ਤੇ ਪ੍ਰਿੰਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹੋ ਕਿ ਮੁੱਲ 7 ਦਾ 12 **times 84** ਹੈ **printf( )** ਦੀ ਵਰਤੋਂ ਕਰਕੇ

printf("The value 7 times 12 is %d" , 7 * 12);

ਇਹ ਸਟੇਟਮੈਂਟ (statement) printf() ਨੂੰ ਦੋ ਦਲੀਲਾਂ ਦਿੰਦੀ ਹੈ। ਪਹਿਲਾ ਸੰਦੇਸ਼ ਹੈ "ਮੁੱਲ (value) 7 times 12 ਅਤੇ ਦੂਜਾ ਪ੍ਰਗਟਾਓ (expression) ਦਾ ਮੁੱਲ 7*12 ਹੈ। ਪਹਿਲੀ ਦਲੀਲ ਜੋ printf( ) ਨੂੰ ਦਿਤੀ ਗਈ ਹੈ ਉਹ ਸੰਦੇਸ਼ (message) ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। ਇਕ ਸੰਦੇਸ਼ ਜਿਸ ਵਿਚ ਬਦਲੀ ਨਿਯੰਤਰਣ ਕ੍ਰਮ ਵੀ ਸ਼ਾਮਲ ਹੋਵੇ ਜਿਵੇਂ ਕਿ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

%d ਨੂੰ ਸਟਰਿੰਗ-ਫਾਰਮੇਟ (string format) ਵਜੋਂ ਕਿਹਾ ਗਿਆ ਹੈ। ਜਦੋਂ printf() ਬਦਲੀ ਨਿਯੰਤਰਣ ਕ੍ਰਮ ਨੂੰ ਇਸਦੇ ਸਟਰਿੰਗ (string) ਵਜੋਂ ਦੇਖਦਾ ਹੈ, ਇਹ ਬਦਲੀ ਨਿਯੰਤਰਣ ਕ੍ਰਮ ਦੀ ਜਗ੍ਹਾ ਤੇ ਅਗਲੀ ਦਲੀਲ ਦਾ ਮੁੱਲ (value) ਦਸਦਾ ਹੈ। ਕਿਉਂ ਜੋ ਅਗਲੀ ਦਲੀਲ ਹੈ ਪ੍ਰਗਟਾਓ (expression) 7*12 ਜੋ ਕਿ 84 ਹੈ, ਇਹ ਉਹ ਮੁੱਲ (value) ਹੈ ਜੋ ਪ੍ਰਦਰਸ਼ਿਤ (display) ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਲਈ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਗਿਆ ਆਊਟਪੁਟ ਹੈ:

The value 7 times 12 is 84

ਇਸ ਫੰਕਸ਼ਨ (function) ਦੀ ਵਰਤੋਂ ਲਈ ਆਓ ਨਿਮਨਲਿਖਤ ਮਿਸਾਲ 2 ਤੇ ਵਿਚਾਰ ਕਰੀਏ si=p*r*t/100; ਨਾਲ ਸਧਾਰਣ ਵਿਆਜ ਮੂਲਧਨ ਦੀ ਦਿਤੀ ਗਈ ਕੀਮਤ, ਦਰ ਅਤੇ ਸਮਾਂ ਜਿਵੇਂ ਕਿ p ਅਤੇ t ਪੂਰਣ ਅੰਕ ਹਨ ਜਦ ਕਿ r, si ਦਸਮਲਵ ਵਿਚ ਹੋ ਸਕਦੇ ਹਨ, ਅਸੀ ਉਨ੍ਹਾਂ ਨੂੰ ਵੇਰੀਏਬਲ ਵਜੋਂ ਲਿਖਦੇ ਹਾਂ ਅਤੇ ਉਨ੍ਹਾਂ ਨੂੰ ਵਰਤਣ ਤੋਂ ਪਹਿਲਾਂ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਭਾਗ ਵਿਚ ਡਿਕਲੇਅਰ (declare) ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਨਾਲੇ ਜੇਕਰ ਵਿਆਜ ਦੀ ਦਰ ਦਾ ਪਤਾ ਹੈ ਤਾਂ ਮੰਨ ਲਓ r = 5.5 ਤਦ ਜੇਕਰ ਅਸੀ ਮੰਨਦੇ ਹਾਂ ਕਿ p = 1000 & t = 3 yrs.

ਮਿਸਾਲ 2.1

```
int p, t ;
declarations are     p=1000, t =3 ;
                     float r, si ;
                     r = 5.5 ;

Statement 1   is   si =p*r*t*/100 ;

Now to display Simple interest,
we write

Statement 2        printf("%f ",si );
```

ਮਿਸਾਲ 2.2

The structure of C program is

```
void main( )
   {
     declaration;
     statement 1 ;
     statement 2 ;......
     statement n ;
   }
```

Every C statement ends with ;

ਮਿਸਾਲ 2.3

```
# include <stdio.h>
    /* To print values on screen */

void main()
    {
     int p ,t ;
      p=1000, t =3 ;
     float r , si ;
      r =5.5 ;
     si = (p*r*t) / 100 ;
    printf("%f" ,si);
    }
```

Output will be : 165.000000

ਆਓ printf ( ) ਦੇ ਵੱਖ-ਵੱਖ ਰੂਪਾਂ ਨੂੰ ਵਿਚਾਰੀਏ

ਹਰ ਇਕ ਵੇਰੀਏਬਲ/ਕਾਂਸਟੈਂਟ ਦਾ ਮੁੱਲ (value) ਪ੍ਰਦਰਸ਼ਿਤ (display) ਕਰਨ ਲਈ

Statement	Output
printf("%d %f % d % f" , p,r,t,si) ;	1000 5.5 3 165
printf("Simple interest = Rs. %f", si);	Simple interest = Rs 165
printf("Principal = %d \nRate = %f " ,p,r); Where \n inserts blank line between two outputs	Principal = 1000 Rate = 5.5

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ 3: ਖੇਤਰੀ ਚੌੜਾਈ ਵਸ਼ਿਸ਼ਠਤਾ ਦਿੰਦੇ ਹੋਏ ਇਕ ਮੁੱਲ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨਾ
To print a value giving field width specification

```
# include <stdio.h>
void main()
 {
  float a =123.465 ;
  printf( "% 7f  %7.3f %7.1f" ,a,a,a);
```

When the program is executed the **output** will be

123.465000   123.456   123.5

Note that the rounding occurs in the 3rd number because of the precision specification (one decimal place)

ਮਿਸਾਲ 4:

```
# include <stdio.h>
 void main()
 {
int a =15;
float b=34.76;
char c[2] = "Rs";
printf("%d %f %s", a,b,c);
```

Output will be 15 34.76 Rs

2.6.2 scanf( ): ਇਹ ਨਿਸ਼ਚਿਤ ਫਾਰਮੇਟ (format) ਵਿਚ ਯੂਜ਼ਰ ਨੂੰ Numeric, character ਅਤੇ ਸਟਰਿੰਗ ਟਾਈਪ ਡਾਟਾ ਇਨਪੁਟ (input) ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦਾ ਹੈ। ਇਹ ਫੰਕਸ਼ਨ stdio.h ਵਿਚ ਹੁੰਦਾ ਹੈ ਜਿਸ ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਦੇ ਸ਼ੁਰੂ ਵਿਚ ਦਿਤਾ ਜਾਂਦਾ ਹੈ।

Its syntax is:

scanf("format string" , list of address of variables) ;

ਜਿਥੇ ਸਟਰਿੰਗ-ਫਾਰਮੇਟ (string format) ਵਿਚ ਫਾਰਮੇਟ (format) ਵਿਸ਼ਿਸ਼ਤਾਵਾਂ (specifier) ਸ਼ਾਮਲ ਹੋਣ ਉਹ ਚਿੰਨ੍ਹ % ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦੇ ਹਨ ਅਤੇ ਕੌਮੇ ਜਾਂ ਸਪੇਸ ਦੇ ਨਾਲ ਵੱਖ ਹੁੰਦੇ ਹਨ। ਇਨ੍ਹਾਂ ਨੂੰ ਦੂਹਰੇ quotes i.e. " " ਨਾਲ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।

ਵੈਰੀਏਬਲਜ਼ (variables) ਦੇ ਪਤਿਆਂ (addresses) ਦੀ ਸੂਚੀ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਤਾਂ ਜੋ scanf() ਫੰਕਸ਼ਨ (function) ਸਟੈਂਡਰਡ ਇਨਪੁਟ (input) ਯੰਤਰ ਭਾਵ ਕੀ-ਬੋਰਡ ਤੋਂ ਪ੍ਰਾਪਤ ਹੋਏ ਡਾਟਾ ਨੂੰ ਰਖ ਸਕੇ । ਇਕ ਵੈਰੀਏਬਲ (variable) ਦਾ ਐਡਰੇਸ (address) ਓਪਰੇਟਰ, ਚਿੰਨ੍ਹ "&" ਦੀ ਵਰਤੋਂ ਨਾਲ ਲਗਦਾ ਹੈ ਅਤੇ ਇਹ ਪੁਆਇੰਟਰ ਆਪਰੇਟਰ ਦੇ ਪਤੇ ਵਾਂਗ ਉਚਾਰਿਆ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ 5 : **scanf("%d %c" , &bp, &city)**

ਇਥੇ scanf ਫੰਕਸ਼ਨ (function) ਚਾਰ ਮਾਪਦੰਡਾ ਨੂੰ ਨਿਰਦੇਸ਼ ਦਿੰਦੇ ਹਨ, ਮਾਪਦੰਡ **"%d"** ਚਿੰਨ੍ਹ ਦਾ ਅਰਥ ਹੈ ਕਿ ਇਕ ਪੂਰਣ ਸੰਖਿਆ ਪੜ੍ਹੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ। ਤੀਜਾ ਮਾਪਦੰਡ **&bp** ਦਸਦਾ ਹੈ ਕਿ ਪੂਰਣ ਸੰਖਿਆ ਮੁੱਲ ਜੋ ਸਵੀਕਾਰ ਕੀਤਾ ਗਿਆ ਹੈ ਵੈਰੀਏਬਲ (variable) ਨਾਮ bp ਵਿਚ ਸਟੋਰ ਹੋਵੇਗਾ। ਦੂਜਾ **"%c"** ਪੜ੍ਹਣ ਅਤੇ **"&city"** ਕਰੈਕਟਰ ਟਾਈਪ ਡਾਟਾ ਵੈਰੀਏਬਲ (variable) ਵਿਚ ਸਟੋਰ ਹੁੰਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਲਾਗੂ-ਕਰਨ (execution) ਤੇ, ਇਨਪੁਟ (input) ਡਾਟਾ ਨਿਸ਼ਚਿਤ ਸਟਰਿੰਗ-ਫਾਰਮੇਟ (string format) ਤੇ ਸਖਤੀ ਨਾਲ ਸਪਲਾਈ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ ਵਰਨਾ ਆਊਟਪੁਟ ਗਲਤ ਹੋਣਗੇ ।

ਮਿਸਾਲ 6 :

```
int x;
float y;
char c;
scanf("%d,%f,%c", &a,&b,&c);

Assume that the input value for a = 10
                                b = 32.65
                character       c = Z
```

ਡਾਟਾ ਨਿਮਨਲਿਖਿਤ ਫਾਰਮੇਟ (format) 10, 32, 65, Z ਵਿਚ ਟਾਈਪ ਕੀਤੇ ਜਾਣੇ ਚਾਹੀਦੇ ਹਨ। ਪਰੰਤੂ ਜੇਕਰ ਇਹ 10, 32, 65 Z (ਬਿਨਾਂ ਸਪੇਸ ਦੇ) ਵਜੋਂ ਪ੍ਰਵੇਸ਼ (enter) ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਤਾਂ ਆਊਟਪੁਟ ਗਲਤ ਹੋਣਗੇ ।

ਮਿਸਾਲ 7 :

- scanf("%d", &int_variable);
  If the input value is 55.75

  The scanf() function will take it as 55 only

- scanf("%f",&float_variable);
  If the input value is 45

  The scanf( ) function will take it as 45.0

ਆਓ ਹੁਣ ਪੰਨੇ 23 ਤੇ ਦਿਤੀ ਗਈ **ਮਿਸਾਲ 2.3** ਵਿਚ ਸੋਧ ਕਰੀਏ। ਹੁਣ ਇਹ ਯੂਜ਼ਰ ਨੂੰ ਇਨਪੁਟ (input) ਮੁੱਲ ਪੁੱਛੇਗਾ ਤਦ ਇਹ ਮੂਲਧਨ (p), ਸਮਾਂ (t) ਅਤੇ ਦਰ (r) ਦੇ ਮੁੱਲ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇਗਾ ਜੋ ਕਿ ਇਕ ਨਵੀਂ ਰੇਖਾ ਰਾਹੀਂ ਵੱਖਰੇ ਹੋਣਗੇ ।

ਸਪਸ਼ਟੀਕਰਨ (Explanation): ਚਿੰਨ੍ਹ '\n' ਨਵੀਂ ਰੇਖਾ ਸ਼ਾਮਲ ਕਰਦੀ ਹੈ ਤਾਂ ਜੋ ਬਾਅਦ ਦਾ ਆਊਟਪੁਟ (output) ਅਗਲੀ ਰੇਖਾ ਤੇ ਜਾਰੀ ਰਹੇ। ਇਸ ਲਈ string ਆਊਟਪੁਟ (output) ਇਸ ਅਵਸਥਾ ਤੇ ਇਸ ਪ੍ਰਕਾਰ ਜਾਪੇਗਾ :

```
/* calculating the simple interest */
void main( )
{
int p,t ;
float r,si ;
printf("\n Enter value of principal p , time t, rate r \n");
scanf("%d %d %f", &p,&t,&r);
si = p*r*t/100;
printf("%f",si);
}
```

Enter value of principal p, time t and rate r
1000
3
5.5

ਆਊਟਪੁਟ (Output)

165.000000

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਇਸ ਉਪਰੰਤ ਇਹ ਇਨਪੁਟ (input) ਮੁੱਲ ਉਨ੍ਹਾਂ ਦੀਆਂ ਸੰਬੰਧਿਤ ਵੇਰੀਏਬਲਜ਼ (variables) ਦੇ ਸਪੁਰਦ ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਜੋ ਕਿ ਅਨੁਪਾਤ si = (p*r*t)/100 ਵਿਚ ਹੁੰਦੇ ਹਨ, ਜਿਥੇ si ਦੀ ਗਿਣਤੀ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਫਿਰ ਇਹ ਅੰਤ ਵਿਚ printf() ਰਾਹੀਂ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਮਿਸਾਲ 8 :

```
// This program accepts the temperature in Fahrenheit and
// converts it into celsius , formula is C = (f-32.0)/18
#include <stdio.h>
void main()
{float ct, ft;
printf("enter the temperature in  Fahrenheit \n");
scanf("%f",&ft);
ct = (ft -32.0)/18;
printf(" Fahrenheit temperature  = %6.2f\n" , ft);
printf(" Celsius temperature = %6.2f\n" , ct);
}
```

ਆਊਟਪੁਟ (Output)

```
enter the temperature in Fahrenheit
34.20
Fahrenheit temperature
34.20
Celsius temperature
0.12
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

2.7.0 ਸੀ(C) ਪ੍ਰੋਗਰਾਮਿੰਗ ਨਾਲ ਸ਼ੁਰੂਆਤ ਕਰਨੀ (Starting with C Programming)

ਇਥੇ ਅਸੀ ਵਿੰਡੋਜ਼ ਓਪਰੇਟਿੰਗ ਸਿਸਟਮ (Windows operating system) ਅਤੇ ਕੀ-ਬੋਰਡ ਇਸਤੇਮਾਲ ਕਰਾਂਗੇ ।

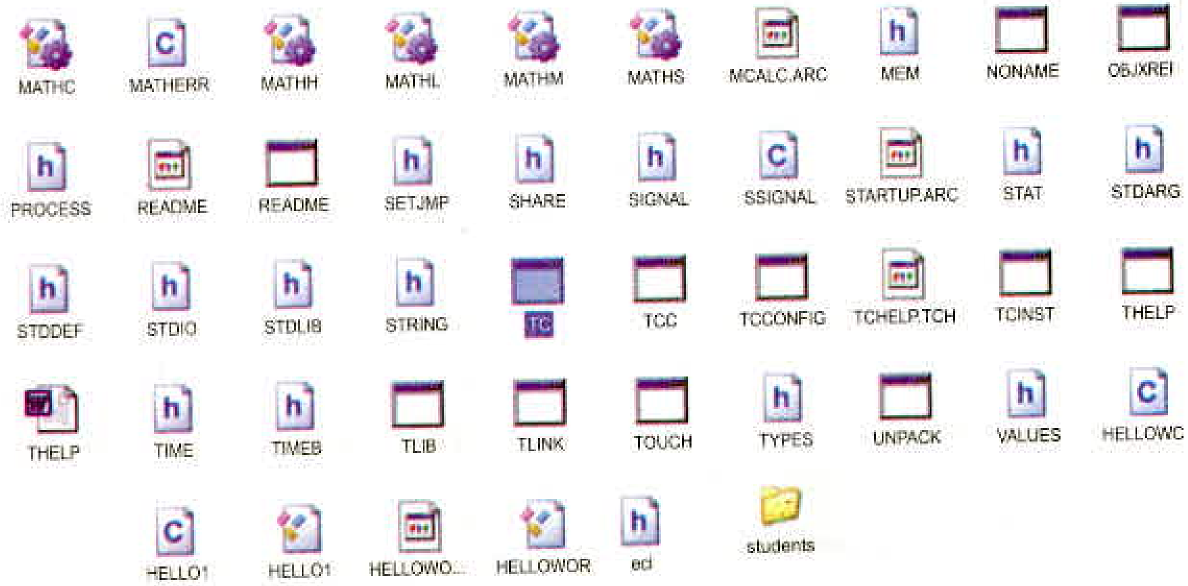
ਵਿੰਡੋਸ ਦੇ ਪਲੇਟਫਾਰਮ ਤੇ ਅਸੀ ਟਰਬੋ ਸੀ (Turbo C) ਦਾ ਇਸਤੇਮਾਲ ਕਰ ਰਹੇ ਹਾ, ਨਿਮਨਲਿਖਤ ਕਦਮ ਤੁਹਾਨੂੰ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ enter ਅਤੇ ਕੰਪਾਈਲ (compile) ਕਰਨ ਲਈ ਸਹਾਇਤਾ ਕਰਨਗੇ ।

2.7.1 ਟਰਬੋ ਸੀ ਨੂੰ ਸਥਾਪਿਤ ਕਰਨਾ (Installing Turbo C)

ਤਸਵੀਰ ਜੋ ਟਰਬੋ ਸੀ (Turbo C) ਫੋਲਡਰ ਨੂੰ ਸੀ-ਡੀ/ਇੰਟਰਨੈਟ ਤੋਂ ਉਤਾਰਣ ਦੇ ਬਾਅਦ ਡੈਸਕਟੋਪ ਤੇ ਦਿਖਦੀ ਹੈ

ਟਰਬੋ ਸੀ (Turbo C) ਫੋਲਡਰ ਖੋਲਣ ਤੋਂ ਬਾਦ ਨਿਮਨਲਿਖਤ ਦੋ ਚਿੰਨ (icon) ਦਿਖਾਈ ਦੇਣਗੇ

ਸੀ (C) ਡਰਾਈਵ ਵਿਚ ਜਾਓ ਜਿਥੇ tctemp ਨਾਮ ਦਾ ਫੋਲਡਰ ਖੋਲਿਆ ਗਿਆ ਸੀ। tctemp ਫੋਲਡਰ ਨੂੰ ਖੋਲ੍ਹੋ ਅਤੇ tc.exe ਤੇ ਕਲਿਕ ਕਰੋ ਜਿਵੇਂ ਤਸਵੀਰ ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ

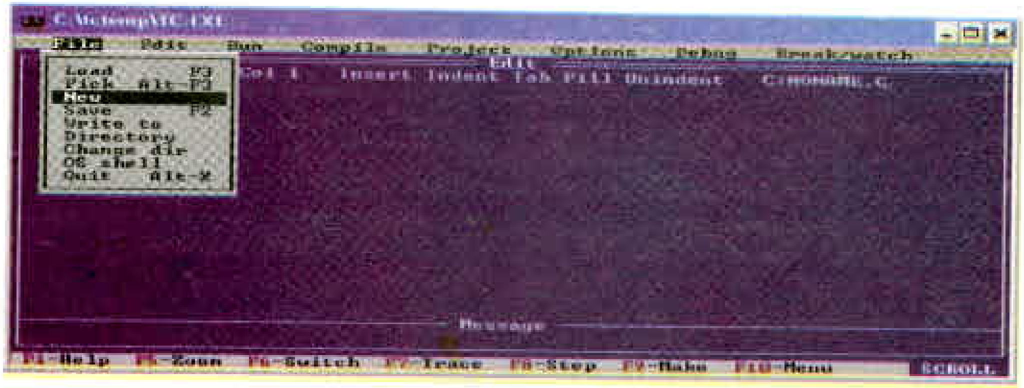


1. TC executive ਫਾਈਲ (tc.exe) ਤੇ ਦੋ-ਵਾਰੀ (double) ਕਲਿੱਕ ਕਰੋ, ਇਹ ਵਿੰਡੋਜ਼ (Windows) ਨੂੰ ਖੋਲ ਦੇਵੇਗਾ ਜਿਸ ਤਰ੍ਹਾਂ ਤਸਵੀਰ ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

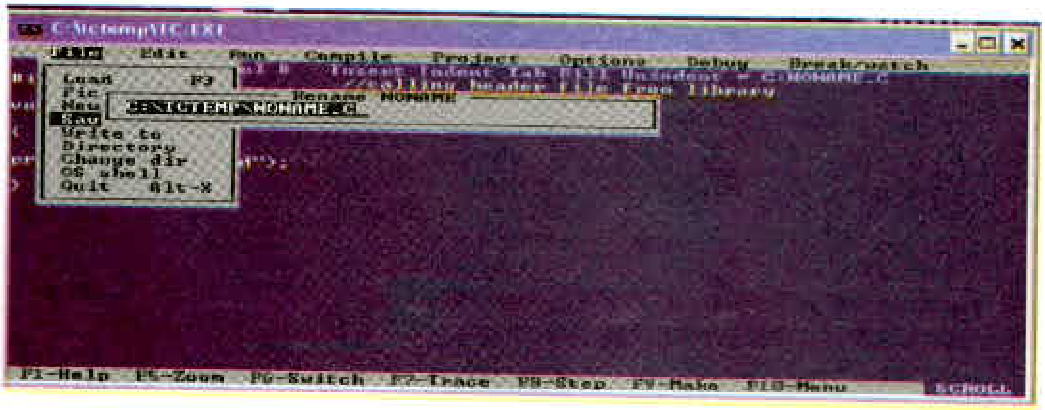
2. ਹੁਣ Alt +F ਨੂੰ ਦਬਾ ਕੇ ਫਾਈਲ ਮੀਨੂੰ ਨੂੰ ਖੋਲੋ ਅਤੇ ਨਵੇਂ ਪ੍ਰੋਗਰਾਮ ਲਈ New ਵਿਕਲਪ ਨੂੰ ਚੁਣੋ



ਹੁਣ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜਿਸ ਤਰ੍ਹਾਂ ਹੇਠਾਂ ਤਸਵੀਰ ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ

ਪ੍ਰੋਗਰਾਮ ਲਿਖਣ ਤੋਂ ਬਾਅਦ ਪ੍ਰੋਗਰਾਮ ਨੂੰ Save ਕਰਨ ਲਈ Alt + S ਨੂੰ ਦਬਾਉਂਦੇ ਹੋਏ ਮੀਨੂੰ ਵਿਚ Save ਵਿੱਕਲਪ ਚੁਣੋ ਅਤੇ ਐਂਟਰ (enter) ਬਟਨ ਨੂੰ ਦਬਾਓ ।

ਟਿਪਣੀ : ਇਹ ਐਕਸਟੈਂਸ਼ਨ ਸੀ (extension C) ਹਰ ਪ੍ਰੋਗਰਾਮ ਨਾਲ ਆਪਣੇ ਆਪ ਆਉਂਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਆਪਣੇ ਫੋਲਡਰ ਵਿਚ ਜਿਸ ਤਰ੍ਹਾਂ ਹੇਠਾਂ ਤਸਵੀਰ ਵਿਚ ਦਿਖਾਏ ਗਏ ਫੋਲਡਰ ਨਾਮ ਸਟੂਡੈਂਟ (student) ਜਾਂ ਟਰਬੋ ਸੀ (Turbo C) ਫੋਲਡਰ ਵਿਚ ਵੀ Save ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ

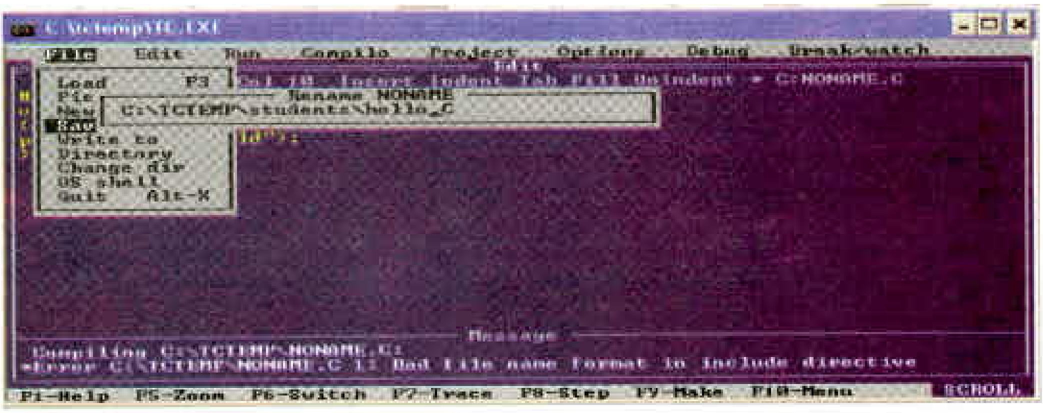


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Default Saving ਹੇਠ ਤਸਵੀਰ ਵਿਚ ਦਿਖਾਈ ਗਈ ਹੈ।

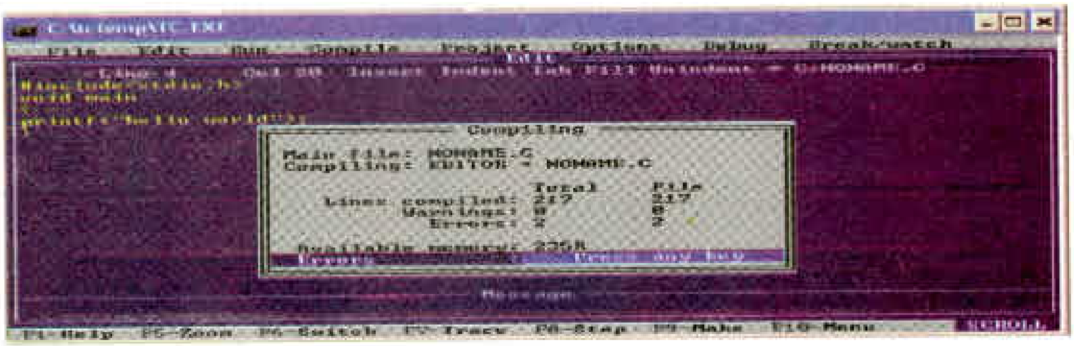
ਵਰਤੋਂ ਕਰਨ ਵਾਲੇ ਦੇ ਫੋਲਡਰ "Student" ਵਿਚ



2.7.2 ਪ੍ਰੋਗਰਾਮ ਦਾ ਕੰਪਾਇਲਿੰਗ ਅਤੇ ਐਗਜ਼ਿਕਿਊਟਿੰਗ (Compiling the program and executing the program)

ਪ੍ਰੋਗਰਾਮ ਕੰਪਾਇਲਿੰਗ (Compiling) ਲਈ ਦਿੱਤੇ ਗਏ ਆਦੇਸ਼ਾਂ ਤੇ ਅਮਲ ਕਰੋ

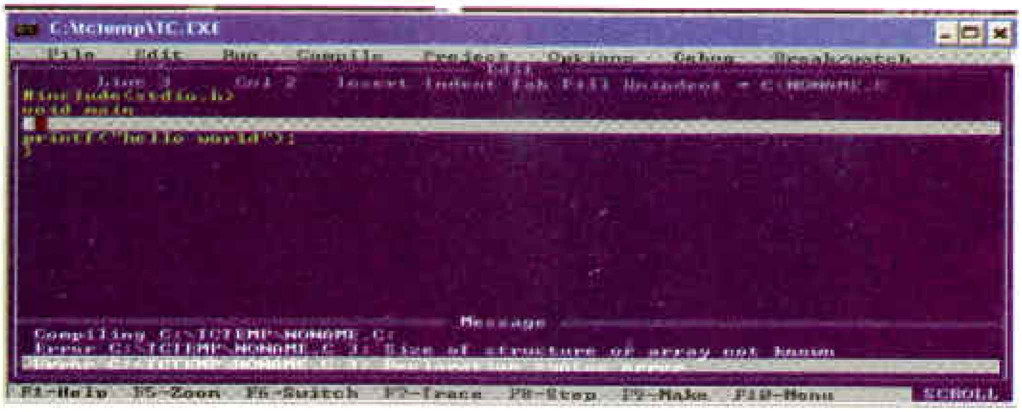
Alt + F9 ਨੂੰ ਇਕਠੇ ਦਬਾਓ ਜੇਕਰ ਕੰਪਾਇਲਰ (compiler) ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕੁਝ ਗਲਤੀਆਂ ਮਿਲਦੀਆਂ ਹਨ ਤਾਂ ਇਹ ਨਿਮਨਲਿਖਤ ਗਲਤੀਆਂ ਦੇ ਸੰਦੇਸ਼ ਸਕਰੀਨ ਤੇ ਦਿਖਣਗੇ ਜਿਸ ਤਰ੍ਹਾਂ ਕਿ ਹੇਠ ਤਸਵੀਰ ਵਿਚ ਦਰਸਾਏ ਗਏ ਹਨ।



29

Downloaded from https:// www.studiestoday.com

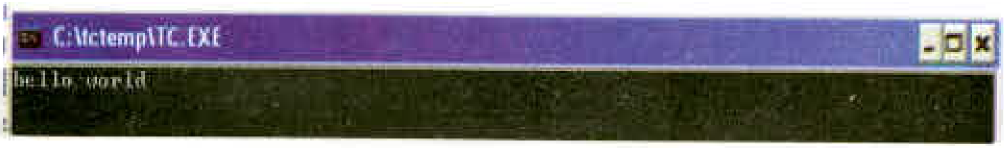
Downloaded from https:// www.studiestoday.com



ਮਿਸਾਲ ਦੇ ਤੌਰ ਤੇ ਉਪਰੋਕਤ ਤਸਵੀਰ ਵਿਚ ਦੋ ਗਲਤੀਆਂ ਦਰਸਾਈਆਂ ਗਈਆਂ ਹਨ। ਕਿਉਂਕਿ ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ main ਇਕ ਫੰਕਸ਼ਨ ਹੈ ਜਿਸ ਨਾਲ ਬਰੈਕਟਾਂ ( ) ਲਾਉਣਿਆਂ ਜਰੂਰੀ ਹਨ, ਜੋ ਕਿ ਉਪਰ ਲਿਖੇ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਨਹੀਂ ਹਨ।

ਇਸ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸਫਲਤਾ ਨਾਲ ਚਲਾਉਣ ਲਈ ਇਹਨਾਂ ਗਲਤੀਆਂ ਦਾ ਸੁਧਾਰ ਕਰੋ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਦੁਬਾਰਾ ਕੰਪਾਈਲ (Compile) ਕਰੋ ਜੇਕਰ ਕੋਈ ਗਲਤੀ ਨਾ ਹੋਵੇ ਤਾਂ ਇਹ ਹੇਠਾਂ ਦਿੱਤੀ ਤਸਵੀਰ ਵਾਂਗ ਦਿਖੇਗਾ। ਇਸ ਦਾ ਮਤਲਬ ਪ੍ਰੋਗਰਾਮ ਹੁਣ ਚਲਣ ਲਈ ਤਿਆਰ ਹੈ

ਪ੍ਰੋਗਰਾਮ ਦਾ ਨਤੀਜਾ ਦੇਖਣ ਲਈ **Alt + F5** ਨੂੰ ਦਬਾਓ, ਇਹ ਸਕਰੀਨ (screen) ਤੇ ਤੁਹਾਨੂੰ ਨਤੀਜਾ ਹੇਠਾਂ ਦਰਸਾਈ ਤਸਵੀਰ ਵਾਂਗ ਦਿਖੇਗਾ



ਜੇਕਰ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸਕਰੀਨ (screen) ਤੇ ਬਿਨਾਂ ਸਾਫ ਕਿਤੇ ਦੂਜੀ ਵਾਰ ਚਲਾਇਆ ਜਾਵੇਗਾ ਤਾਂ ਆਉਟਪੁਟ ਹੇਠਾਂ ਦਿਖਾਈ ਤਸਵੀਰ ਵਾਂਗ ਦਿਖੇਗਾ

ਇਸ ਲਈ ਹੈਡਰ ਫਾਈਲ <conio.h> ਅਤੇ clrscr( ) ਨੂੰ stdio.h ਨਾਲ ਵੇਰੀਏਬਲਜ਼ (variables) ਦੇ ਸ਼ੁਰੂਆਤ ਤੋਂ ਬਾਅਦ ਸ਼ਾਮਿਲ ਕਰੋ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਸੀ ਭਾਸ਼ਾ ਵਿਚ ਪ੍ਰੋਗਰਾਮਿੰਗ ਲਈ ਭੂਮਿਕਾ (Introduction to Programming in C Language)

C ਇਕ ਆਮ ਮੰਤਵ ਦੀ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਹੁੰਦੀ ਹੈ। ਇਹ ਦੋ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਜਿਵੇਂ B ਭਾਸ਼ਾ ਅਤੇ BCPL ਦਾ ਪਰਿਣਾਮ *(outcome)* ਹੁੰਦੀ ਹੈ। ਸਮੂਹ ਤਿੰਨ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਬੈਲ ਲੈਬਾਰਟਰੀ USA ਵਿਚ ਵਿਕਸਿਤ ਹੋਈਆਂ ਸਨ। ਇਹ 1978 ਵਿਚ ਡੇਨਿਸ ਰਿਚ ਅਤੇ ਬਰੇਨ ਕੇਰਮਿੰਘਨ *(Dennis Ritchie & Brain Kerninghan)* ਦੁਆਰਾ ਲਿਖੀਆਂ ਗਈਆਂ ਸਨ। **C** ਦੋਵੇਂ ਸਿਸਟਮ ਪ੍ਰੋਗਰਾਮ ਅਤੇ ਐਪਲੀਕੇਸ਼ਨ-ਪ੍ਰੋਗਰਾਮ *(application program)* ਵਿਕਾਸ ਉਪਕਰਣ ਹਨ। ਇਹ ਹਾਈ ਅਤੇ ਲੋਅ ਲੈਵਲ ਦੇ ਦਰਮਿਆਨੀ ਭਾਸ਼ਾ ਹੁੰਦੀ ਹੈ ਇਸ ਲਈ ਇਸ ਨੂੰ ਦਰਮਿਆਨੀ-ਲੈਵਲ *(middle level)* ਭਾਸ਼ਾ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। C ਚਿੰਨ੍ਹ ਸੈੱਟ *(character set)* ਵਰਣਮਾਲਾ *(alphabets)*, ਹਿੰਦਸਿਆਂ *(digits)* ਅਤੇ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹਾਂ *(special symbols)* ਦੇ ਬਣੇ ਹੁੰਦੇ ਹਨ। C ਵਿਚ 91 ਚਿੰਨ੍ਹ *(character)* ਹੁੰਦੇ ਹਨ। C ਟੋਕਣ *(token)* C ਪ੍ਰੋਗਰਾਮ ਦੇ ਬੁਨਿਆਦੀ ਯੂਨਿਟ ਹੁੰਦੇ ਹਨ ਅਤੇ ਉਹ ਕੀ-ਵਰਡਸ, ਆਈਡੈਂਟੀਫਾਈਰਸ *(identifiers)*, ਕੌਨਸਟੈਂਟ ਸਟਰਿੰਗਜ਼ *(constants strings)*, ਆਪਰੇਟਰਾਂ ਅਤੇ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹਾਂ *(special symbols)* ਵਿਚ ਸ਼੍ਰੇਣੀਬੱਧ ਹੁੰਦੇ ਹਨ। C ਵਿਚ ਸਾਰੇ ਕੀ-ਵਰਡਜ਼ ਦੇ ਇਕ ਜਾਂ ਵਧੇਰੇ ਨਿਸ਼ਚਿਤ ਅਰਥ ਹੁੰਦੇ ਹਨ ਅਤੇ ਉਹ ਬਦਲੇ ਨਹੀਂ ਜਾ ਸਕਦੇ। ਕੀ-ਵਰਡਾਂ ਛੋਟੇ ਅੱਖਰਾਂ ਵਿਚ ਲਿਖੇ ਜਾਣੇ ਚਾਹੀਦੇ ਹਨ। ਆਈਡੈਂਟੀਫਾਈਰਸ *(identifiers)* ਨਹੀਂ ਬਦਲੇ ਜਾ ਸਕਦੇ। ਆਈਡੈਂਟੀਫਾਈਰਸ *(identifiers)* ਉਹ ਨਾਮ ਹਨ ਜੋ ਪ੍ਰੋਗਰਾਮ ਅੰਸ਼ਾਂ ਨੂੰ ਦਿਤੇ ਜਾਂਦੇ ਹਨ ਜਿਵੇਂ ਕਿ ਵੇਰੀਏਬਲਜ਼ *(variables)*, ਐਰੇਸ *(arrays)* ਅਤੇ ਫੰਕਸ਼ਨਜ਼ *(functions)*। ਸਮੂਹ ਵੇਰੀਏਬਲਸ *(variables)* ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਵਰਤਣ ਤੋਂ ਪਹਿਲਾਂ ਉਨ੍ਹਾਂ ਦੀਆਂ ਕਿਸਮਾਂ ਲਈ ਡਿਕਲੇਅਰ *(declare)* ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਕੌਨਸਟੈਂਟ *(constant)* ਇਕ ਮਾਤਰਾ ਹੁੰਦੀ ਹੈ ਜਿਹੜੇ ਇਕ ਪ੍ਰੋਗਰਾਮ ਲਾਗੂਕਰਨ *(execution)* ਦੇ ਦੌਰਾਨ ਨਹੀਂ ਬਦਲਦੇ ਜਦ ਕੀ ਫਲੋਟਿੰਗ *(floating)* ਬਦਲ ਜਾਂਦੇ ਹਨ। ਹਰ ਇਕ ਹਿਦਾਇਤ ਅਰਧਵਿਰਾਮ-ਚਿੰਨ੍ਹ *(semi colon character)* ";" ਦੇ ਨਾਲ ਖਤਮ ਹੁੰਦੀ ਹੈ ਅਤੇ ਟਿੱਪਣੀਆਂ *(comments)* ਕਿਤੇ ਵੀ ਰਖੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ। ਇਕ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਸਮੂਹ ਸ਼ਬਦ ਟੈਬ ਜਾਂ ਵਿਰਾਮ ਚਿੰਨ੍ਹ *(punctuation character)* ਘਟੋ ਘਟ ਇਕ space/tab ਦੁਆਰਾ ਇਕ ਦੂਜੇ ਤੋਂ ਵੱਖ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਹਰ ਇਕ C ਪ੍ਰੋਗਰਾਮ ਵਿਚ main( ) ਫੰਕਸ਼ਨ *(function)* ਅਤੇ ਉਸ ਸਥਾਨ ਜਿਥੇ ਪ੍ਰੋਗਰਾਮ ਲਾਗੂ-ਕਰਨ *(execution)* ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ ਦੀ ਲੋੜ ਉਥੇ ਹੁੰਦੀ ਹੈ।

ਫੰਕਸ਼ਨ (function): ਫੰਕਸ਼ਨ ਦਾ ਲਾਗੂ-ਕਰਨ-ਫੰਕਸ਼ਨ *(execution function)* ਦਾ *opening brace* ਤੋਂ ਸ਼ੁਰੂ ਅਤੇ *closing brace* ਤੇ ਖਤਮ ਹੁੰਦਾ ਹੈ। ਸਾਨੂੰ ਇਸ ਗੱਲ ਨੂੰ ਯਕੀਨੀ ਬਣਾਉਣਾ ਚਾਹੀਦਾ ਹੈ ਕਿ # ਦੀ ਵਰਤੋਂ ਨਾਲ ਹੈਡਰ ਫਾਈਲਾਂ ਸ਼ਾਮਲ ਕੀਤੀਆਂ ਜਾਣ ਜਦੋਂ ਕਿ ਪ੍ਰੋਗਰਾਮ ਵਿਸ਼ੇਸ਼ ਨਾਵਾਂ ਅਤੇ ਫੰਕਸ਼ਨਸ *(functions)* ਦਾ ਹਵਾਲਾ ਦਿੰਦਾ ਹੋਵੇ ਕਿ ਇਹ ਪਰਿਭਾਸ਼ਤ *(define)* ਨਹੀਂ ਕਰ ਸਕਦਾ ।

ਕੰਪਾਈਲਰ-ਨਿਰਦੇਸ਼ (Compiler directives) :ਜਿਵੇਂ ਕਿ ਪਰਿਭਾਸ਼ਾ *(definition)* ਅਤੇ ਵਿਸ਼ੇਸ਼ ਹਿਦਾਇਤਾਂ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ ਜੋ ਕੰਪਾਈਲਰ *(Compiler)* ਨੂੰ ਇਕ ਪ੍ਰੋਗਰਾਮ ਦਾ ਕੰਪਾਈਲ *(Compile)* ਕਰਨ ਲਈ ਮਦਦ ਕਰਦੇ ਹਨ। ਉਹ ਅਰਧ ਵਿਰਾਮ *(semi colon)* ਦੇ ਨਾਲ ਖਤਮ ਨਹੀਂ ਹੁੰਦੇ। ਚਿੰਨ੍ਹ # ਜੋ ਕੰਪਾਈਲਰ *(compiler)* ਦਾ ਹੁੰਦਾ ਹੈ ਲਾਈਨ ਦੇ ਪਹਿਲੇ ਕਾਲਮ ਉਤੇ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। ਜਦੋਂ ਬਰੇਸਿਸ ਨੂੰ ਸਟੇਟਮੈਂਟ *(statement)* ਬਣਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਸ ਗੱਲ ਨੂੰ ਯਕੀਨੀ ਬਣਾਓ ਕਿ *opening brace* ਦਾ ਪੂਰਕ *closing brace* ਹੋਵੇ। ਇਨਪੁਟ ਡਿਵਾਈਸ *(input device)* ਤੋਂ ਡਾਟਾ ਪੜ੍ਹਣ ਲਈ ਇਨਪੁਟ ਸਟੇਟਮੈਂਟ *(input statement)* ਵਰਤੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਪ੍ਰੋਸੈਸ ਡਾਟਾ ਨੂੰ ਇਨਪੁਟ ਡਿਵਾਈਸ *(input device)* ਵਿਚ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਆਊਟਪੁਟ ਡਿਵਾਈਸ *(output device)*, ਆਊਟਪੁਟ ਸਟੇਟਮੈਂਟ *(output statement)* ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਇਨਪੁਟ/ ਆਊਟਪੁਟ (I/O) ਸਟੇਟਮੈਂਟ ਫਾਰਮੈਟ *(statement format)* ਜਾਂ ਗੈਰ-ਫਾਰਮੈਟ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਗੈਰ-ਫਾਰਮੈਟਿਡ ਇਨਪੁਟ/ ਆਊਟਪੁਟ (I/O) ਸਟੇਟਮੈਂਟਸ *(statements)* ਵਿਚ ਡਾਟਾ ਟਾਈਪ ਦੀ ਕੋਈ ਵਿਸ਼ੇਸ਼ਤਾ ਨਹੀਂ ਹੁੰਦੀ ਅਤੇ ਜਿਸ ਢੰਗ ਨਾਲ ਉਹ ਪੜ੍ਹੇ ਜਾਂਦੇ ਹਨ ਜਾਂ ਲਿਖੇ ਜਾਂਦੇ ਹਨ ਦਾ ਵਰਣਨ ਨਹੀਂ ਕੀਤਾ ਗਿਆ ਹੁੰਦਾ। ਪਰੰਤੂ ਫਾਰਮੈਟਿਡ ਇਨਪੁਟ/ ਆਊਟਪੁਟ (I/O) ਸਟੇਟਮੈਂਟਸ *(statements)* ਇਹ ਨਿਸ਼ਚਿਤ ਕਰਦਾ ਹੈ ਕਿ ਇਸ ਟਾਈਪ ਦਾ ਡਾਟਾ ਇਨਪੁਟ ਹੈ ਜਾਂ ਉਸ ਢੰਗ ਨਾਲ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਜਾ ਰਿਹਾ ਹੈ ਜਿਸ ਨਾਲ ਇਹ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। scanf( ) ਅਤੇ printf( ) ਦੇ ਫੰਕਸ਼ਨਸ *(functions)* ਹਨ ਜੋ ਡਾਟਾ ਪੜ੍ਹਨ ਅਤੇ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਉਹ ਹਰ ਟਾਈਪ ਦੇ ਡਾਟਾ ਦੀਆਂ ਕਿਸਮਾਂ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ ਜਿਵੇਂ int, char, float, double, ਆਦਿ। ਆਮ ਤੌਰ ਇਹ ਸੰਭਵ ਨਹੀਂ ਹੈ ਕਿ ਬਹੁ-ਸ਼ਬਦੀ ਸਟਰਿੰਗ ਨੂੰ scanf () ਫੰਕਸ਼ਨ ਨਾਲ ਪੜ੍ਹਿਆ ਜਾਵੇ। ਐਸਕੇਪ-ਸਾਰ (escape sequence) ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਸੀਂ ਆਪਣੀਆਂ ਜਰੂਰਤਾਂ ਅਨੁਸਾਰ ਆਊਟਪੁਟ ਨਿਰਮਿਤ ਕਰ ਸਕਦੇ ਹਾਂ ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1: ਬਹੁਪੱਖੀ ਪਸੰਦ ਦੇ ਪ੍ਰਸ਼ਨ

1.1 ਇਸ ਹਿਦਾਇਤ ਨੂੰ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਅੰਤ ਕਰਨ ਲਈ ਕਿਹੜਾ ਨਿਮਨਲਿਖਤ ਚਿੰਨ੍ਹ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ
a), (comma) b): (colon) c) ;(semicolon) d). (period)

1.2 ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕਿੰਨੇ ਮੇਨ ਫੰਕਸ਼ਨ (main function) ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ ?
a) 2 b) 3 c) 1 d) any number

1.3 ਇਕ ਫੰਕਸ਼ਨ ਦੀ ਬਾਡੀ (body) ਨੂੰ ਸ਼ਾਮਲ ਕਰਨ ਲਈ ਨਿਮਨਲਿਖਤ ਵਿਚੋਂ ਕਿਸ ਦਾ ਇਸਤੇਮਾਲ ਹੁੰਦਾ ਹੈ
a) [ ] b) { } c) ( ) d) None of these

1.4 ਸੀ (C) ਭਾਸ਼ਾ ਦੇ ਬਲਾਕ (block) ਬਣਾਉਣ ਲਈ ਸਾਨੂੰ ਲੋੜ ਹੁੰਦੀ ਹੈ ?
a) Characters b) Data type c) names d) expression & statements e) all of these

1.5 ਸੀ (C) ਵਿਚ ਕਿਹੜਾ ਕੀ-ਵਰਡ (key-word) ਨਹੀਂ ਹੁੰਦਾ ?
a) const. b) main c) sizeof d) void

1.6 ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਕਿਹੜਾ ਆਪਸ਼ਨ (option) ਸਭ ਤੋਂ ਉੱਤਮ ਹੁੰਦਾ ਹੈ ?
(ਏ) ਉੱਚ ਨਿਪੁੰਨਤਾ ਵਾਲਾ (ਬੀ) ਸਿਸਟਮ ਸੁਤੰਤਰ
(ਸੀ) ਸੀਮਿਤ ਡਾਟਾ-ਟਾਈਪਸ (ਡੀ) ਇਹ ਸਭ ਰਖਦਾ ਹੋਵੇ

1.7 ਗਣਿਤ ਲਾਇਬ੍ਰੇਰੀ ਦੀ ਸਥਾਪਨਾ ਇਕ ਯੂਜ਼ਰ ਵੱਲੋਂ ਕਿਸ ਫਾਈਲ ਦੁਆਰਾ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ?
a) float.h b) limits.h c) math.h d) time.h

1.8 ਕਿਹੜਾ printf ਵੈਰੀਏਬਲ (variable) ਚਿੰਨ੍ਹ ਇਕ ਸਾਈਨਡ (signed) ਦਸ਼ਮਲਵ ਪੂਰਣ-ਅੰਕ (integer) ਵਜੋਂ ਡਾਟਾ-ਆਈਟਮ (data item) ਪ੍ਰਦਰਸ਼ਿਤ (display) ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ?
a) d b) f c) i d) u

1.9 ਖਾਲੀ ਚਿੰਨ੍ਹ (null character) ਦੀ ਪ੍ਰਤਿਨਿਧਤਾ (representation) ਕਿਸ ਵਲੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ?
a) \n b) \0 c) \o d) \t

1.10 ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਨਿਮਨਲਿਖਤ ਵਿਚੋਂ ਕਿਸ ਦਾ ਇਸਤੇਮਾਲ ਨਹੀਂ ਹੁੰਦਾ ਹੈ ?
a) a digit b) an integer c) a character d) a word

ਪ੍ਰਸ਼ਨ 2: ਨਿਮਨਲਿਖਿਤ ਦਾ ਸਟੇਟਮੈਂਟ ਲਿਖੋ

2.1 ਨਿਮਨਲਿਖਿਤ ਸਟੇਟਮੈਂਟ (statement) ਲਾਗੂ ਕਰਨ ਦਾ ਕੀ ਪ੍ਰਭਾਵ ਹੁੰਦਾ ਹੈ ?

```
printf("\nOne\nTwo\nThree\n");
```

2.2 printf ਸਟੇਟਮੈਂਟ (statement) ਲਿਖੋ, ਜਿਹੜਾ ਤੁਹਾਡੀ ਸਕਰੀਨ ਤੇ ਨਿਮਨਲਿਖਤ ਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰੇ ?

This is a test 1 2 3
123.33
These are the Characters a b c

2.3 ਸਟੇਟਮੈਂਟ (statement) ਲਿਖੋ, ਜਿਹੜੇ ਨਿਮਨਲਿਖਤ ਵੈਰੀਏਬਲਜ਼ (variables) ਨੂੰ ਸੂਚਿਤ ਕੀਤੇ ਗਏ ਢੰਗ ਨਾਲ ਡਿਕਲੇਅਰ (declare) ਕਰੇ ?

up ,down	as integers
first , last	as single precision floating point
c	as a character

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

2.4 ਕੀ ਹੇਠ ਲਿਖਿਆ ਪ੍ਰੋਗਰਾਮ ਕੰਮ ਕਰੇਗਾ ? ਜੇ ਨਹੀਂ ਤਾਂ ਗਲਤੀਆਂ ਦੱਸੋ ਅਤੇ ਠੀਕ ਕਰੋ ?

```
#include<stdio.h>
main ( ) {printf("How many persons !);}
```

2.5 ਸੀ (C) ਕੋਡ ਦੇ ਨਿਮਨਲਿਖਤ ਭਾਗ ਨਾਲ ਕੀ ਗਲਤ ਹੈ ?

```
printf("c","o","m",p","u","t","e","r",\n");
```

2.6 ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਗਲਤੀਆਂ ਦਸੋ ?

```
name { }                    \* This is one great program ;stand buy .... /*
(
 print('I think I am getting hungry. \n');
```

2.7 **printf** ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਇਆਂ ਸੀ **(C)** ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜੋ ਨਾਮ ਨੂੰ ਇਕ ਲਾਈਨ ਤੇ, ਐਡਰੇਸ **(address)** ਦੂਜੀ ਲਾਈਨ ਅਤੇ ਸ਼ਹਿਰ **(city)**, ਰਾਜ **(state)** ਦਾ ਨਾਮ ਤੀਜੀ ਲਾਈਨ ਉਤੇ ਲਿਖੇ ?

2.8 **printf** ਦੀ ਵਰਤੋਂ ਕਰਕੇ - ਤਿੰਨ ਅੰਕਾਂ ਦੀ ਔਸਤ ਅਤੇ ਜੋੜ ਪ੍ਰਿੰਟ ਕਰੋ ?

2.9 ਮੀਟਰਾਂ ਨੂੰ ਕਿਲੋਮੀਟਰਾਂ ਵਿਚ ਬਦਲਣ ਲਈ ਪ੍ਰੋਗਰਾਮ ਵਿਚੋਂ ਗਲਤੀਆਂ ਨੂੰ ਸਹੀ ਕਰੋ ?

```
main( )
{ float mtrs , km;
 printf("Enter the value in meters ");
  scanf("%f" ,&mtr);
km = mtrs/1000.00
 printf("\n The value of metre(s) % 7.2f is converted into %7.2f kilometer (s) ', mtr,km); }
```

2.10 ਸੀ **(C) = (5.0/9.0)*(f 32.0);** ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਤਾਪਮਾਨ ਨੂੰ ਫਾਰਨਹੀਟ ਤੋਂ ਸੈਲਸਿਅਸ ਵਿਚ ਬਦਲੋ ?

Hint Step :

c and f are declared to be floats (c ਅਤੇ f ਅਸਥਾਈ ਡਿਕਲੇਅਰ (declare) ਕੀਤੇ ਗਏ ਹਨ)

Fahrenheit temperature is accepted (ਫਾਰਨਹੀਟ ਤਾਪਮਾਨ ਸਵੀਕ੍ਰਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ)

Computes the celcius value w.r.t. Fahrenheit (ਫਾਰਨਹੀਟ ਦੇ ਸੰਬੰਧ ਵਿਚ ਸੈਲਸਿਅਸ ਮੁੱਲ ਦੀ ਗਣਨਾ ਕਰੋ)

Prints the computed celcius. (ਸੈਲਸੀਅਸ ਗਣਨਾ ਪ੍ਰਿੰਟ ਕਰੋ)

ਪ੍ਰਸ਼ਨ 3: ਦਸੋ ਸਹੀ ਹੈ ਜਾਂ ਗਲਤ

1. ਸੀ (C) ਭਾਸ਼ਾ ਚੰਜ ਕੇਸ (change case) ਹੈ ।
2. ਜੇਕਰ ਇਕ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਲੌਜਿਕਲ (logical) ਗਲਤੀਆਂ ਹੋਣ ਤਾਂ ਇਹ ਕੰਪਾਈਲੇਸ਼ਨ (Compilation) ਨਹੀਂ ਕੀਤਾ ਜਾ ਸਕਦਾ ।
3. ਖਾਲੀ ਸਥਾਨ ਨੂੰ ਸਫੈਦ ਸਥਾਨ ਚਿੰਨ੍ਹ (white space character) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ।
4. Null string ਦੀ ਨੁਮਾਇੰਦਗੀ (representation) " " ਦੁਆਰਾ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ।
5. ਸੀ (C) ਇਕ ਸਧਾਰਣ (simple), ਬਹੁ-ਉਪਯੋਗੀ (versatile) ਅਤੇ ਵਧੇਰੇ ਸਪਸ਼ਟ ਆਮ-ਮੰਤਵ-ਭਾਸ਼ਾ (general purpose language) ਹੈ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰਸ਼ਨ 4: ਨਿਮਨਲਿਖਤ ਲਈ ਉਚਿਤ ਸ਼ਬਦ ਲਿਖੋ।

1. C ਭਾਸ਼ਾ ਕਿਸ ਲੈਵਲ-ਦੀ-ਭਾਸ਼ਾ (level language) ਹੁੰਦੀ ਹੈ।
2. C ਭਾਸ਼ਾ ਦੀ ਹਰ ਇਕ ਸਟੇਟਮੈਂਟ (statement) ਕਿਸ ਦੇ ਨਾਲ ਖਤਮ ਹੁੰਦੀ ਹੈ।
3. 0 ਮੁੱਲ ਦੇ ਨਾਲ ਚਿੰਨ੍ਹ ਹੁੰਦਾ ਹੈ।
4. \t , "C" ਵਿਚ ਵਿਸ਼ੇਸ਼ ਚਿਨ੍ਹ ਦੀ ਨੁਮਾਇੰਦਗੀ ਕਰਦਾ ਹੈ।
5. ਕੰਮਪਾਈਲੇਸ਼ਨ (Compilation) ਸਮੇਂ ਨਿਰਦੇਸ਼।

ਪ੍ਰਸ਼ਨ 5: ਨਿਮਨਲਿਖਤ ਪ੍ਰਸ਼ਨਾਂ ਦੇ ਉੱਤਰ ਦਿਓ।

1. C ਭਾਸ਼ਾ ਦੀਆਂ ਕਿਹੜੀਆਂ ਸੀਮਾਵਾਂ ਹਨ ?
2. C ਭਾਸ਼ਾ ਦੇ ਪ੍ਰਯੋਗ ਖੇਤਰਾਂ ਦਾ ਵਰਣਨ ਕਰੋ ?
3. ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਨਿਰਦੇਸ਼ (directives) ਜਾਂ ਸਟੇਟਮੈਂਟ (statement) ਕੀ ਹੁੰਦੇ ਹਨ ?
4. ਗਲੋਬਲ ਡਿਕਲੇਰੇਸ਼ਨ (global declaration) ਤੋਂ ਤੁਸੀ ਕੀ ਸਮਝਦੇ ਹੋ ?
5. C ਪ੍ਰੋਗਰਾਮ ਦੀ ਆਮ ਰਚਨਾ ਦੱਸੋ ?
6. C ਪ੍ਰੋਗਰਾਮ ਦੇ ਲਾਗੂ ਕਰਨ ਵਿਚ ਕਿਹੜੇ ਬੁਨਿਆਦੀ ਕਦਮ ਹੁੰਦੇ ਹਨ ?
7. ਕੰਪਾਈਲੇਸ਼ਨ (Compilation) ਅਤੇ ਮੇਲ ਦੀਆਂ ਕਿਹੜੀਆਂ ਅਵਸਥਾਵਾਂ ਹੁੰਦੀਆਂ ਹਨ ?
8. printf( ) ਅਤੇ scanf( ) ਵਿਚ ਕੀ ਅੰਤਰ ਹੁੰਦਾ ਹੈ ?
9. printf( ) ਅਤੇ scanf( ) ਨੂੰ ਇਨਪੁਟ/ਆਊਟਪੁਟ ਫੰਕਸ਼ਨ ਕਿਉਂ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ?
10. & ਦਾ scanf( ) ਵਿਚ ਕੀ ਕੰਮ ਹੁੰਦਾ ਹੈ ?

Answer to objective questions

Q	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
Q1	c	c	b	e	b	d	c	c	b	c
Q3	True	False	True	True	False					
Q4	Middle	semicolon	'\0'	tab	#define					

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ **3**

ਕਾਂਸਟੈਂਟਸ, ਵੈਰੀਏਬਲਜ਼ ਅਤੇ ਡਾਟਾ ਟਾਈਪਸ
Constants, Variables & Data types

3.0.0 ਭੂਮਿਕਾ (Introduction)

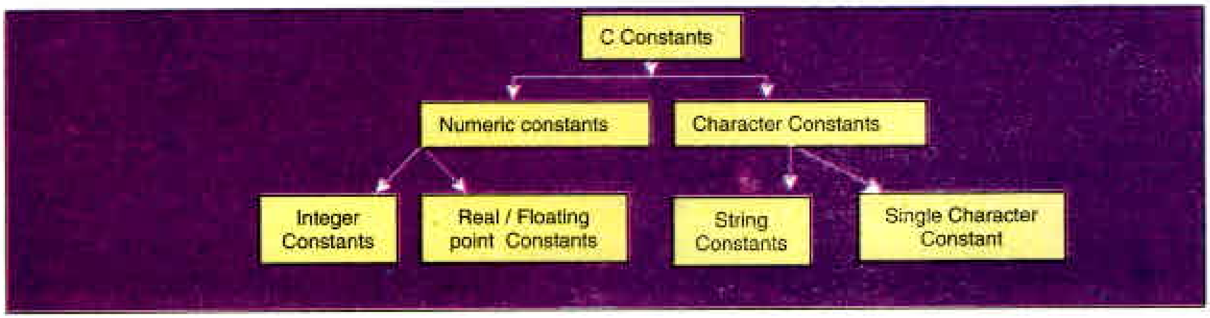
ਇਸ ਅਧਿਆਇ ਵਿਚ ਅਸੀ ਕਾਂਸਟੈਂਟ, ਵੈਰੀਏਬਲ ਅਤੇ ਡਾਟਾ ਟਾਈਪਸ ਬਾਰੇ ਸਿਖਾਂਗੇ ਜਿਸ ਵਿਚ ਵੈਰੀਏਬਲ ਟਾਈਪਸ ਜਿਵੇਂ ਕਿ short int, int, long int, double, long double, char ਆਦਿ ਦੀ ਮਦਦ ਨਾਲ ਵੈਰੀਏਬਲ ਦੇ ਨਾਮ ਅਤੇ ਕਾਂਸਟੈਂਟ ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਨ ਦੇ ਨਿਯਮਾਂ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਾਂਗੇ ।

3.1.0 ਕਾਂਸਟੈਂਟਸ/ਸ਼ਾਬਦਿਕ (Constants / literals)

ਸੀ (C) ਵਿਚ ਇਕ ਕਾਂਸਟੈਂਟ (constant) ਦੇ ਮੁੱਲ ਕਦੇ ਨਹੀਂ ਬਦਲਦੇ।ਅੱਖਰ (alphabets), ਅੰਕ (numbers) ਅਤੇ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਜਦੋਂ ਇਹ ਐਕਸਪ੍ਰੇਸ਼ਨ ਵਿਚ ਸਹੀ ਰੂਪ ਵਿਚ ਮਿਲਦੇ ਹਨ ਤਾਂ ਕਾਂਸਟੈਂਟਸ (constants), ਵੈਰੀਏਬਲਜ਼ (variables) ਅਤੇ ਕੀ-ਵਰਡਸ ਦੇ ਰੂਪ ਵਿਚ ਬਦਲ ਜਾਂਦੇ ਹਨ। ਇਹ ਮਾਤਰਾ (quantity) ਕੰਪਿਊਟਰ ਦੀ ਮੈਮਰੀ ਵਿਚ ਸਥਿਤ ਜਗ੍ਹਾ ਤੇ ਸਟੋਰ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਵੈਰੀਏਬਲ ਨੂੰ ਇਕ ਅਜਿਹਾ ਨਾਮ ਸਮਝਣਾ ਚਾਹੀਦਾ ਹੈ ਜਿਹੜਾ ਮੈਮਰੀ ਵਿਚ ਉਸ ਸਥਿਤੀ ਨੂੰ ਦਿਤਾ ਜਾਏ ਜਿਥੇ ਕਾਂਸਟੈਂਟ ਸਟੋਰ ਕੀਤਾ ਗਿਆ ਹੋਵੇ।

ਮਿਸਾਲ ਲਈ y=3.6*x+8.3; ਇਥੇ 3.6 ਅਤੇ 8.3 ਬਦਲੇ ਨਹੀਂ ਜਾ ਸਕਦੇ। ਉਨ੍ਹਾਂ ਨੂੰ ਕਾਂਸਟੈਂਟ (constant) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜਦਕਿ ਮਾਤਰਾਵਾਂ x & y ਬਦਲ ਸਕਦੀਆਂ ਹਨ ਅਤੇ ਉਨ੍ਹਾਂ ਨੂੰ ਵੈਰੀਏਬਲਜ਼ (variables) ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

3.1.1 ਸੀ (C) ਕਾਂਸਟੈਂਟ ਦੀਆਂ ਟਾਈਪਸ (The types of C Constants): ਸੀ (C) ਕਾਂਸਟੈਂਟ (constant) ਨਿਮਨਲਿਖਿਤ ਦੇ ਸ਼੍ਰੇਣੀਆਂ ਵਿਚ ਵੰਡਿਆ ਗਿਆ ਹੈ:



1) ਨਿਊਮੈਰਿਕ ਕਾਂਸਟੈਂਟ (Numeric Constant)

ਇਹ ਨਿਊਮੈਰਿਕ ਅੰਕਾਂ (Numeric digits) ਦੇ ਸਾਰ ਨਾਲ ਬਣੀ ਹੁੰਦੀ ਹੈ ਜਿਸ ਵਿਚ ਦਸ਼ਮਲਵ ਬਿੰਦੂ (.) ਦਾ ਵਿਕਲਪ ਮੌਜੂਦ ਹੁੰਦਾ ਹੈ।

- **ਪੂਰਨ-ਅੰਕ ਨਿਊਮੈਰਿਕ ਕਾਂਸਟੈਂਟ (integer numeric constant):** ਇਹ ਦਸ਼ਮਲਵ ਬਿੰਦੂ (.) ਦੇ ਬਗੈਰ ਨਿਊਮੈਰਿਕ ਅੰਕਾਂ ਦਾ ਸਾਰ ਹੁੰਦਾ ਹੈ

  ਆਮ ਤੌਰ ਤੇ ਪੂਰਨ-ਅੰਕ ਕਾਂਸਟੈਂਟ (Integer constant) ਦਾ ਰੂਪ ਹੁੰਦਾ ਹੈ :

ਚਿੰਨ੍ਹ (Sign)	ਅੰਕ (digit)

  ਜਿਥੇ ਚਿੰਨ੍ਹ **(Sign) :** ਵਿਕਲਪੀ ਹੈ "+" ਪੋਜਿਟਿਵ ਅੰਕ ਲਈ, "-" ਨੈਗਿਟਿਵ ਅੰਕ ਲਈ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅੰਕ (digit) : ਅੰਕਾਂ ਦਾ ਕ੍ਰਮ (a sequence of digits)

ਮਿਸਾਲ : +23, - 3421, 9999 etc.

- ਰੀਅਲ/ਫਲੋਟਿੰਗ ਪੁਆਇੰਟ ਕਾਂਸਟੈਂਟ (Real/Floating point Constant): ਨਿਊਮੈਰਿਕ ਅੰਕਾਂ (numeric digits) ਦੇ ਨਾਲ ਨਿਊਮੈਰਿਕ ਅੰਕਾਂ (numeric digits) ਦਾ ਕ੍ਰਮ

ਮਿਸਾਲ : 19 ਪੂਰਨ-ਅੰਕ ਕਾਂਸਟੈਂਟ (integer constant) ਹੈ
14.78 ਫਲੋਟ ਕਾਂਸਟੈਂਟ (float constant) ਹੈ।

2) ਚਿੰਨ੍ਹ ਕਾਂਸਟੈਂਟ (Character Constants)

ਚਿੰਨ੍ਹ ਕਾਂਸਟੈਂਟ ਪੂਰਨ-ਅੰਕ ਮੁੱਲ ਹੁੰਦੇ ਹਨ ਅਤੇ ਇਕਲੇ ਕੋਟਸ (single quotes) ਦੇ ਅੰਦਰ ਦਿੱਤੇ ਜਾਂਦੇ ਹਨ। ਚਿੰਨ੍ਹ ਕਾਂਸਟੈਂਟ ਦੇ ਮੁੱਲ ASCII character ਸੈੱਟ ਵਿਚ ਮਿਲਦੇ ਹਨ।

ਮਿਸਾਲ ਲਈ

Character Constant	ASCII Value
'A'	65
'B'	66
'C'	67

(ਹੋਰ Character Constants ਦੇ ਮੁੱਲਾਂ ਲਈ ASCII ਟੇਬਲ ਨੂੰ refer ਕਰੋ)

ਹੇਠ ਲਿਖੇ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਪੂਰਨ-ਅੰਕ ਅਤੇ ਚਿੰਨ੍ਹ ਡਾਟਾ ਪਰਿਵਰਤਨ (conversion) ਦਰਸਾਇਆ ਗਿਆ ਹੈ

```
#include<stdio.h>
main()
{
  char c;int i;
   i=1;
   c=i;           /*this is allowed*/
/*c holds the character whose ASCII value is 1*/
   c='A';
    i=c;
/*i holds the ASCII equivalent of A which is 65*/
    printf("Value of c is % c and i is % d \n",c,i);
}
```

Output
Value of c is A and i is 65

3) ਸਟਰਿੰਗ ਕਾਂਸਟੈਂਟ (String Constants)

ਇਹ ਉਹ ਕਾਂਸਟੈਂਟ ਹੁੰਦੇ ਹਨ ਜੋ ਕਿ ਇਕ ਤਰਤੀਬ ਵਿਚ ASCII character ਹੁੰਦੇ ਹਨ ਜੋ ਕਿ ਡਬਲ ਕੋਟ (double quotes) ਦੇ ਵਿਚ ਦਿੱਤੇ ਜਾਂਦੇ ਹਨ। ਇਹਨਾ ਨੂੰ ਚਿੰਨ੍ਹ-ਕਾਂਸਟੈਂਟ (character constant) ਦਾ ਐਰੇ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜਿਸ ਦਾ ਅੰਤਿਮ ਚਿੰਨ੍ਹ \0 ਹੈ ਅਤੇ ਸੀ (C) ਕੰਪਾਈਲਰ (compiler) ਵਲੋਂ ਆਪਣੇ ਆਪ ਸਟਰਿੰਗ (string) ਦੇ ਅੰਤ ਤੇ ਰਖਿਆ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ ਲਈ : "is a Null string" , "this is C" , "x+2" , "2006"

3.2.0 ਸੀ (C) ਵੇਰੀਏਬਲਜ਼/ਆਈਡੈਂਟੀਫਾਈਰ ਦੀਆਂ ਟਾਈਪਸ (Types of C Variables)

ਵੇਰੀਏਬਲ (variable)

ਉਹ ਮਾਤਰਾ (quantity) ਜੋ ਪ੍ਰੋਗਰਾਮ ਦੇ ਲਾਗੂਕਰਨ (execution) ਦੌਰਾਨ ਆਪਣੇ ਮੁੱਲਾ ਨੂੰ ਬਦਲ ਲੈਂਦੀ ਹੈ ਨੂੰ ਵੇਰੀਏਬਲ ਆਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਇਕ ਵਿਸ਼ੇਸ਼ ਮੈਮਰੀ ਸਥਿਤੀ ਦੀ ਪ੍ਰਤਿਨਿਧਤਾ (representation) ਕਰਦੇ ਹਨ ਜਿਥੇ ਡਾਟਾ ਸਟੋਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

ਮਿਸਾਲ ਲਈ : ਖੇਤਰ (area), ਨਾਮ (name), ਸ਼ਹਿਰ (city), ਉਮਰ (age) ਆਦਿ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Invalid variables	Correction	Valid variables
RUN;	Include a punctuation character	marks
1st term	Start with a number	Total_marks
-	Underscore alone	Gross_salary_2007
Net-pay	Includes a punctuation character	area_of_circle()
Net pay	Include space	num[20]
Switch	C Keyword	
\n	Starts with punctuation character	
ਵੈਰੀਏਬਲਜ਼ (variables) ਦੀਆਂ ਹੋਰ ਟਾਈਪਸ ਹਨ ਜਿਹੜੀਆਂ ਸੀ (C) ਦੀਆਂ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਸਟੇਟਮੈਂਟਸ ਵਿਚ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ।		int si, m_hra; float basic_sal char code;

3.2.1 ਡਿਲੀਮਿਟਰ (Delimiters)

ਡਿਲੀਮਿਟਰ ਇਕ ਚਿੰਨ੍ਹ (symbol) ਹੁੰਦਾ ਹੈ ਜੋ ਕਿ ਪ੍ਰੋਗਰਾਮ ਦੇ ਬੁਨਿਆਦੀ ਐਲੀਮੈਂਟਸ ਦੀ ਸੀਮਾ (limit) ਨਿਸ਼ਚਿਤ ਕਰਦੇ ਹਨ, ਇਸ ਲਈ ਇਹਨਾ ਨੂੰ **ਡਿਲਿਮਿਟਰ ਜਾਂ ਸੈਪਰੇਟਰਸ** ਵੀ ਕਹਿੰਦੇ ਹਨ। ਇਹ ਕਾਂਸਟੈਂਟ, ਵੈਰੀਏਬਲਜ਼ ਅਤੇ ਸਟੇਟਮੈਂਟਸ ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਦੇ ਵਿਚ ਵੱਖ ਵੱਖ ਕਰਦੇ ਹਨ ਪਰੰਤੂ ਮੁੱਲ ਦੀ ਪ੍ਰਾਪਤੀ ਲਈ ਕਿਸੇ ਓਪਰੇਸ਼ਨ (operation) ਨੂੰ ਨਿਸ਼ਚਿਤ ਨਹੀਂ ਕਰਦੇ।

C Language Sentences, ਲੇਬਲ (Labels), ਐਰੇਜ਼ (Arrays) ਆਦਿ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹਾਂ ਦੁਆਰਾ ਅਲਗ ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਇਨ੍ਹਾਂ ਨੂੰ ਡਿਲਿਮਿਟਰ (Delimiters) ਕਹਿੰਦੇ ਹਨ।

ਡਿਲੀਮਿਟਰ ਕੁਝ ਮਿਸਾਲਾਂ ਨਿਮਨਲਿਖਤ ਹਨ :-

Hash	#	Pre- Processor directive
Comma	,	Variable delimiter in variable list
Curly braces	{ }	used to block C sentences
Square brackets	[ ]	used with arrays
Parenthesis	( )	used in expressions
Colon	:	Label delimiter

3.2.2 ਵੈਰੀਏਬਲਜ਼ ਦਾ ਡਿਕਲੇਰੇਸ਼ਨ / ਇਨਿਸ਼ਿਯਲਾਇਜ਼ੇਸ਼ਨ (Declaration/Initialization of Variables)

ਸਮੂਹ ਵੈਰੀਏਬਲਜ਼ (Variables) ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਵਰਤਣ ਤੋਂ ਪਹਿਲਾਂ ਡਿਕਲੇਅਰ (declare) ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਡਿਕਲੇਅਰ (declare) ਕਰਨ ਦਾ ਮੂਲ ਉਦੇਸ਼ ਇਨ੍ਹਾਂ ਵੈਰੀਏਬਲਜ਼ (Variables) ਲਈ ਲੋੜੀਂਦੀ ਮੈਮੋਰੀ ਦੀ ਮਾਤਰਾ (amount) ਨੂੰ ਨਿਸ਼ਚਿਤ ਕਰਨਾ ਹੁੰਦਾ ਹੈ ਤਾਂ ਜੋ ਵੈਰੀਏਬਲਜ਼ ਦੇ ਮੁੱਲ ਜਾਂ ਮੁੱਲਾਂ ਨੂੰ ਸਟੋਰ ਕੀਤਾ ਜਾ ਸਕੇ।

ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੇ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਹਿੱਸੇ ਵਿਚ ਕੀਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ ਅਤੇ ਇਹ ਸੀ (C) ਦਾ ਸਿਧਾਂਤ ਹੈ ਅਤੇ ਇਸ ਦਾ ਵਾਕ (syntax) ਹੈ:-

Data type variable list ;

ਡਾਟਾ ਟਾਈਪ (Data type)

ਇਹ ਬੁਨਿਆਦੀ ਡਾਟਾ ਟਾਈਪਸ ਹਨ ਜਿਵੇਂ ਫਲੋਟ (float), ਕੈਰ (character), ਇੰਟ (int) ਜਾਂ ਡਬਲ (double)

ਵੈਰੀਏਬਲ ਲਿਸਟ (Variable list)

ਇਹ ਡਾਟਾ_ਟਾਈਪ (data_type) ਕਿਸਮ ਦੇ ਇਕ ਜਾਂ ਇਕ ਤੋਂ ਜਿਆਦਾ ਵੈਰੀਏਬਲਜ਼ (variables) ਹਨ ਅਤੇ ਇਹ ਵੈਰੀਏਬਲਜ਼ (variables) ਅਰਧ ਵਿਰਾਮ (comma) ਦੁਆਰਾ ਵੱਖ ਹੋਣੀਆਂ ਚਾਹੀਦੀਆਂ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੈਮੀਕੋਲਨ (Semicolon) ;

ਇਕ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਦਾ ਡੇਲੀਮਿਟਰ (delimiter) ਹੈ।

ਅਸਾਈਨਿੰਗ ਮੁੱਲ ਵੇਰੀਏਬਲਜ਼ ਨੂੰ (Assigning value to Variables)

ਵੇਰੀਏਬਲਜ਼ (variables) ਨੂੰ ਮੁੱਲ ਦੇਣ ਵਾਲੀ ਪ੍ਰਕਿਰਿਆ ਨੂੰ ਮੁੱਲਾਂ ਦੀ ਅਸਾਈਨਮੈਂਟ (assignment) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਹ ਅਸਾਈਨਮੈਂਟ-ਓਪਰੇਟਰ (assignment operator) "=" ਦੁਆਰਾ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਇਸ ਦਾ ਵਾਕ (syntax) ਹੈ:- **Variable_name = value;**

ਵੇਰੀਏਬਲ_ਨਾਮ (Variable_name)

ਇਹ ਮੇਮਰੀ ਸਥਿਤੀ ਦੀ ਪ੍ਰਤਿਨਿਧਤਾ (representation) ਕਰਦੇ ਹਨ ਜਿਥੇ "ਮੁੱਲ" ਸਟੋਰ ਕੀਤਾ ਗਿਆ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।

ਮਿਸਾਲ :

```
int i = 0;                              int i;
                 } are equivalent to {  char ch;
char ch = 'a';                          i = 0;
                                        ch = 'a';
```

ਇਸ ਨੂੰ ਵੇਰੀਏਬਲਜ਼ (variables) ਦੀ ਇਨੀਸ਼ਿਯਲਾਈਜੇਸ਼ਨ (initialization) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਸੀ (C) ਹਮੇਸ਼ਾ ਪ੍ਰੋਗਰਾਮਰ ਨੂੰ ਇਸ ਤਰੀਕੇ ਨਾਲ ਡਿਕਲੇਰੇਸ਼ਨ/ਇਨੀਸ਼ਿਯਲਾਈਜ਼ਰ ਲਿਖਣ ਲਈ ਆਗਿਆ ਦਿੰਦਾ ਹੈ ਪਰੰਤੂ ਹਮੇਸ਼ਾ ਅਜਿਹਾ ਕਰਨਾ ਕੋਈ ਜਰੂਰੀ ਨਹੀ ਹੁੰਦਾ।

ਮਿਸਾਲ :

int x =1; float total = 0.0 ; char ch = 'Y' ; double r = 0.123e-3 ;	x = 10 ; total = 3000.00 ; name = "Ram" ; ch = ' y'
Assignment within an executable part does not include data type.	

3.2.3 ਵੇਰੀਏਬਲ ਵਿਚ ਕਾਂਸਟੈਂਟ ਸਟੋਰ ਕਰਨਾ (Storing constant in a variable)

ਇਹ ਕਾਂਸਟੈਂਟ-ਮੋਡੀਫਾਇਰ (const Modifier) ਦੀ ਵਰਤੋਂ ਦੁਆਰਾ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਜਿਹੜਾ ਡਿਕਲੇਰੇਸ਼ਨ-ਸਟੇਟਮੈਂਟ (declaration statement) ਵਿਚ ਨਿਮਨਲਿਖਤ ਅਨੁਸਾਰ ਵੇਰੀਏਬਲ (variable) ਨਾਮ ਤੋਂ ਪਹਿਲਾਂ ਹੁੰਦਾ ਹੈ।

const type variablename = value;

[ਜਿਥੇ ਟਾਈਪ (type) ਇਕ ਟਾਈਪ ਦਾ ਸਧਾਰਣ ਡਾਟਾ ਟਾਈਪ ਹੁੰਦਾ ਹੈ, ਮੁੱਲ ਕਾਂਸਟੈਂਟ (constant) ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਹ ਵੇਰੀਏਬਲ-ਨਾਮ (variable name) ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਜਾਵੇਗਾ]

ਮਿਸਾਲ : **const float pie =3.1428;**

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

3.3.0 ਡਾਟਾ ਟਾਈਪਸ (Data Types)

ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਵਿਚ ਡਾਟਾ ਟਾਈਪਸ ਪਰਿਭਾਸ਼ਤ ਮੁੱਲਾਂ ਦਾ ਇਕ ਸਮੂਹ ਹੁੰਦਾ ਹੈ। ਪੂਰਣ-ਅੰਕ, ਫਲੋਟ, ਕਰੈਕਟਰ, ਸਟਰਿੰਗ, ਪੁਆਂਇਟਰ ਆਦਿ ਡਾਟਾ ਟਾਈਪਸ ਦੀਆਂ ਮਿਸਾਲਾਂ ਹਨ। ਕੁਝ ਨਿਰਧਾਰਿਤ ਡਾਟਾ ਟਾਈਪਸ ਹੀ ਸੀ ਭਾਸ਼ਾ ਵਲੋਂ ਪ੍ਰਦਾਨ ਕੀਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਆਮ ਤੋਰ ਤੇ ਡਾਟਾ ਟਾਈਪ ਦੀ ਸੀਮਾ (limit) ਸੀ ਭਾਸ਼ਾ ਦੁਆਰਾ ਨਿਰਧਾਰਿਤ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਇਹ ਡਾਟਾ ਦੀ ਟਾਈਪ ਦਾ ਸੰਕੇਤ ਦਿੰਦੇ ਹਨ ਜਿਹੜੇ ਵੇਰੀਏਬਲਜ਼ (variables) ਦੁਆਰਾ ਗ੍ਰਹਿਣ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਡਾਟਾ, ਅੰਕ ਵਿਚ ਜਾਂ ਗੈਰ-ਅੰਕ ਵਿਚ ਹੋ ਸਕਦਾ ਹੈ। ਸੀ (C) ਵਿਚ ਡਾਟਾ ਟਾਈਪਸ ਨਿਮਨਲਿਖਤ ਹਨ :-

1. ਬਿਲਟ ਇਨ ਡਾਟਾ ਟਾਈਪਸ — **Built in data types**
2. ਯੂਜ਼ਰ-ਡਿਫਾਇੰਡ ਡਾਟਾ ਟਾਈਪਸ — **User defined data types**

3.3.1 ਬਿਲਟ ਇਨ ਡਾਟਾ ਟਾਈਪਸ (Built in data types)

ਇਹ ਸੀ (C) ਦੀਆਂ ਬੁਨਿਆਦੀ ਡਾਟਾ ਟਾਈਪਸ (data types) ਹੁੰਦੀਆਂ ਹਨ ਅਤੇ ਇਕ ਮੁੱਲ (single value) ਨੂੰ ਗ੍ਰਹਿਣ ਕਰਦੀਆਂ ਹਨ ਜੋ ਕਿ ਹੇਠਾਂ ਦਿਖਾਈਆਂ (displayed) ਗਈਆਂ ਹਨ ।

Types	Keywords	size
Integer	int	2
Character	char	1

Types	Keywords	size
real (floating-point)	float	4
Double precision	double	8
non-specific	void	---

ਗੈਰ ਵਿਸ਼ੇਸ਼ (non specific) ਡਾਟਾ-ਟਾਈਪ (data type) ਕੇਵਲ ਇਕੋ ਹੀ ਹੁੰਦਾ ਹੈ ਜਿਸਨੂੰ ਸੀ (C) ਵਿਚ ਵੋਆਇਡ (void) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜੋ ਕਿਸੇ ਚੀਜ ਨੂੰ ਨਿਸ਼ਚਿਤ ਨਹੀਂ ਕਰਦਾ ।

1) **Integers (int):** Any integer number is a sequence of digits without a decimal point

integer type	bytes	Range	Valid integer	invalid integer
short int or int	2	-32768 to +32767	0X23a , 0300	.123 (decimal not allowed)
Long int	4	- 2147483648 to + 2147483647	-3578 , -0210 123 , 0	45324 (not permissible range) 3,234 comma not allowed

2) **ਫਲੋਟਿੰਗ-ਪੁਆਂਇੰਟ-ਡਾਟਾ ਟਾਈਪਜ਼ (floating point data types)**

ਫਲੋਟਿੰਗ-ਪੁਆਂਇੰਟ-ਡਾਟਾ ਟਾਈਪਸ ਨੰਬਰਸ ਅੰਸ਼-ਭਾਗ ਸਮੇਤ (numbers with fractional parts) ਨੂੰ ਸਟੋਰ ਕਰਨ ਅਤੇ ਬਹੁਤ ਵੱਡੇ ਆਕਾਰ (range) ਦੇ ਖੇਤਰ (sizes) ਦੇ ਕੁਸ਼ਲ-ਪ੍ਰਬੰਦਨ ਲਈ ਸਾਧਨ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ

Real/floating-point	bytes	Range	Valid float	invalid float
Float	4	3.4×10^{-38} to $3.4\times10^{+38}$	-123.543, 1.E-2	1,5(decimal missing, comma not allowed)
double	8	1.7×10^{-308} to $1.7\times10^{+308}$	.0001E+2, 1.5E+10	-12.65E+40 (Exponent too large)
Long double	10	3.4×10^{-4932} to $3.4\times10^{+4932}$	1234. , 10.00E0	+132.80E no digit for exponent

*E ਜਾਂ e ਇਕ ਪ੍ਰਤਿਨਿਧ (representing) ਭਾਗ ਹੁੰਦਾ ਹੈ, ਜਿਹੜਾ ਇਸ ਗੱਲ ਦਾ ਸੰਕੇਤ ਦਿੰਦਾ ਹੈ ਕਿ ਮੁੱਲ ਜੋ ਪਹਿਲਾਂ ਵਰਤੀਆ ਗਿਆ ਹੈ ਉਹ ਪ੍ਰਤਿਨਿਧ (representing) ਭਾਗ ਦੇ ਪੂਰਣ-ਅੰਕ (integer) ਦੇ ਮੁੱਲ (value) ਨੂੰ ਦਸ ਨਾਲ ਗੁਣਾ ਕਰਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

3) ਕੈਰੇਕਟਰ ਡਾਟਾ ਟਾਈਪ (ਕੈਰ) [character data type (char)]

char ਇਕ ਕੀ-ਵਰਡ ਹੈ ਜੋ ਕਿ ਕੈਰੇਕਟਰ ਟਾਈਪ ਡਾਟਾ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇਹ ਡਾਟਾ ਕੈਰ-ਕਾਂਸਟੈਂਟ (character constant) ਜਾਂ ਸਟਰਿੰਗ-ਕਾਂਸਟੈਂਟ (string constant) ਹੋ ਸਕਦਾ ਹੈ। ਕੈਰ-ਕਾਂਸਟੈਂਟ (character constant) ਉਹ ਹੁੰਦਾ ਹੈ ਜਿਸ ਵਿਚ ਕੋਈ ਵੀ ਇੱਕਲਾ ਕੈਰੇਕਟਰ ਲੋਪ-ਚਿੰਨ੍ਹ (" ") ਦੇ ਵਿਚਕਾਰ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਟੋਰੇਜ਼ ਲਈ 8 ਬਿਟ ਲੈਂਦਾ ਹੈ। ਕੈਰ ਕਾਂਸਟੈਂਟ ਸਾਇੰਡ (signed) ਜਾਂ ਅਨਸਾਇੰਡ (unsigned) ਹੋ ਸਕਦੇ ਹਨ। ਸਾਇੰਡ (signed) character ਮੁੱਲ -128 ਤੋਂ +127 ਤਕ ਅਤੇ ਅਨਸਾਇੰਡ (unsigned) character ਵਾਲੇ ਕੈਰ 0 ਤੋਂ 255 ਤਕ ਹੁੰਦੇ ਹਨ।

char ਡਾਟਾ ਟਾਈਪ ਦੀ ਮਿਸਾਲ char ch = "y";

4) ਡਬਲ (Double)

ਇਹ ਇਕ ਕੀ-ਵਰਡ ਹੈ ਜੋ ਡਬਲ-ਸੁਨਿਸ਼ਚਤਤਾ (double precision) ਫਲੋਟਿੰਗ-ਪੁਆਇੰਟ ਨੰਬਰ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਥੇ ਸੁਨਿਸ਼ਚਤਤਾ (precision) ਦਾ ਸੰਬੰਧ ਡਾਟਾ ਦੇ ਸਹੀਪਨ (accuracy) ਦੇ ਨਾਲ ਹੈ ਅਤੇ ਉਦੋਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਜ਼ਿਆਦਾ ਸ਼ੁਧਤਾ (accuracy) ਦੀ ਲੋੜ ਹੋਵੇ। ਆਮ ਤੌਰ ਤੇ ਇਹ (ਡਬਲ) ਫਲੋਟ ਦੇ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ ਪਰੰਤੂ ਮਹੱਤਵਪੂਰਨ-ਅੰਕਾਂ (significant digits) ਦੀ ਸੰਖਿਆ ਦਸ਼ਮਲਵ ਬਿੰਦੂ ਦੇ ਬਾਅਦ ਸਟੋਰ ਕੀਤੀ ਗਈ ਹੁੰਦੀ ਹੈ ਜੋ ਕਿ ਫਲੋਟ (float) ਤੋਂ ਦੋਗੁਣਾ ਹੁੰਦੇ ਹਨ।

ਫਲੋਟ (float) ਆਮ ਤੌਰ ਤੇ ਦਸ਼ਮਲਵ ਬਿੰਦੂ ਦੇ ਬਾਅਦ ਵੱਧ ਤੋਂ ਵੱਧ 6 ਅੰਕਾਂ (digits) ਨੂੰ ਸਟੋਰ ਕਰਦੇ ਹਨ। ਪਰੰਤੂ ਡਬਲ (double) ਦਸ਼ਮਲਵ ਬਿੰਦੂ ਦੇ ਬਾਅਦ ਦੇ 16 ਮਹੱਤਵਪੂਰਨ ਅੰਕਾਂ (digits) ਨੂੰ ਸਟੋਰ ਕਰਦੇ ਹਨ।

ਮਿਸਾਲ : **234.0000000000000000** or **-0.0000001023999001**

5) ਵੋਆਇਡ ਡਾਟਾ ਟਾਈਪ (The void Data type)

ਸੀ (C) ਦੇ ਹਰ ਇਕ ਫੰਕਸ਼ਨ ਦੀ ਰਿਟਰਨ ਡਾਟਾ-ਟਾਈਪ ਹੁੰਦੀ ਹੈ। ਇਹ ਰਿਟਰਨ ਡਾਟਾ ਟਾਈਪ ਵੀ ਇਕ ਤਰ੍ਹਾਂ ਦੀ ਡਾਟਾ ਟਾਈਪ ਹੀ ਹੁੰਦੀ ਹੈ ਜੋ ਕਿ ਫੰਕਸ਼ਨ ਦੇ ਵਾਪਸ ਕੀਤੇ ਹੋਏ ਮੁੱਲ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਜੇਕਰ ਕੋਈ ਫੰਕਸ਼ਨ (function) ਕਿਸੇ ਮੁੱਲ ਨੂੰ ਵਾਪਿਸ ਨਹੀਂ ਕਰਦਾ ਤਾਂ ਇਸ ਦੀ ਵਾਪਸੀ ਟਾਈਪ ਵੋਆਇਡ (void) ਮੰਨੀ ਜਾਵੇਗੀ। ਜਿਵੇਂ ਕਿ main() ਇਕ ਫੰਕਸ਼ਨ (function) ਹੈ ਜਿਥੋਂ ਲਾਗੂਕਰਨ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। ਇਹ ਆਮ ਤੌਰ ਤੇ ਵੋਆਇਡ-ਟਾਈਪ (void type) ਨਾਲ ਡਿਕਲੇਅਰ (declare) ਹੁੰਦਾ ਹੈ। ਮੇਨ ਅੱਖਰ ਤੋਂ ਬਾਅਦ ਛੋਟੀ ਬ੍ਰੈਕਟ ਸ਼ੁਰੂ ਤੇ ਬੰਦ ਹੁੰਦੀ ਹੈ। ਇਹਨਾਂ ਬ੍ਰੈਕਟਾਂ ਦੇ ਵਿਚ ਜੇਕਰ ਕੁਝ ਹੋਵੇ ਤਾਂ ਉਹ ਫੰਕਸ਼ਨ ਦੀਆਂ ਰਸਮੀ (formal) ਆਰਗੂਮੈਂਟਸ (arguments) ਅਖਵਾਉਂਦੀਆਂ ਹਨ ਜੇਕਰ ਕੋਈ ਵੀ ਆਰਗੂਮੈਂਟਸ (arguments) ਨਾ ਹੋਣ ਤਾਂ ਅਸੀ ਲਿਖਦੇ ਹਾਂ ਵੋਆਇਡ (void)।

3.3.2 ਮੇਨ ਫੰਕਸ਼ਨ ਹੈਡਰ (main function header)

[ਵੋਆਇਡ ਮੇਨ(ਵੋਆਇਡ)] [void main(void)] ਨੂੰ main () ਵਜੋਂ ਵੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ।

3.4.0 ਟੋਕਨਜ਼ (ਆਈਡੈਂਟਿਫਾਇਰਜ਼, ਕੀ-ਵਰਡਜ਼, ਕਾਂਸਟੈਂਟ, ਓਪਰੇਟਰਸ) [Tokens (Identifiers, keywords, constants, operators)]

ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਦੀਆਂ ਬੁਨਿਆਦੀ ਅਤੇ ਸਭ ਤੋਂ ਛੋਟੀ ਇਕਾਈਆਂ (units) ਨੂੰ ਟੋਕਨ (token) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਹਰ ਇਕ ਟੋਕਨ (token), ਸੀ (C) ਦੇ ਇਕ ਜਾਂ ਇਕ ਤੋਂ ਵੱਧ ਚਿੰਨ੍ਹਾਂ ਦੇ ਬਣੇ ਹੁੰਦੇ ਹਨ। ਸੀ (C) ਵਿਚ 6 ਟਾਈਪਸ ਦੇ ਟੋਕਨ (token) ਹੁੰਦੇ ਹਨ।

ਉਹ ਹਨ (They are) :-

1. Keywords	2. Identifiers	3. Constants	4. Strings	5. Operators	6. Special Symbols

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

3.4.1 ਕੀ-ਵਰਡਜ਼ ਅਤੇ ਆਈਡੈਂਟੀਫਾਇਰਜ਼ (Keywords and Identifiers)

ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਹਰੇਕ ਸ਼ਬਦ ਜਾਂ ਤਾਂ ਇਕ ਕੀ ਵਰਡ ਹੁੰਦਾ ਹੈ ਜਾਂ ਆਈਡੈਂਟੀਫਾਈਰ (identifier) ਹੁੰਦਾ ਹੈ। ਸਾਰੇ ਕੀ-ਵਰਡ ਬੁਨਿਆਦੀ ਤੌਰ ਤੇ ਚਿੰਨ੍ਹਾਂ (characters) ਦੇ ਸਾਰ ਹੁੰਦੇ ਹਨ ਜਿਨ੍ਹਾਂ ਦਾ ਇਕ ਜਾਂ ਵਧੇਰੇ ਨਿਸ਼ਚਿਤ ਅਰਥ ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਹ ਅਰਥ ਕਿਸੇ ਵੀ ਹਾਲਾਤ ਵਿਚ ਬਦਲੇ ਨਹੀਂ ਜਾ ਸਕਦੇ। ਸਾਰੇ ਸੀ (C) ਕੀ-ਵਰਡ ਲੋਅਰ ਕੇਸ (lower case) ਅੱਖਰਾਂ ਵਿਚ ਲਿਖੇ ਜਾਣੇ ਚਾਹੀਦੇ ਹਨ ਕਿਉਂਕਿ ਸੀ (C) ਵਿਚ ਅਪਰ-ਕੇਸ (upper case) ਅਤੇ ਲੋਅਰ-ਕੇਸ (lower case) ਅੱਖਰ ਮਹੱਤਵਪੂਰਣ ਹੁੰਦੇ ਹਨ।

auto	break	case	char	continue	default
do	double	else	enum	extern	float
for	goto	if	int	long	register
return	short	sizeof	static	struct	switch
typedef	union	unsigned	void	while	

ਆਈਡੈਂਟੀਫਾਈਰਸ (Identifiers)

ਆਈਡੈਂਟੀਫਾਈਰ ਦੀ ਵਰਤੋਂ ਸੀ (C) ਵਿਚ ਵੇਰੀਏਬਲ (variable), ਕਾਂਸਟੈਂਟ (constant) ਅਤੇ ਫੰਕਸ਼ਨਸ (functions) ਦੇ ਨਾਮ ਬਿਆਨਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਬੁਨਿਆਦੀ ਤੌਰ ਤੇ ਅੱਖਰਾਂ (letters) ਅਤੇ ਅੰਕਾਂ (digits) ਦਾ ਸਾਰ ਹੁੰਦਾ ਹੈ। ਸੀ (C) ਵਿਚ ਵੇਰੀਏਬਲ (variable) ਇਕ ਅਜਿਹੀ ਮਾਤਰਾ ਹੁੰਦੀ ਹੈ ਜੋ ਪ੍ਰੋਗਰਾਮ ਲਾਗੂ ਕਰਨ ਸਮੇਂ ਬਦਲੀ ਜਾ ਸਕਦੀ ਹੈ।

ਆਈਡੈਂਟੀਫਾਈਰ (Identifiers) ਨਾਵਾਂ ਦੀ ਬਣਤਰ ਲਈ ਅਪਣਾਏ ਜਾਣ ਵਾਲੇ ਨਿਯਮ ਨਿਮਨਲਿਖਤ ਅਨੁਸਾਰ ਹਨ :

(ੳ) ਪਹਿਲਾ ਚਿੰਨ੍ਹ (character) ਇਕ ਅੱਖਰ (letter) ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।
(ਅ) ਅਗਲੇ ਸਾਰੇ (succeeding) ਚਿੰਨ੍ਹ (character), ਅੱਖਰਾਂ (letters) ਜਾਂ ਅੰਕਾਂ (digits) ਦੇ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ।
(ੲ) ਸੀ (C) ਵਿਚ ਅਪਰ-ਕੇਸ (upper case) ਅਤੇ ਲੋਅਰ-ਕੇਸ (lower case) ਆਈਡੈਂਟੀਫਾਈਰਸ (Identifiers) ਵੱਖ ਵੱਖ ਹੁੰਦੇ ਹਨ।
(ਸ) ਅੰਡਰ ਸਕੋਰ (i.e "_") ਤੋਂ ਇਲਾਵਾ ਵਿਰਾਮ ਚਿੰਨ੍ਹ ਜਾਂ ਕੋਈ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਨਹੀਂ ਹੋਣਾ ਚਾਹੀਦਾ।
(ਹ) ਕਦੇ ਵੀ ਲਗਾਤਾਰ ਦੋ ਅੰਡਰ ਸਕੋਰ ਚਿੰਨ੍ਹ (i.e "__") ਇਸਤੇਮਾਲ ਨਹੀਂ ਕੀਤੇ ਜਾ ਸਕਦੇ।
(ਕ) ਕੀ-ਵਰਡਸ ਆਈਡੈਂਟੀਫਾਈਰਸ (Identifiers) ਦੀ ਤਰ੍ਹਾਂ ਇਸਤੇਮਾਲ ਨਹੀਂ ਕੀਤੇ ਜਾ ਸਕਦੇ।

2.5.2 ਟਾਈਪ ਮੋਡੀਫਾਈਰ ਜਾਂ ਕੁਆਲੀਫਾਈਰ (Type modifier or Qualifier)

ਇਹ ਬੁਨਿਆਦੀ ਪੂਰਣ-ਅੰਕ (integer) ਨੂੰ ਸੁਧਾਰਣ ਲਈ ਤਿੰਨ ਜੁੜਵੇਂ ਕੀ-ਵਰਡ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ : ਛੋਟੇ, ਲੰਬੇ ਅਤੇ ਅਨਸਾਈਂਡ (unsigned)। ਇਥੇ ਦਿਮਾਗ ਵਿਚ ਰਖਣ ਯੋਗ ਕੁਝ ਨੁਕਤੇ ਹਨ:-

short int <= int <= long int
float <= double <= long double

ਅਨਸਾਈਨਡ int (Unsigned int)

ਇਹ ਯੋਗਤਾ ਉਨ੍ਹਾਂ ਵੇਰੀਏਬਲਜ਼ (variables) ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਜਿਹੜੇ ਕੇਵਲ ਪੋਜ਼ਿਟਿਵ ਮੁੱਲ ਹੀ ਲੈਂਦੇ ਹਨ। Int ਡਿਕਲੇਰੇਸ਼ਨ (declaration) (ਜਾਂ ਲੰਬਾ int ਜਾਂ ਛੋਟਾ int) ਸ਼ਬਦ ਅਨਸਾਈਂਡ (Unsigned) ਦੇ ਨਾਲ ਪਿਛੇਤਰ ਵਿਚ ਲਗਾ ਕੇ, ਪੋਜ਼ਿਟਿਵ ਅੰਕਾਂ ਦੀ ਰੇਂਜ (range) ਡਬਲ *(double)* ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਅਨਸਾਈਂਡ ਅੰਕ ਹਮੇਸ਼ਾ ਪੋਜ਼ਿਟਿਵ ਹੁੰਦੇ ਹਨ ਜਾਂ 0 ਅਤੇ 2^n ਦੀ ਆਗਿਆ ਮੰਨਦੇ ਹਨ, ਜਿਸ ਵਿਚ n ਬਿਟਸ ਦੇ ਨੰਬਰ ਹਨ। ਮਿਸਾਲ ਲਈ, ਜੇਕਰ ਚੈਰ (chars) 8 ਬਿਟਸ ਹੈ, ਤਾਂ ਅਨਸਾਈਂਡ-ਚੈਰ-ਵੇਰੀਏਬਲਜ਼ (Unsigned char variables) ਦਾ ਮੁੱਲ 0 ਅਤੇ 255 ਦੇ ਵਿਚ ਹੁੰਦਾ ਹੈ।

ਡਬਲ ਲੋਂਗ (Double long): ਇਹ ਹੋਰ ਜ਼ਿਆਦਾ ਵੱਡਾ ਫਲੋਟਿੰਗ ਨੰਬਰ ਦਰਸਾਉਂਦਾ ਹੈ।

o

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਡਾਟਾ ਟਾਈਪਸ, ਵੈਰੀਏਬਲਜ਼ ਅਤੇ ਕੌਨਸਟੈਂਟ (Data Types, Variables & Constants)

ਡਾਟਾ ਟਾਈਪ ਆਗਿਆਯੋਗ ਕੰਮਾਂ ਦੇ ਨਾਲ ਮੁੱਲਾਂ ਦਾ ਇਕ ਸੈੱਟ ਹੁੰਦਾ ਹੈ। ਸਾਧਾਰਣ ਡਾਟਾ ਟਾਈਪ ਵਿਚ ਪੂਰਣ ਅੰਕ, (ਸੰਖੇਪ, int ਜਾਂ long) ਅਸਲ ਅੰਕ (ਅਸਥਾਈ, ਦੁਗਣਾ) ਅਤੇ ਚਿੰਨ੍ਹ ਕੈਰ (char) ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਹਰ ਡਾਟਾ ਟੈਕਸਟ *(text)* ਜਿਹੜਾ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਵਰਤਿਆ ਜਾਣਾ ਹੁੰਦਾ ਹੈ ਨੂੰ ਡਿਕਲੇਅਰ ਅਤੇ ਆਰੰਭ ਕਰਨ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਬੈਕਸਲੈਸ਼ (\) ਚਿੰਨ੍ਹ ਨਿਰੰਤਰ ਇਕ ਵਿਸ਼ੇਸ਼ ਕੰਮ ਕਰਦੇ ਹਨ। ਇਹ ਬੈਕਸਲੈਸ਼ (\) ਦੇ ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦੇ ਹਨ। ਦੋਵੇਂ ਬੈਕਸਲੈਸ਼ ਅਤੇ ਸੰਬੰਧਿਤ ਚਿੰਨ੍ਹ ਇਕਹਰੇ ਕੋਟਾਂ (quote) ' ' ਦੇ ਨਾਲ ਨੱਥੀ ਹੁੰਦੇ ਹਨ।

ਡਾਟਾ ਕਿਸਮਾਂ ਚਾਰ ਟਾਈਪਸ ਦੀਆਂ ਹੁੰਦੀਆਂ ਹਨ ਜਿਵੇਂ signed, unsigned, short & long ਆਦਿ। ਇਹ ਚਿੰਨ੍ਹ ਟਾਈਪ ਡਾਟਾ ਅਤੇ ਪੂਰਣ-ਅੰਕ ਦੀ ਸ਼ੁਧਤਾ ਪ੍ਰਦਾਨ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।

ਚਿੰਨ੍ਹਾਤਮਕ ਕਾਨਸਟੈਂਟ ਸਮਰੂਪਾਂ ਦੇ ਮੁੱਲ ਹੁੰਦੇ ਹਨ ਜੋ ਪ੍ਰੀਪ੍ਰੋਸੈਸਰ ਹਿਦਾਇਤ # define ਦੇ ਨਾਲ ਪਰਿਭਾਸ਼ਤ ਹੁੰਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1: ਬਹੁ ਪਸੰਦੀ ਪ੍ਰਸ਼ਨ

1.1	ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕਿਹੜਾ ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਇਕ ਠੀਕ (ਸਹੀ) ਡਾਟਾ ਟਾਈਪ ਨਹੀਂ ਹੁੰਦਾ ? a) Charn b) float c) long d) double
1.2	ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕਿਹੜਾ ਅਰਥਮੈਟਿਕ ਓਪਰੇਟਰ ਨਹੀਂ ਹੈ ? a) + b) & c) % d) *
1.3	ਓਪਰੇਟਰ % ਨੂੰ ਲਾਗੂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ? a) Float Values b) Double values c) Integral Values d) All of these
1.4	ਨਿਮਨਲਿਖਤ ਵਿਚੋਂ ਕਿਹੜਾ ਟਾਈਪ int ਦਾ ਸਹੀ ਪੂਰਣ ਅੰਕ ਨਹੀਂ ਹੈ ? a) 3750 b) 32800 c) -32767 d) 0
1.5	ਸੀ (C) ਵਿਚ ਟਾਈਪ int ਦਾ ਪਰਿਵਰਤਨ ਰੇਂਜ ਵਿਚ ਮੁੱਲ ਲੈਂਦਾ ਹੈ ? a) 0 to 32767 b) 0 to 65535 c) -32768 to 32767 d) -32767 to 32768
1.6	ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕਿਹੜਾ ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਰਾਖਵਾਂ ਸ਼ਬਦ ਨਹੀਂ ਹੈ ? a) for b) goto c) doo d)switch
1.7	**5/6/3+8/3** ਦੇ ਐਕਸਪ੍ਰੇਸ਼ਨ **(expression)** ਦਾ ਕੀ ਮੁੱਲ ਹੋਵੇਗਾ ? a) 4 b) 4 c) 2.333(approx) d) None of these
1.8	ਸ਼ਨਾਖਤ ਕਰੋ ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕਿਹੜੇ ਸੀ (C) ਟੋਕਨ ਹੁੰਦੇ ਹਨ ? a) keywords b) constants c) operators d) All of these
1.9	ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕਿਹੜੇ ਸੀ (C) ਵਿਚ ਕੀ-ਵਰਡ ਨਹੀਂ ਹੁੰਦੇ ? a) const. b) main c) size of d) void
1.10	ਸੀ (C) ਵਿਚ ਦੋਹਰੇ ਅਰਥਮੈਟਿਕ ਔਪਰੇਟਰਾਂ ਦੀ ਗਿਣਤੀ ? a) 5 b) 4 c) 6 d) 7

ਪ੍ਰਸ਼ਨ 2: ਨਿਮਨਲਿਖਤ ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਵਰਤੋ।

2.1	ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕਿਹੜੇ ਗਲਤ ਵੈਰੀਏਬਲ **(variable)** ਨਾਮ ਹਨ ਤੇ ਕਿਉਂ ? a) roll-no or roll_no b) interest_paid c) SUM d) none of these

2.2	ਪ੍ਰੋਗਰਾਮ ਭਾਗ ਦਾ ਪਰਿਣਾਮ ਕੀ ਹੋਵੇਗਾ ?
	int x;
	x=11;
	x=12;
	x=13;
	printf("%d %d %d\n", x ,x ,x); }
	Ans 13 13 13

2.3	ਨਿਮਨਲਿਖਿਤ ਪ੍ਰੋਗਰਾਮ ਦੁਆਰਾ ਕੀ ਪ੍ਰਿੰਟ ਹੁੰਦਾ ਹੈ ?
	main ( )
	{ int a, b, c ;
	b=4;
	c=a+b;
	}
	Ans ਇਹ ਗਰਬੇਜ ਆਊਟਪੁਟ ਦੇਵੇਗਾ ਕਿਉਂਕੀ a ਨੂੰ ਕੋਈ ਕੀਮਤ ਨਹੀਂ ਦਿੱਤੀ ਗਈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

2.4	ਨਿਮਨਲਿਖਿਤ ਪ੍ਰੋਗਰਾਮਾਂ (2.4 to 2.10) ਦਾ ਪਰਿਣਾਮ ਕੀ ਹੋਵੇਗਾ ?
	void main (void)
	{
	printf("%d",'B');
	}
	Ans 66

2.5	ਨਿਮਨਲਿਖਿਤ ਦਾ ਕੀ ਪਰਿਣਾਮ ਹੋਵੇਗਾ ?
	void main (void)
	{
	float x=1/2.0-1/2;
	printf("%.2f",x);
	}
	Ans 0.50

2.6	ਜੇਕਰ x = 12.4568 ਅਤੇ printf ਫੰਕਸ਼ਨ (function) ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਕਿਹੜਾ ਮੁੱਲ ਛਪੇਗਾ ? printf("%.3f",x); **Ans 12.457**

2.7	ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਜੇਕਰ ਕੋਈ ਗਲਤੀ ਹੈ ਤਾਂ ਲੱਭੋ ?
	void main (void)
	{
	float a,b ;
	printf("\nEnter value of a: ");
	scanf("%f",a);
	b =a*3 ;
	printf("\nValue of b = %f\n" ,b) ;
	}

2.10	int x=3;n=4 ; x=++n; printf("%d",X) x=x++ ; printf("%d",X) **Ans 5 6**

ਪ੍ਰਸ਼ਨ 3: ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਗਈਆਂ ਸਮੱਸਿਆਵਾਂ ਦਾ ਹੱਲ ਲਭੋ।

1.	ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਸਹੀ ਕਰੋ। X =(float) 5/2		

2.	c ਦਾ ਮੁੱਲ ਕੀ ਹੋਵੇਗਾ?
	main( )
	int c ;
	float a ,b ;
	a=245.05;
	b=40.02;
	c=a+b;}
	
	Ans 285
		3.	a ਅਤੇ c ਦਾ ਮੁੱਲ ਲਭੋ ?
			main ( )
			{
			int a, b ,c ;
			b=2;
			a=2*(b++);
			c=2*(++b);
			}
			Ans a =4 ,c =8.

4.	a ਅਤੇ b ਦਾ ਮੁੱਲ ਲਭੋ। main ( ) int a,b ; a=2 ; b=++a+2; printf("Value of a is %d and b is % d\n" a, b);		**Ans a =3, b =5**

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰਸ਼ਨ 4: ਦੱਸੋ ਸਹੀ ਕੀ ਗਲਤ

1. ਵੈਰੀਏਬਲਜ਼ ਅਜਿਹੀ ਮਾਤਰਾ ਹੈ ਜੋ ਪ੍ਰੋਗਰਾਮ ਦੇ ਲਾਗੂਕਰਨ (execution) ਦੌਰਾਨ ਆਪਣੇ ਮੁੱਲਾਂ ਨੂੰ ਬਦਲ ਲੈਂਦੇ ਹਨ।
2. ਡਿਲਿਮਿਟਰ ਇਕ ਚਿੰਨ੍ਹ (character) ਹੁੰਦਾ ਹੈ ਜਿਸਦਾ ਅਰਥ ਵਾਕ ਰਚਨਾ ਅਤੇ ਮਹਤੱਤਾ ਹੁੰਦੀ ਹੈ।
3. ਇਕ ਕੈਰ (char) ਡਾਟਾ ਟਾਈਪ ਵਿਚ ਹਮੇਸ਼ਾ ਇਕ ਬਾਈਟ ਹੁੰਦਾ ਹੈ।
4. ਆਪਰੇਟਰ ਦਾ ਅਕਾਰ ਡਾਟਾ ਟਾਈਪ ਹੈ।
5. ਸੈਮੀਕੋਲੋਨ ਇਕ ਡਿਕਲੇਰੇਸ਼ਨ ਦਾ ਡੀਲਿਮਿਟਰ (delimiter) ਹੁੰਦਾ ਹੈ।

ਪ੍ਰਸ਼ਨ 5: ਨਿਮਨਲਿਖਿਤ ਲਈ ਉਚਿਤ ਸ਼ਬਦ ਜਾਂ ਪਰਿਣਾਮ ਲਿਖੋ ।

1. ਕੰਪਿਊਟਰ ਵਲੋ ਇਕ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਅਣਗਿਣਤ ਵਿਚ ਨਿਖੇੜਿਆ ਜਾਂਦਾ ਹੈ।
2. ਸੋਧਕ ਦੀ ਵਰਤੋਂ ਪਰਿਵਰਤਣ (variable) ਨੂੰ ਇਕ ਕਾਂਸਟੈਂਟ (constant) ਵਜੋਂ ਡਿਕਲੇਅਰ (declare) ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
3. ਜੇਕਰ a=-11 ਅਤੇ b=-3 ਤਾਂ a%b ਦਾ ਮੁੱਲ ਕੀ ਹੋਵੇਗਾ।
4. ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਸੰਧਿਰਕ ਓਪਰੇਟਰਾਂ ਦੀ ਗਿਣਤੀ ਕਿੰਨੀ ਹੁੰਦੀ ਹੈ।
5. ਜੇਕਰ ਸਾਡੇ ਕੋਲ *, /, (), % ਓਪਰੇਟਰ ਹੋਵੇ ਤਾਂ ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਸਭ ਤੋ ਉੱਚ ਮਹਤੱਤਾ ਵਾਲਾ ਕਿਹੜਾ ਹੈ?

ਪ੍ਰਸ਼ਨ 6: ਨਿਮਨਲਿਖਤ ਪ੍ਰਸ਼ਨਾਂ ਦੇ ਉੱਤਰ ਦਿਓ।

1. ਇਕ ਵੈਰੀਏਬਲਜ਼ (variable) ਵਿਚ ਕਾਂਸਟੈਂਟ (constant) ਸਟੋਰ ਕਰਨ ਲਈ ਕੀ ਵਿਧੀ ਹੈ? ਮਿਸਾਲ ਦਿਓ ?
2. ਸਟੇਟਮੈਂਟ (statement) big = a>b?a:b; ਕੀ ਸੰਕੇਤ ਦਿੰਦਾ ਹੈ ?
3. ਨਿਮਨਲਿਖਤ if: ਲਈ ਲੋਜਿਕਲ ਸਟੇਟਮੈਂਟ (logical statement) ਲਿਖੋ ?
4. (ੲ) Mark-obt ਸਭ ਵਿਦਿਆਰਥੀਆਂ ਲਈ 95 ਤੋਂ ਵੀ ਵੱਧ ਹਨ ਜਿਨ੍ਹਾਂ ਦੀ ਸ੍ਰੇਣੀ T ਨਹੀਂ ਹੈ ?
   (ਬੀ)Total marks ਸਭ ਵਿਦਿਆਰਥੀਆਂ ਲਈ 550 ਤੋਂ ਵੀ ਵਧ ਹਨ ਜਿਨ੍ਹਾਂ ਦੀ ਸ਼੍ਰੇਣੀ 3 A ਜਾਂ 3 B ਹੈ?
   ਮੰਨ ਲਓ ਕਿ y=y*3*(x+y*2) ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਹੈ ? ਇਸ ਨੂੰ ਇੰਝ ਮੁੜ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ ?
   y*=3;
   y*=x+ y*z
5. ਨਿਮਨਲਿਖਤ ਸਟੇਟਮੈਂਟਸ (statements) ਵਿਚ ਬਗੈਰ ਜਰੂਰਤੀ ਬਰੈਕਟਾਂ ਦੀ ਸ਼ਨਾਖਤ ਕਰੋ।
   a) ((x-(y/5)+z)%8)+25 b) ((z-k)*y)+a c) (a*b)+(-y/z)

ਜਾਂਚ ਕਰੋ ਕਿ ਹੇਠਾਂ ਦਿਤੇ ਗਏ ਪ੍ਰੋਗਰਾਮਸ ਵਿਚ ਕੀ ਛਪੇਗਾ

6
```
#include <stdio.h>
main( )
 { int x ,y;
   y=2 ;
   x=y+1;
   printf("%d %d\n" x, y);
   y=y+2;
   x= 5;
   printf("%d %d \n",x ,y)
 }
```

7)
```
#include <stdio.h>
main ( )
{  int a, b;
     a=6;
     a=a+3;
     b=4%a;
     printf("%d %d \n", a, b);
 }
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

8	#include <stdio.h>
	main( )
	{
	int a, c ;
	a=4;
	c=a*a*a;
	printf("%d%d%d\n",a,c,a*a);
	}
		9)	# include <stdio.h>
			main ( )
			{ int rate ,time, dist, junk ;
			rate=7,time=3;
			junk=rate+time;
			dist=rate*time;
			printf("%d %d %d %d \n",rate ,time,,junk,dist);
			}

ਪ੍ਰਸ਼ਨ 7: ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ

1. ਡੀਲਿਮਿਟਰ (delimiter) ਕੀ ਹੁੰਦੇ ਹਨ?
2. ਆਈਡੈਂਟੀਫਾਇਰ (identifier) ਕੀ ਹੈ ?
3. ਡਾਟਾ ਟਾਈਪ ਕੀ ਹੁੰਦਾ ਹੈ ?

Answer to objective questions

Q	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
Q1	a	b	d	b	c	c	d	b	a	a
Q4	T	T	T	T	T					
Q5	tokens	const	-2	6	( )					

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ 4

ਓਪਰੇਟਰਸ ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ
Operators and Expressions

4.0.0 ਭੂਮਿਕਾ (introduction)

ਓਪਰੇਟਰਸ (operators) ਭਾਸ਼ਾ ਦੀ ਕਿਰਿਆ ਹਨ, ਜੋ ਉਪਭੋਗਤਾ ਨੂੰ ਮੁੱਲਾਂ ਦੀ ਗਣਨਾ ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦੇ ਹਨ। ਗਣਿਤ ਅਤੇ ਆਮ ਕਾਰਜ ਕਰਨ ਲਈ ਓਪਰੇਟਰਸ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਇਕ formula ਹੈ ਜੋ ਕਿ ਇਕ ਜਾਂ ਇਕ ਤੇ ਵੱਧ ਓਪਰੈਂਡਸ (operands) ਦੇ ਬਣੇ ਹੋਏ ਹੁੰਦੇ ਹਨ ਅਤੇ ਓਪਰੇਟਰਸ ਜ਼ਿਰੋ ਜਾਂ ਇਕ ਜਾਂ ਇਕ ਤੋਂ ਵੱਧ ਹੋ ਸਕਦੇ ਹਨ ਜੋ ਕਿ ਇਕ ਦੂਜੇ ਨਾਲ ਮੁੱਲਾਂ ਦੀ ਗਣਨਾ ਕਰਨ ਲਈ ਜੁੜੇ ਹੁੰਦੇ ਹਨ। ਹੱਲ ਕੀਤੇ ਗਏ ਮੁੱਲ ਨੂੰ ਭਵਿੱਖ ਵਿਚ ਵਰਤਣ ਲਈ, ਵੇਰੀਏਬਲ ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਕੰਮ ਅਸਾਇਨਮੈਂਟ (assignment) ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋਂ ਦੁਆਰਾ ਕੀਤੇ ਜਾਂਦੇ ਹਨ।

4.1.0 ਓਪਰੇਟਰਸ ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ (Operators and Expressions)

ਬੁਨਿਆਦੀ ਅਰਥਮੈਟਿਕ ਅਤੇ ਲੌਜ਼ਿਕਲ (logical) ਓਪਰੇਸ਼ਨ ਨੂੰ ਓਪਰੇਟ ਕਰਨ ਲਈ ਓਪਰੇਟਰਾਂ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਓਪਰੇਟਰਕੁਨੈਕਟਰਾਂ (connectors) ਦੀ ਤਰ੍ਹਾਂ ਕੰਮ ਕਰਦੇ ਹਨ ਅਤੇ ਉਹ ਇਸ ਗੱਲ ਦਾ ਸੰਕੇਤ ਦਿੰਦੇ ਹਨ ਕਿ ਕਿਸ ਟਾਈਪ ਦਾ ਕੰਮ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਮੁੱਲ ਜੋ ਇਨ੍ਹਾਂ ਓਪਰੇਟਰਾਂ ਦੁਆਰਾ ਓਪਰੇਟ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ਨੂੰ ਔਪਰੈਂਡ (operand) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਦਾ ਫਾਰਮੈਟ (format) ਹੈ :

ਓਪਰੈਂਡ1 ਓਪਰੇਟਰ ਓਪਰੈਂਡ2 (Operand1 operator Operand2)

ਔਪਰੈਂਡ2 ਦਾ ਮੁੱਲ ਔਪਰੈਂਡ1 ਤੇ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਸ ਦਾ ਨਿਰਦੇਸ਼ ਓਪਰੇਟਰ ਦੁਆਰਾ ਹੁੰਦਾ ਹੈ

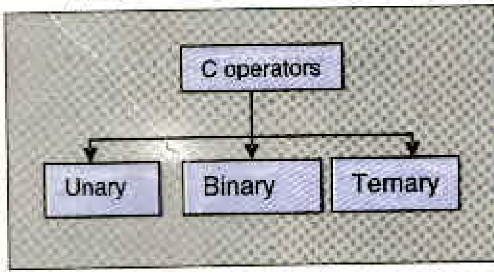
4.1.1 ਐਕਸਪ੍ਰੇਸ਼ਨ (expression)

ਉਹ ਹੈ ਜੋ ਕਿ ਇਕ ਨਤੀਜੇ ਨੂੰ ਵਾਪਿਸ ਕਰਦਾ ਹੈ। ਅਸੀ ਇਸ ਨਤੀਜੇ ਦਾ ਇਸਤੇਮਾਲ ਜਿਸ ਤਰ੍ਹਾਂ ਚਾਹੁੰਦੇ ਹਾਂ ਆਪਣੇ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕਰ ਸਕਦੇ ਹਾਂ । ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਇੰਨਾ ਸਾਧਾਰਣ ਹੋ ਸਕਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਇਕ ਮੁੱਲ ਜਾਂ ਇਕ ਗੁੰਝਲਦਾਰ ਵੱਡੀ ਗਣਨਾਂ ਵੀ ਹੋ ਸਕਦਾ ਹੈ । ਇਹ ਦੋ ਚੀਜ਼ਾਂ ਦੇ ਬਣੇ ਹੁੰਦੇ ਹਨ, ਓਪਰੇਟਰ ਅਤੇ ਓਪਰੈਂਡ (operator & operand).

ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਦੀਆਂ ਕੁਝ ਮਿਸਾਲਾਂ :

2+3*4, -1+3, (2+3)*4

ਸੀ (C) ਓਪਰੇਟਰਾਂ ਨੂੰ ਤਿੰਨ ਮੁੱਖ ਸ਼੍ਰੇਣੀਆਂ ਵਿਚ ਵੰਡਿਆ ਗਿਆ ਹੈ :-



ਯੂਨੇਰੀ ਓਪਰੇਟਰ (Unary operator): ਉਹ ਓਪਰੇਟਰ ਜਿਹੜਾ ਕੇਵਲ ਇਕ ਓਪਰੈਂਡ (operand) ਤੇ ਹੀ ਕੰਮ ਕਰਦਾ ਹੈ।

ਯੂਨੇਰੀ ਮਾਈਨਸ (Unary minus): ਇਸ ਓਪਰੇਟਰ ਦਾ ਇਕ ਮਹੱਤਵਪੂਰਨ ਭਾਗ ਹੁੰਦਾ ਹੈ। ਯੂਨੇਰੀ ਮਾਈਨਸ (unary minus) ਦੇ ਨਾਲ ਸੰਬੰਧਿਤ ਕੋਈ ਵੀ ਪੋਜਿਟਿਵ ਓਪਰੈਂਡ (operand) ਦੀ ਮੁੱਲਾਂ ਵਿਚ ਤਬਦੀਲੀ ਹੁੰਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

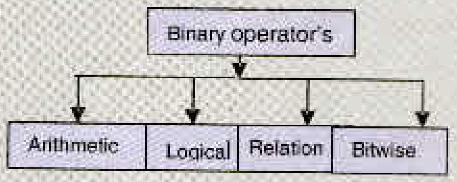
Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ a= 5 , b =7 and c = a+(-b)
c = 5+ (-7) = -2

ਕਿਉਂ ਜੋ b ਮੁੱਢ ਵਿਚ ਇਕ ਪੋਜ਼ਿਟਿਵ ਪੂਰਨ-ਅੰਕ *(integer)* ਵੇਰਿਏਬਲ ਹੁੰਦੀ ਹੈ ਜਦੋਂ ਯੂਨੇਰੀ ਮਾਈਨਸ (unary minus) ਦੁਆਰਾ ਓਪਰੇਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਸਦਾ ਮੁੱਲ ਬਦਲ ਜਾਂਦਾ ਹੈ ।

4.1.2 ਬਾਇਨਰੀ ਓਪਰੇਟਰ (Binary operators)

ਇਹ ਓਪਰੇਟਰ ਦੋ ਓਪਰੈਂਡਸ (operands) ਤੇ ਕੰਮ ਕਰਦੇ ਹਨ ਉਨ੍ਹਾਂ ਨੂੰ ਅਗੇ ਸ਼੍ਰੇਣੀ ਬੱਧ ਕੀਤਾ ਗਿਆ ਹੈ



ਅਰਥਮੈਟਿਕ ਓਪਰੇਟਰਜ਼ (Arithmetic Operators):

ਸੀ (C) ਭਾਸ਼ਾ ਦੇ ਪੰਜ ਅਰਥਮੈਟਿਕ ਓਪਰੇਟਰ :

Operator	Meaning
+	Addition or Unary Plus
-	Subtraction or Unary Minus
*	Multiplication
/	Division
%	Modulus Operator

ਓਪਰੇਟਰ %, ਮਾਡਯੂਲਸ ਓਪਰੇਟਰ (**Modulus Operator**) ਪੂਰਨ-ਅੰਕ ਵੰਡ ਬਾਅਦ ਰਿਮੇਂਡਰ (remainder) ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਓਪਰੇਟਰ (Operator) ਨੂੰ ਫਲੋਟਿੰਗ (floating) ਬਿੰਦੂ ਅੰਕਾਂ ਨਾਲ ਨਹੀਂ ਵਰਤਿਆ ਜਾ ਸਕਦਾ।

ਮਿਸਾਲ : ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) x%y ਵਿਚ ਜਦੋਂ x ਨੂੰ y ਨਾਲ ਭਾਗ (Divide) ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਰਿਮੇਂਡਰ (remainder) ਪੈਦਾ ਹੁੰਦਾ ਹੈ ਅਤੇ ਜਦੋਂ ਪੂਰਨ-ਅੰਕ (integer) y ਨਾਲ ਪੂਰਾ ਭਾਗ ਹੋ ਜਾਂਦਾ ਹੈ ਤਾਂ ਰਿਮੇਂਡਰ (remainder) ਜ਼ੀਰੋ ਹੁੰਦਾ ਹੈ, ਤਦ ਟਰੰਕੇਟਸ ਅਤੇ ਫਰੈਕਸ਼ਨਲ (Truncates & fractional) ਭਾਗ ਹੇਠ ਲਿਖੇ ਅਨੁਸਾਰ ਦਿਖਾਈ ਦਿੰਦੇ ਹਨ :

ਮੰਨ ਲਓ a=12.5 ਅਤੇ b=2.0 ਇਹ ਫਲੋਟਿੰਗ ਬਿੰਦੂ ਹਨ ਜੋ ਅਸਮਾਨ ਅਰਥਮੈਟਿਕ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਹਨ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਆਊਟਪੁਟ ਹੇਠਾਂ ਦਿਖਾਏ ਗਏ ਹਨ।

Expression	Value
a + b	14.5
a - b	10.5
a * b	25.0
a / b	6.25

ਮੰਨ ਲਓ a=10 ਅਤੇ b=3 ਜੋ ਪੂਰਨ-ਅੰਕ ਅਸਮਾਨ ਹਨ ਤੇ ਅਰਥਮੈਟਿਕ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਹਨ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਨਤੀਜੇ ਹੇਠਾਂ ਦਿਖਾਏ ਗਏ ਹਨ।

Expression	Value
a + b	13
a - b	7
a * b	30
a / b	3
a % b	1

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

4.1.3 ਆਪਰੇਸ਼ਨ ਅਤੇ ਹਿਰੈਚੀਕਲ ਆਰਡਰ

ਭਾਵੇਂ ਕਿ ਐਕਸਪ੍ਰੇਸ਼ਨ, ਓਪਰੇਟਰਸ (operators) ਅਤੇ ਵੇਰੀਏਬਲਜ਼ ਦੇ ਨਾਲ ਪ੍ਰਗਟਾਏ ਜਾਂਦੇ ਹਨ ਪਰ ਸੀ ਕੰਪਾਈਲਰ ਇਹਨਾਂ ਨੂੰ ਚਲਾਉਣ ਲਈ ਇਕ ਵਿਸ਼ੇਸ਼ ਰੂਪ ਵਿਚ ਦਰਜਾਬੰਦ ਕਰਦੇ ਹਨ ਕਿਉਂਕਿ ਸੀ (C) ਕੰਪਾਈਲਰ ਦਰਜਾਬੰਦੀ ਨੂੰ ਅਪਣਾਉਂਦੇ ਹਨ ਜਿਸ ਤਰ੍ਹਾਂ ਕਿ ਹੇਠਾਂ ਦਰਸਾਇਆ ਗਿਆ ਹੈ:

1.	* / %	ਖੱਬੇ ਤੋਂ ਸੱਜੇ ਪਾਸੇ
2.	+ -	ਖੱਬੇ ਤੋਂ ਸੱਜੇ ਪਾਸੇ

ਓਪਰੇਟਰਾਂ ਦੀ ਪ੍ਰਾਥਮਿਕਤਾ ਜਾਂ ਓਪਰੇਟਰਾਂ ਦੀ ਦਰਜਾਬੰਦੀ (Hierarchy of operator's or The Precedence of the Operators):

ਓਪਰੇਟਰ ਪ੍ਰਾਥਮਿਕਤਾ ਉਸ ਕ੍ਰਮ ਨੂੰ ਬਿਆਨਦੀ ਹੈ ਜਿਸ ਵਿਚ ਸੀ (C) ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਨੂੰ ਪੜ੍ਹਦਾ ਹੈ।

ਮਿਸਾਲ ਲਈ, ਐਕਸਪ੍ਰੇਸ਼ਨ a = 4+b*2 ਜਿਸ ਵਿਚ ਦੋ ਓਪ੍ਰੇਸ਼ਨ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ, ਇਕ ਵਧਾਓ ਅਤੇ ਦੂਜਾ ਗੁਣਾ। ਕੀ ਸੀ (C) ਕੰਪਾਈਲਰ 4+b ਨੂੰ ਪਹਿਲਾਂ ਮੁੱਲਾਂਕਣ ਕਰਦਾ ਹੈ ਤੇ ਫਿਰ ਪਰਿਣਾਮ ਨੂੰ 2 ਨਾਲ ਗੁਣਾ ਕਰਦਾ ਹੈ ਜਾਂ ਕੀ ਇਹ b*2 ਦਾ ਮੁੱਲਾਂਕਣ ਕਰਦਾ ਹੈ ਅਤੇ ਫਿਰ ਪਰਿਣਾਮ ਵਿਚ 4 ਜਮ੍ਹਾਂ ਕਰਦਾ ਹੈ?

ਓਪਰੇਟਰ ਪ੍ਰਾਥਮਿਕਤਾ ਚਾਰਟ ਵਿਚ ਉੱਤਰ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ। ਚਾਰਟ ਵਿਚ ਉੱਚ ਓਪਰੇਟਰਾਂ ਨੂੰ ਉੱਚ ਪ੍ਰਾਥਮਿਕਤਾ ਮਿਲਦੀ ਹੈ ਭਾਵ ਸੀ (C) ਕੰਪਾਈਲਰ (compiler) ਉਨ੍ਹਾਂ ਦਾ ਪਹਿਲਾਂ ਮੁਲਾਂਕਣ ਕਰਦਾ ਹੈ।

ਹੇਠਾਂ ਦਰਸਾਏ ਟੇਬਲ ਵਿਚ, **x** ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਦੀ ਪ੍ਰਤਿਨਿਧਤਾ ਕਰਦਾ ਹੈ, ਜਿਸ ਵਿਚ ਓਪਰੇਟਰ ਨੂੰ ਉਹਨਾ ਦੀ ਪ੍ਰਾਥਮਿਕਤਾ ਦੇ ਅਧਾਰ ਤੇ ਘਟਦੇ ਕ੍ਰਮ ਅਨੁਸਾਰ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।

<table>
<tr><th>Operator classification</th><th>Operators</th></tr>
<tr><td>parentheses</td><td>( )</td></tr>
<tr><td>postfix operators</td><td>[ ] , x++ , x--</td></tr>
<tr><td>unary / prefix operators</td><td>++x, --x, +x, -x , !</td></tr>
<tr><td>Multiplicative</td><td>* , / , %</td></tr>
<tr><td>Additive</td><td>+ ,-</td></tr>
<tr><td>Relational</td><td><, >, >=, <=</td></tr>
<tr><td>equality (boolean)</td><td>= = , !=</td></tr>
<tr><td>logical AND</td><td>&&</td></tr>
<tr><td>logical OR</td><td>||</td></tr>
<tr><td>assignment</td><td>=</td></tr>
</table>

ਆਓ ਹੁਣ ਨਿਮਨਲਿਖਤ ਤੇ ਵਿਚਾਰ ਕਰੀਏ ਤਾ ਜੋ ਔਪਰੇਸ਼ਨਾਂ ਦੀ ਦਰਜਾਬੰਦੀ ਦਾ ਫੈਸਲਾ ਹੋ ਸਕੇ ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਦਾ ਮੁੱਲਾਂਕਣ ਹੋਵੇ: **a = 3 * 3 / 7 + 3 / 3 + 8 – 2 + 5 / 8**

a = 9 / 7 + 3 / 3 + 8 - 2 + 5 / 8	**operation : ***
a = 1 + 3 / 3 + 8 – 2 + 5 / 8	**operation : / as 9/7 gives integer 1 not 1.3**
a = 1 + 1+ 8 – 2 + 5/ 8	**operation : / as 3/3 gives integer 1**
a = 1 + 1+ 8 – 2 + 0	**operation : / as 5 /8 gives 0 , 5 & 8 are integer constants and returns integer value**
a = 2 + 8 – 2 +0	**operation : +**
a = 10 – 2 +0	**operation : +**
a = 8 + 0	**operation : -**
a = 8	**operation : +**

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ 1: ਹੇਠਾਂ ਲਿਖੇ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਬੁਨਿਆਦੀ ਅਰਥਮੈਟਿਕ ਓਪਰੇਸ਼ਨਸ ਨੂੰ ਦਰਸਾਉਣ ਲਈ ਵੈਰੀਏਬਲਜ਼ (variables) ਨੂੰ ਕਾਂਸਟੈਂਟ (constant) ਮੁੱਲਾਂ ਦੇ ਨਾਲ ਪਰਿਭਾਸ਼ਤ (define) ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਹਰ ਇਕ ਅਰਥਮੈਟਿਕ ਓਪਰੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਦੇ ਰਾਹੀ ਲਾਗੂ ਕੀਤਾ ਗਿਆ ਹੈ

```
# include <stdio.h>
void main ()
{
        int n1 , n2;
        int sum ,diff, prod ,quotient , remainder ;
        n1 = 5 ;
        n2 = 3 ;
       sum = n1+ n2 ;
       diff  = n1 – n2 ;
       prod =  n1 * n2 ;
       quotient = n1/ n2;
       remainder = n1 % n2 ;
       printf("SUM  = %d\n " ,sum);
        printf("DIFFERENCE = %d\n " ,diff);
      printf("PRODUCT = %d\n " ,prod);
      printf("QUOTIENT = %d\n " ,quotient);
      printf("REMAINDER = %d\n " ,remainder );
}
```

Output

```
SUM          = 8
DIFFERENCE   = 2
PRODUCT      = 15
QUOTIENT     = 1
REMAINDER    = 2
```

ਮਿਸਾਲ 2: ਹੇਠਾਂ ਲਿਖੇ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਕੁਲ ਅਦਾਇਗੀ ਦੀ ਗਣਨਾ ਕੀਤੀ ਗਈ ਹੈ ਜਿਸ ਦੀ ਪ੍ਰਤੀ ਘੰਟਾ ਦਰ = 80.75 ਹੈ ਅਤੇ ਕੰਮਕਾਜੀ ਘੰਟੇ = 45.75 ਹਨ। ਇਥੇ ਸਮੁੱਚੀ ਅਦਾਇਗੀ ਕੰਮਕਾਜੀ ਘੰਟਿਆਂ* ਦੀ ਪ੍ਰਤੀ ਘੰਟਾ ਦਰ = (Hours*rate) ਹੈ

```
#include <stdio.h>
void main()
{                    //declaring float variables
  float hourRate ;
 float hourwrkd ;
 float grosspay;
                    // assigning values to variables
 hourRate = 80.75;
 hourwrkd = 45.75 ;
 grosspay = hourRate* hourwrkd;
              // displaying the variables on the screen
printf(" Hourly Rate = Rs. %f\n Hour worked = %f
 hours\n Gross pay = Rs. %f" ,hourRate,
hourwrkd* ;  grosspay) ;

}
```

Output

```
Hourly Rate = Rs. 80.750000
Hours worked = 45.75 0000
Gross Pay = Rs. 36943.312500
```

ਉਪਰ ਦਿਤੇ ਪਰਿਣਾਮ ਨੂੰ ਸੁਧਾਰੋ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਲਿਖੋ ਜਿਹੜਾ ਹੇਠਾਂ ਦਿਤੇ ਪਰਿਣਾਮਾਂ ਨੂੰ ਦਰਸਾਏ

(Modify the above output & write the program to display the output as):

```
SUM   (5+3)           = 8
DIFFERENCE (5-3)      = 2
PRODUCT  (5*3)        = 15
QUOTIENT   (5/3)      = 1
REMAINDER  (5%3)      = 2
```

ਸੀ (C) ਵਿਚ ਹੇਠਾਂ ਦਿਤੇ ਪ੍ਰੋਗਰਾਮ ਦਾ ਪਰਿਣਾਮ ਕੀ ਹੋਵੇਗਾ

```
#include <stdio.h>
void main()
{
  int months, days;
printf("Enter days\n")  // input 3 values 45 , 265 ,364
  scanf("%d", &days);
  months = days /30 ;
  days = days % 30 ;
  printf("Months =%d Days = %d" , months, days);
}
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

4.2.0 ਰਿਲੇਸ਼ਨਲ ਅਤੇ ਲੌਜਿਕਲ ਓਪਰੇਟਰ (Relational & Logical Operator)

ਰਿਲੇਸ਼ਨਲ ਓਪਰੇਟਰ (Relational Operator)

ਇਹ ਓਪਰੇਟਰਜ਼ ਦੋ ਓਪਰੈਂਡਸ ਦੀ ਤੁਲਨਾ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਇਹ ਦੋ ਕਾਂਸਟੈਂਟਸ ਜਾਂ ਵੇਰੀਏਬਲਜ਼ ਦੇ ਵਿਚ ਸੰਬੰਧ ਨੂੰ ਦਰਸਾਉਂਦੇ ਹਨ। ਇਹਨਾਂ ਦਾ ਨਤੀਜਾ (result) ਸਹੀ (true) ਜਾਂ ਗਲਤ (false) ਮੁੱਲ ਹੁੰਦਾ ਹੈ। ਜਿਥੇ ਕਿ ਸਹੀ (true) ਮੁੱਲ ਗੈਰ ਜ਼ੀਰੋ ਜਾਂ ਇਕ ਹੁੰਦਾ ਹੈ ਅਤੇ ਗਲਤ (false) ਮੁੱਲ ਜ਼ੀਰੋ ਹੁੰਦਾ ਹੈ।

ਦੋ ਵਿਦਿਆਰਥੀਆਂ ਦੇ ਨੰਬਰਾਂ ਜਾਂ ਦੋ ਵਿਅਕਤੀਆਂ ਦੀ ਤਨਖਾਹ ਦੀ ਤੁਲਨਾ ਕਰਨ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ, ਅਸੀਂ ਉਨ੍ਹਾਂ ਦੀ ਤੁਲਨਾ ਰਿਲੇਸ਼ਨਲ (relational) ਓਪਰੇਟਰਾਂ ਦੇ ਨਾਲ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਹ ਸਧਾਰਣ ਰਿਲੇਸ਼ਨਲ (relational) ਐਕਸਪ੍ਰੇਸ਼ਨ ਵਿਚ ਕੇਵਲ ਇਕ ਰਿਲੇਸ਼ਨਲ ਓਪਰੇਟਰ ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਹ ਨਿਮਨਲਿਖਤ ਰੂਪ ਧਾਰਦਾ ਹੈ :-

exp1 relational operator exp2

ਜਿਥੇ ਐਕਸਪ੍ਰੇਸ਼ਨ 1 (exp1) ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ 2 (exp2) ਐਕਸਪ੍ਰੇਸ਼ਨ (expressions) ਹਨ, ਜੋ ਕਿ ਉਨ੍ਹਾਂ ਦੇ ਸਧਾਰਣ ਕਾਂਸਟੈਂਟ (simple constant) ਵੇਰੀਏਬਲਜ਼ (variable) ਜਾਂ ਉਨ੍ਹਾਂ ਦਾ ਸੁਮੇਲ (combination) ਹੋ ਸਕਦੀਆਂ ਹਨ ।

Operator	Meaning
<	ਤੋਂ ਘੱਟ ਹੈ
<=	ਤੋਂ ਘੱਟ ਜਾਂ ਬਰਾਬਰ ਹੈ
>	ਤੋਂ ਵੱਡਾ ਹੈ
>=	ਤੋਂ ਵੱਡਾ ਜਾਂ ਬਰਾਬਰ ਹੈ
==	ਦੇ ਬਰਾਬਰ
!=	ਦੇ ਬਰਾਬਰ ਨਹੀਂ

ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਨ੍ਹਾਂ ਓਪਰੇਟਰਾਂ ਨੂੰ ਵਰਤਣ ਲਈ ਅਸੀਂ if ਸਟੇਟਮੈਂਟ (if statement) ਵਰਤਾਂਗੇ, ਇਹ ਕੰਡੀਸ਼ਨਲ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ (statement) ਹੁੰਦਾ ਹੈ ਜੋ ਸਹੀ ਜਾਂ ਗਲਤ ਮੁੱਲ ਲਈ ਕੰਡੀਸ਼ਨ ਦੀ ਪਰਖ ਕਰਦਾ ਹੈ।

ਮਿਸਾਲ : ਪਰਖ ਕਰਨ ਲਈ ਕਿ ਦੋਵੇਂ ਅੰਕ ਸਹੀ ਹਨ।

```
#include <stdio.h>
main()
{
 int x =2, y =2;
 if  (x = = y)
     printf("x and y are equal\n") ;
}
```

Output:
x and y are equal

ਮਿਸਾਲ: ਓਪਰੇਟਰਾਂ ਦੀ ਅਸਮਾਨਤਾ ਪਰਖ ਕਰਨ ਲਈ।

```
#include <stdio.h>
main()
{
 float a =3.533 ,b = 0.0232;
 if (a>b)
    printf("a is greater than b\n");
}
```

Output:
a is greater than b

4.2.1 ਲੋਜੀਕਲ ਓਪਰੇਟਰ (Logical Operators)

ਇਹ ਓਪਰੇਟਰਜ਼ ਸੀ (C) ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਫੈਸਲਾ ਲੈਣ ਦੇ ਕੰਮ ਆਉਂਦੇ ਹਨ। ਜੋ ਕਿ ਲੋਜੀਕਲ ਅਤੇ ਰਿਲੇਸ਼ਨਲ ((logical & relational) ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਦੀ ਤੁਲਨਾ (comparison) ਕਰਦੇ ਹਨ। ਸੀ (C) ਦੇ ਨਿਮਨਲਿਖਿਤ ਲੋਜਿਕਲ (logical) ਓਪਰੇਟਰ ਹਨ :

Operator	Meaning
&&	Logical AND
||	Logical OR
!	Logical NOT

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਲੌਜੀਕਲ AND (&&) ਓਪਰੇਟਰ (Logical AND (&&) operator):

ਇਹ ਓਪਰੇਟਰਸ ਦੇ ਐਕਸਪ੍ਰੈਸ਼ਨ ਜਾਂ ਕੰਡੀਸ਼ਨਸ ਨੂੰ ਰਿਲੇਸ਼ਨਲ (relational) ਓਪਰੇਟਰਸ ਦੇ ਨਾਲ ਗਣਨਾ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਜੇਕਰ ਲੋਜਿਕਲ (logical) ਓਪਰੇਟਰ ਦੇ ਖੱਬੇ ਪਾਸੇ ਅਤੇ ਸੱਜੇ ਪਾਸੇ ਦੀ ਦੋਵੇਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਜਾਂ ਕੰਡੀਸ਼ਨਸ ਸਹੀ (true) ਹੋਣ ਤਾਂ ਸਮੁੱਚਾ ਮਿਸਰਤ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਸਹੀ ਹੁੰਦਾ ਹੈ।

ਮਿਸਾਲ : ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਮੰਨ ਲਓ ਕਿ **a > b && x = = 10**

ਖੱਬੇ ਵੱਲ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਅਤੇ ਸੱਜੇ ਸਟੇਟਮੈਂਟ x = = 10 ਹੈ, ਇਥੇ ਸਮੁੱਚਾ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਸਹੀ ਹੈ। ਕਿਉਂਕਿ "b" ਨਾਲੋਂ "a" ਵੱਡਾ ਹੈ ਅਤੇ "x" 10 ਦੇ ਬਰਾਬਰ ਹੈ ਇਸ ਲਈ ਦੋਵੇਂ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਸਹੀ ਹਨ ।

ਲੌਜੀਕਲ OR (¦¦) [Logical OR (¦¦)]

ਲੌਜੀਕਲ OR (**Logical OR**) ਦੇ ਕੰਡੀਸ਼ਨ ਜਾਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਦੇ ਸੁਮੇਲ (combine) ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਜੇਕਰ ਦੋਵੇਂ ਕੰਡੀਸ਼ਨ ਜਾਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਵਿਚੋਂ ਇਕ ਵੀ ਸਹੀ (true) ਜਾਂ ਗਲਤ (false) ਹੈ ਤਾਂ ਮੁੱਲਾਂਕਣ ਸਹੀ ਹੋਵੇਗਾ।

ਮਿਸਾਲ : ਸਟੇਟਮੈਂਟ (statement) ਤੇ ਵਿਚਾਰ ਕਰੋ **a < m ¦¦ a < n**

ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਸਹੀ ਹੁੰਦਾ ਹੈ ਜੇਕਰ ਉਨ੍ਹਾਂ ਵਿਚੋਂ ਕੋਈ ਇਕ ਸਹੀ ਹੈ ਜਾਂ ਉਹ ਦੋਵੇਂ ਸਹੀ ਹਨ । ਇਹ ਸਹੀ ਮੁਲਾਂਕਣ ਹੁੰਦਾ ਹੈ ਜੇਕਰ a ਜੋ m ਜਾਂ n ਨਾਲੋਂ ਘੱਟ ਹੁੰਦਾ ਹੈ ਅਤੇ a ਜਦੋਂ ਦੋਵੇਂ m ਅਤੇ n ਨਾਲੋ ਘੱਟ ਹੁੰਦਾ ਹੈ।

ਲੌਜੀਕਲ NOT (!) [Logical NOT (!)]

ਲੋਜਿਕਲ NOT ਓਪਰੇਟਰ ਇਕਹਰਾ ਐਕਸਪ੍ਰੈਸ਼ਨ (expressions) ਲੈਂਦਾ ਹੈ ਅਤੇ ਇਹ ਸਹੀ ਮੁਲਾਂਕਣ ਕਰਦਾ ਹੈ ਜੇਕਰ ਐਕਸਪ੍ਰੈਸ਼ਨ (expressions) ਗਲਤ ਹੁੰਦਾ ਹੈ ਅਤੇ ਮੁੱਲਾਂਕਣ ਗਲਤ ਕਰਦਾ ਹੈ ਜੇਕਰ ਐਕਸਪ੍ਰੈਸ਼ਨ (expressions) ਸਹੀ ਹੈ ਤਾਂ ਦੂਜੇ ਸ਼ਬਦਾਂ ਵਿਚ ਇਹ ਕੇਵਲ ਐਕਸਪ੍ਰੈਸ਼ਨ (expressions) ਦੇ ਮੁੱਲ ਨੂੰ ਹੀ ਉੱਲਟ ਕਰ ਦਿੰਦਾ ਹੈ।

ਮਿਸਾਲ : ਸਟੇਟਮੈਂਟ (statement) ਤੇ ਵਿਚਾਰ ਕਰੋ **! (x >= y)**

NOT (expressions) ਐਕਸਪ੍ਰੈਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਸਹੀ ਤਾਂ ਹੀ ਹੁੰਦਾ ਹੈ ਜੇਕਰ x ਦਾ ਮੁੱਲ ਨਾ ਤਾਂ ਵੱਡਾ ਹੈ ਅਤੇ ਨਾ ਹੀ y ਦੇ ਬਰਾਬਰ ਹੈ। NOT ਓਪਰੇਟਰ ਅਕਸਰ ਐਕਸਪ੍ਰੈਸ਼ਨ (expressions) ਵਿਚ ਸਟਰਿੰਗ ਵੇਰੀਏਬਲਜ਼ (string variables) ਦੇ ਲੋਜੀਕਲ (logical) ਮੁੱਲ ਦੇ ਉਲਟ ਨੂੰ ਹੀ ਇਸਤੇਮਾਲ ਕਰਦਾ ਹੈ।

if (! a) ਇਹ ਕਹਿਣ ਦਾ ਹੋਰ ਢੰਗ ਹੈ ਕਿ **if (a == 0)**

ਉਪਰੋਕਤ ਕਹੀਆਂ ਗਈਆਂ ਮਿਸਾਲਾਂ ਦੇ ਆਧਾਰ ਤੇ ਆਓ ਨਿਮਨਲਿਖਿਤ ਦੇ ਮਿਸਾਲਾਂ ਤੇ ਵਿਚਾਰ ਕਰੀਏ :-

- ਜੇਕਰ ਅਸੀਂ ਦੋ ਕੰਡੀਸ਼ਨਸ ਨੂੰ ਨਾਲੋ-ਨਾਲ ਪਰਖਣਾ ਹੋਵੇ ਜਿਵੇਂ ਇਮਤਿਹਾਨ ਵਿਚ ਵਿਦਿਆਰਥੀਆਂ ਦਾ ਗਰੇਡ 1 ਦੇ ਬਰਾਬਰ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਅਤੇ ਉਸਦੇ ਕੁਲ ਅੰਕ 550 ਤੋਂ ਵੱਧ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ ਫਿਰ ਅਸੀਂ ਕਿਸੇ ਕੰਡੀਸ਼ਨ ਨਾਲ ਲਿਖ ਸਕਦੇ ਹਾਂ **if (grade = = 1 && tmarks > 550)**

ਇਸ ਕੇਸ ਵਿਚ ਦੋਵੇਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਸਹੀ ਹੋਣੀਆਂ ਚਾਹੀਦੀਆਂ ਹਨ ਤਾਂ ਜੋ ਇਹ ਸੰਪੂਰਨ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਦਾ ਸਹੀ ਮੁੱਲ ਵਾਪਿਸ ਕਰ ਸਕਣ ।

ਸਮੱਸਿਆ : ਨਿਮਨਲਿਖਿਤ ਸਟੇਟਮੈਂਟ (statement) ਲਈ ਕੀ ਸਪਸ਼ਟੀਕਰਨ ਹੋਵੇਗਾ

if(grade= = 1 ¦¦ tmarks>550)

4.2.2 ਅਸਾਈਨਮੈਂਟ ਓਪਰੇਟਰ (Assignment operator)

ਅਸਾਈਨਮੈਂਟ (assignment/ਸਮਰਪਣ) ਓਪਰੇਟਰ ਨੂੰ "set equal to" ਕਿਹਾ ਜਾਂਦਾ ਹੈ। **ਅਸਾਈਨਮੈਂਟ** (assignment /ਸਮਰਪਣ) ਓਪਰੇਟਰ ਲਈ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਵਿਸ਼ੇਸ਼ ਚਿੰਨ੍ਹ ਹਨ (=)

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

This:	Is identical to this:	Operation Name
x+=y;	x=x+y;	Addition assignment
x-=y;	x=x–y;	Subtraction assignment
x*=y;	x=x*y;	Multiplication assignment
x/=y;	x=x/y;	Division assignment
x%=y;	x=x%y;	Remainder assignment (int only)

ਇਹ ਇਸ ਤਰ੍ਹਾਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ **ਓਪਰੈਂਡ1 = ਓਪਰੈਂਡ2 (operand1 = operand2)**
ਅਸਾਈਨਮੈਂਟ (assignment) ਓਪਰੇਟਰ ਨਿਸ਼ਚਿਤ ਕਰਦਾ ਹੈ ਕਿ ਪਹਿਲਾ ਓਪਰੈਂਡ (operand) ਦੂਜੇ ਔਪਰੈਂਡ (operand) ਦੇ ਮੁੱਲ ਨੂੰ ਅਸਾਈਨਮੈਂਟ (assignment) ਹੈ। ਓਪਰੈਂਡ 2 (operand 2) ਦਾ ਮੁੱਲਾਂਕਣ ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਪਰਿਣਾਮ (result) ਓਪਰੈਂਡ 1 (operand 1) ਨੂੰ ਅਸਾਈਨ ਕੀਤਾ ਗਿਆ ਹੈ।

ਮਿਸਾਲ : **x = y;** /* ਦਾ ਮੁੱਲ y ਦੇ ਬਰਾਬਰ ਨਿਰਧਾਰਤ ਕੀਤਾ ਗਿਆ ਹੈ। *\

ਸਟੇਟਮੈਂਟ (statement) ਤੇ ਵਿਚਾਰ ਕਰੋ

i=i+5; **ਇਸਨੂੰ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ i+=5;**
i=i*5; **ਇਸਨੂੰ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ i*=5**

ਮਲਟੀਪਲ ਅਸਾਈਨਮੈਂਟ ਸਟੇਟਮੈਂਟ (Multiple assignment statement)

ਜਦੋਂ ਵੀ ਇੱਕਲੇ ਸਟੇਟਮੈਂਟ (statement) ਵਿਚ ਵੱਖ ਵੱਖ ਵੇਰੀਏਬਲਜ਼ (variables) ਨੂੰ ਇਨਿਸ਼ਿਯਲਾਇਜ਼ *(initialize)* ਕਰਨਾ ਹੋਵੇ। ਇਸ ਪ੍ਰਕਾਰ ਦੇ ਇਨਿਸ਼ਿਯਲਾਇਜ਼ੇਸ਼ਨ (initialization) ਦੀ ਟਾਈਪ ਨੂੰ ਵਧੇਰੇ ਅਸਾਈਨਮੈਂਟ (assignment) ਓਪਰੇਟਰਾਂ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਇਸ ਪ੍ਰਕਾਰ, ਇਸਦਾ ਨਾਮ ਹੈ ਮਲਟੀਪਲ ਅਸਾਈਨਮੈਂਟ (assignment)

ਮਿਸਾਲ
i **int x = y = z = 10;**
ii **Float a = b =c = d = 5.75;**

ਪਹਿਲਾ ਸਟੇਟਮੈਂਟ (statement) ਤਿੰਨ int ਵੇਰੀਏਬਲ (variable) ਇਨਿਸ਼ਿਯਲਾਇਜ਼ *(initialize)* ਕਰਦਾ ਹੈ x,y ਅਤੇ z ਜਿਸਦੀ ਕੀਮਤ 10 ਹੈ। ਦੂਜਾ ਸਟੇਟਮੈਂਟ (statement) ਚਾਰ ਫਲੋਟ (float) ਵੇਰੀਏਬਲਜ਼ (variables) ਇਨਿਸ਼ਿਯਲਾਇਜ਼ *(initialize)* ਕਰਦਾ ਹੈ ਜਿਸਦੀ ਕੀਮਤ a, b, c, d ਤੋਂ 5.75 ਹੈ।

ਸਮਸਿਆਂ: ਜੇਕਰ x= 11, y= 6, ਪਰਿਣਾਮੀ (resultant) ਮੁੱਲਾਂ ਨੂੰ ਜੋ ਕਿ ਹੇਠਾਂ Value ਦੇ column ਵਿਚ ਦਿਖਾਏ ਗਏ ਹਨ ਨੂੰ ਪੜਤਾਲੋ

Expression	Value
x>9 && y! =3	**1**
x = = 5¦¦y! = 3	**1**
5&&y!=8¦¦0	**0**
!(x>9&&y!=23)	**0**

4.2.3 ਇਨਕਰੀਮੈਂਟ ਐਂਡ ਡਿਕਰੀਮੈਂਟ (ਵਾਧਾਉਣ ਅਤੇ ਘਟਾਉਣ) ਓਪਰੇਟਰਸ (Increment & Decrement Operators)

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਇਹ ਓਪਰੇਟਰ ਨਿਸ਼ਚਿਤ ਕਰਦੇ ਹਨ ਕਿ ਔਪਰੈਂਡ (operand) ਸੀ (C) ਸਟੇਟਮੇਂਟ (statement) ਦੇ ਸੰਦਰਭ ਵਿਚ ਔਪਰੈਂਡ (operand) ਦੇ ਨਾਲ ਅਗੇਤਰ (pre) ਜਾਂ ਪਿਛੇਤਰ (post) ਵਜੋਂ ਮੁੱਲ ਨੂੰ ਇਨਕਰੀਮੈਂਟ ਜਾਂ ਡਿਕਰੀਮੈਂਟ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।

Operator	Usage
++	increment the variable
--	decrement the variable

ਜਿਸ ਤਰ੍ਹਾਂ ++ i ਜਾਂ - - i
i++ ਜਾਂ i - -

ਇਨਕਰੀਮੈਂਟ (increment) ਅਤੇ ਡਿਕਰੀਮੈਂਟ (decrement) ਦੇ ਟਾਈਪਸ ਦੋ ਹੁੰਦੇ ਹਨ : 1. prefix 2. postfix

1. ਪ੍ਰਸੈਸਿੰਗ (prefix) ਵੇਰੀਏਬਲ ਲਈ ਪਹਿਲੇ ਇਨਕਰੀਮੈਂਟ (increment) ਜਾਂ ਡਿਕਰੀਮੈਂਟ (decrement) ਲਓ ਫਿਰ ਮੁੱਲ ਲਓ।
2. ਪ੍ਰਸੈਸਿੰਗ (postfix) ਵੇਰੀਏਬਲ ਲਈ ਮੁੱਲ ਲਓ ਅਤੇ ਫਿਰ ਇਨਕਰੀਮੈਂਟ (increment) ਜਾਂ ਡਿਕਰੀਮੈਂਟ (decrement).

ਮਿਸਾਲ 1 ਪੋਸਟ ਇਨਕਰੀਮੈਂਟ (Post Increment)

a = 10
b = a ++;
ਇਥੇ b ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਮੁੱਲ 10 ਹੋਵੇਗਾ ਅਤੇ a ਦਾ ਨਵਾਂ ਮੁੱਲ 11 ਹੋਵੇਗਾ।
b = a
a = a + 1;

ਮਿਸਾਲ 2 ਪੋਸਟ ਡਿਕਰੀਮੈਂਟ (Post Decrement)

a = b -- ;
ਇਹ ਦੋ ਸਮਰਪਣ ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਬਰਾਬਰ ਹੁੰਦੇ ਹਨ।
a = b; and b = b - 1;

ਮਿਸਾਲ 3 ਪ੍ਰੀ ਇਨਕਰੀਮੈਂਟ (Pre Increment)

a = 10
b = ++ a;
ਇਥੇ b ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਮੁੱਲ ਹੋਵੇਗਾ 11 ਅਤੇ ਨਵਾਂ ਮੁੱਲ ਹੋਵੇਗਾ 11
a = a+1,
b = a;

ਮਿਸਾਲ 4 ਪ੍ਰੀ ਇਨਕਰੀਮੈਂਟ (Pre Increment)

a = ++b;
ਇਹ ਦੋ ਅਸਾਈਨਡ (assigned) ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਬਰਾਬਰ ਹੁੰਦੇ ਹਨ।
b = b+1; and a = b;

ਮਿਸਾਲ 5 ਪ੍ਰੀ ਡਿਕਰੀਮੈਂਟ (Pre Decrement)

a = --b;
ਇਹ ਅਸਾਈਨਮੈਂਟ (assignment) ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਬਰਾਬਰ ਹਨ
b = b - 1; and a = b;

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Prefix ਇਨਕਰੀਮੈਂਟ ਅਤੇ ਡਿਕਰੀਮੈਂਟ (Increment & Decrement) :

ਮਿਸਾਲ 1:

```
k = 2;        /* k is assigned with 2 */
i = ++k;      /* Increments k by 1, then
assigns to i therefore i = 3 and k = 3*/
i = --k;      /* decrements k by 1, then
assigns to i therefore i = 2 and k = 2. */
```

Postfix ਇਨਕਰੀਮੈਂਟ & ਡਿਕਰੀਮੈਂਟ (increment & decrement) :

ਮਿਸਾਲ 2:

```
k = 2;        k is assigned with 2
i = k++;      /*Assigns i with k, then
increments k therefore i = 2 and k = 3.*\
i = k--;      /*Assigns i with k, then
decrements k therefore i = 3 and k = 2. *\
```

ਮਿਸਾਲ 3 :

```
/* 'program to explain the prefix increment
operator */
void main ( )
{
        int i, j ;
        i = 5;
        j = ++i + 5;      */ I Is incremented then
                          processed * /
printf ("Value of i is %d and j is %d \n", i, j) ;
}
```

Output: Value of i is 6 and j is 11

Explanation: j ਨੂੰ ++ 5 ਦੇ ਪਰਿਣਾਮ ਨਾਲ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ। i ਦਾ ਇਨਕਰੀਮੈਂਟ ਹੋਇਆ ਹੈ, ਪਰਿਣਾਮ 5 ਦੇ ਵਿਚ ਜਮ੍ਹਾਂ ਕਰਕੇ ਹੋਇਆ ਹੈ ਅਤੇ ਕੁਲ ਪਰਿਣਾਮ j ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ।

ਮਿਸਾਲ 4 :

```
/* Program to explain the postfix increment operator
*/
   void main ( )
   {
        int i, j ;
        i = 5;
        j = 5 + i++;   /* i is added to 5, assigned
                        to   j and i is incremented */
printf ("Value of i is %d and j is %d \n",i, j);
}
```

Output:: Value of i is 6 and j is 10

Explanation: j ਨੂੰ 5 + i++ ਦੇ ਪਰਿਣਾਮ ਨਾਲ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ ਪਹਿਲਾਂ j ਨੂੰ 5 ਵਿਚ ਜੋੜਿਆ ਗਿਆ ਹੈ, i ਵਿਚ ਇਨਕਰੀਮੈਂਟ ਹੋਇਆ ਹੈ ਅਤੇ ਸ਼ੁਧ ਪਰਿਣਾਮ j ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ।

ਮਿਸਾਲ 5 :

```
/* Program to explain the prefix decrement
operator */
void main ( )
{
int i, j;
i = 5;
j = --i + 5;    /* i is decremented & then processed*/
printf("Value of i is %d and j is %d \n", I, j);
}
```

Output: Value of i is 4 and j is 9

Explanation: J ਨੂੰ --i + 5 ਦੇ ਪਰਿਣਾਮ ਨਾਲ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ ਪਹਿਲਾਂ i ਨੂੰ ਡਿਕਰੀਮੈਂਟ ਕੀਤਾ ਗਿਆ ਹੈ, ਪਰਿਣਾਮ ਨੂੰ 5 ਵਿਚ ਜਮ੍ਹਾਂ ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਕੁਲ ਪਰਿਣਾਮ j ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ।

ਮਿਸਾਲ 6:

```
/*To explain postfix decrement operator */
void main()
{
 int i, j ;
i =5 ;    /* i is added to 5 ,assigned to j and
          then decremented */
j = 5 + i- - ;
printf("Value of i is %d and j is %d \n " ,i, j) ;
}
```

Output : Value if i is 4 and j is 10

Explanation: j ਨੂੰ 5 + i-- ਦੇ ਪਰਿਣਾਮ ਨਾਲ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ। ਪਹਿਲਾਂ i ਨੂੰ 5 ਵਿਚ ਜੋੜਿਆ ਗਿਆ ਹੈ, i ਨੂੰ ਡਿਕਰੀਮੈਂਟ ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਸ਼ੁਧ ਪਰਿਣਾਮ j ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਗਿਆ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆ ਇਨਕਰੀਮੈਂਟ ਓਪਰੇਟਰ (Problem increment operator):
x ਅਤੇ y ਦਾ ਕੀ ਮੁੱਲ ਹੋਵੇਗਾ, ਤਾਂ ਜੋ ਆਊਟਪੁੱਟ x=3 ਅਤੇ y=7 ਹੋਵੇ

```
# include <stdio.h>
void main()
 {
 int x =? , y =?;
 printf("x =%d\n" , x++) ;
 printf("y =%d\n" , ++y) ;
 }
```

ਸਮੱਸਿਆ ਡਿਕਰੀਮੈਂਟ ਓਪਰੇਟਰ (Problem decrement operator):
x ਅਤੇ y ਦਾ ਕੀ ਮੁੱਲ ਹੋਵੇਗਾ, ਤਾਂ ਜੋ ਆਊਟਪੁੱਟ x=5 ਅਤੇ y=8 ਹੋਵੇ

```
# include <stdio.h>
void main()
 {
 int x =? , y = ?;
 printf("x =%d\n" , - -x) ;
 printf("y =%d\n" , y- -) ;
 }
```

4.2.4 ਟਰਨਰੀ ਓਪਰੇਟਰ (Ternary operator)

ਇਹ ਸੀ (C) ਵਿਚ ਕੰਡੀਸ਼ਨਲ ਓਪਰੇਟਰ *(conditional operator)* ਹੈ। ਇਹ ਤਿੰਨ ਔਪਰੇਂਡਸ (Operands) ਲੈਂਦਾ ਹੈ। ਸੀ (C) ਵਿਚ (ਟਰਨਰੀ / **Ternary**) ਓਪਰੇਟਰ ਚਿੰਨ੍ਹ '?' ਵਜੋਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਨੂੰ ਹੇਠਾਂ ਦਰਸਾਈਆ ਗਿਆ ਹੈ :

<expression>? <value1>: <value2>;

ਜਿਥੇ : ਐਕਸਪ੍ਰੈਸ਼ਨ → ਰਿਲੇਸ਼ਨਲ ਓਪਰੇਟਰ (relational operator)
ਮੁੱਲ 1 → ਇਹ ਉਦੋਂ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਮੁੱਲ ਸਹੀ ਹੋਵੇ
ਮੁੱਲ 2 → ਇਹ ਉਦੋਂ ਅਸਾਈਨਡ (assigned) ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਐਕਸਪ੍ਰੈਸ਼ਨ ਦਾ ਪਰਿਣਾਮ (result) ਗਲਤ ਹੋਵੇ।

ਮਿਸਾਲ : **x=y<z?y:z**

ਇਥੇ x ਨੂੰ y ਦਾ ਮੁੱਲ ਅਸਾਈਨਡ (assigned) ਕਰੇਗਾ ਬਸ਼ਰਤੇ y ਜੇ z ਨਾਲੋਂ ਘੱਟ ਹੋਵੇ, ਵਰਨਾ ਇਹ z ਦੇ ਮੁੱਲ ਨੂੰ ਅਸਾਈਨਡ (assigned) ਕਰੇਗਾ।

4.2.5 ਕੌਮਾ ਓਪਰੇਟਰ (Comma operator)

ਇਹ ਦੋ ਜਾਂ ਵਧੇਰੇ ਸੰਬੰਧਿਤ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਨੂੰ ਇਕਠਿਆਂ ਜੋੜਨ ਲਈ ਵਰਤੀਆ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ : **n = (i = 5,j = 3,++j);**

ਪਹਿਲਾਂ i ਨੂੰ 5 ਮੁੱਲ ਫਿਰ j ਨੂੰ 3 ਅਤੇ ਅੰਤ ਵਿਚ n ਨੂੰ 4 ਅਸਾਈਨਡ (assigned) ਕਰੋ।
(First assigns the value 5 to i, then assign 3 to j, and finally assign 4 to n)

ਮਿਸਾਲ :

```
# include <stdio.h>
void main ( )
{
 int a =4, b =5, result1,result2;
result1 = a > b? a : b ;
printf("The result1 = %d\n" ,result1);
result2 = a < b? a : b ;
printf("The result2 = %d\n" ,result2);
}
```

Output :

```
The result 1 = 5
The result 2 = 4
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

4.2.6 ਸਾਈਜਆਫ( ) ਓਪਰੇਟਰ [sizeof( ) of operator]

ਇਹ ਓਪਰੇਟਰ, ਓਪਰੈਂਡ (operand) ਦੇ ਸਾਇਜ (ਮਿਸਾਲ ਲਈ ਬਾਈਟਸ ਦੀ ਗਿਣਤੀ) ਨੂੰ ਵਾਪਿਸ ਕਰਦਾ ਹੈ। ਇਹ ਲੋਅਰ-ਕੇਸ (lower case) ਵਿਚ ਲਿਖਿਆ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਇਹ ਆਪਣੇ ਓਪਰੈਂਡ (operand) ਤੋਂ ਪਹਿਲਾਂ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਇਹ ਕਾਂਸਟੈਂਟ (constant) ਵੇਰੀਏਬਲ (variable) ਜਾਂ ਡਾਟਾ ਟਾਈਪ (data type) ਹੋ ਸਕਦਾ ਹੈ। ਇਸਨੂੰ ਨਿਮਨ ਵਜੋ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ

Sizeof(operand);

4.2.7 ਬਿਟਵਾਈਜ ਓਪਰੇਟਰ (Bitwise operator): ਸਮੁੱਚੀ ਡਾਟਾ ਮੱਦ (item) ਮੇਮਰੀ ਵਿਚ ਬਿਟਸ (0's ਜਾਂ 1's) ਦੇ ਸਾਰ ਵਜੋਂ ਸਟੋਰ ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਅਤੇ ਕੁਝ ਪ੍ਰਯੋਗਾਂ ਵਿਚ ਇਨ੍ਹਾਂ ਬਿਟਾਂ ਦੀ ਵਿਉਂਤਬੰਦੀ ਦੀ ਜਰੂਰਤ ਹੁੰਦੀ ਹੈ। ਬਿਟਵਾਈਜ ਓਪਰੇਸ਼ਨ ਕਰਨ ਲਈ, ਸੀ (C) ਵਲੋਂ 6 ਓਪਰੇਟਰ ਪ੍ਰਦਾਨ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਓਪਰੇਟਰ int ਜਾਂ char ਟਾਈਪ ਦੇ ਡਾਟਾ ਨਾਲ ਕੰਮ ਕਰਦੇ ਹਨ।

ਮਿਸਾਲ :

```
# include <stdio.h>
void main ( )
{
 int x ;
 float y ;
 char ch = 'y' ;
 x = 10;
 y = 100.00;
 printf ("Size of x = %d\n",sizeof(x));
 printf ("Size of y = %d\n",sizeof(y));
 printf ("Size of ch = %d\n",sizeof(ch));
 printf ("Size of double = %d\n",sizeof(double));
}
```

Output:

```
Size of x  =2
Size of y = 4
Size of ch =1
Size of double =8
```

o

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਓਪਰੇਟਰਸ ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ (Operators and Expression)

ਸੀ (C) ਓਪਰੇਟਰ ਬੁਨਿਆਦੀ ਤੌਰ ਤੇ ਉਨਰੀ *(unary)*, ਬਾਈਨਰੀ *(binary)* ਅਤੇ ਟਰਨਰੀ *(ternary)* ਔਂਪਰੇਸ਼ਨਾਂ ਵਿਚ ਸ਼੍ਰੇਣੀਬੱਧ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਉਨਰੀ *(unary)* ਵਿਚ ਪੰਜ ਓਪਰੇਟਰ ਹੁੰਦੇ ਹਨ, ਜਿਸ ਵਿਚ ਚਾਰ ਦੋਹਰੇਪਨ ਅਤੇ ਇਕ ਟਰਨਰੀ *(ternary)* ਓਪਰੇਟਰ ਜਿਥੇ ਉਨਰੀ *(unary)* !,++, - ਅਤੇ - - ਬਾਈਨਰੀ *(binary)* ਓਪਰੇਟਰ ਹਨ। +,-,*/ ਬੁਨਿਆਦੀ ਅਰਥਮੇਟਿਕ ਓਪਰੇਟਰ ਹਨ। ਮਾਡਯੂਲ ਓਪਰੇਟਰ ਇਕ ਹੋਰ ਅਰਥਮੇਟਿਕ ਓਪਰੇਟਰ ਹੈ ਜਿਸਦਾ ਪਰਿਣਾਮ ਪੂਰਨ-ਅੰਕ ਵੰਡ ਦੇ ਬਾਅਦ ਰਿਮੇਂਡਰ *(remainder)* ਵਿਚ ਹੁੰਦਾ ਹੈ।

<, <=, >, =>, = =, != ਸੰਪਰਕੀ ਓਪਰੇਟਰ ਹਨ, ਇਹ ਕੌਨਸਟੈਂਟ ਅਤੇ ਵੇਰੀਏਬਲਜ਼ ਦੇ ਦਰਮਿਆਨ ਸੰਪਰਕ ਨੂੰ ਪਰਿਭਾਸ਼ਤ ਕਰਦੇ ਹਨ।

C ਭਾਸ਼ਾ ਐਸਾਈਨਡ ਸਟੇਟਮੈਂਟ ਦੀ ਕਿਸਮ ਦੀ ਸਹਾਇਤਾ ਕਰਦੀ ਹੈ। ਓਪਰੇਟਰਾਂ ਦੀ ਉਤਮਤਾ ਹੁੰਦੀ ਹੈ ਜਿਹੜੇ ਏਕਸਪ੍ਰੇਸ਼ਨ ਦਾ ਮੁੱਲਾਂਕਣ ਕਰਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1: ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ (Fill in the blanks)

1.1 Exponentiation ਦਾ ________ ਓਪਰੇਟਰ ਹੁੰਦਾ ਹੈ ?
1.2 _______ ਓਪਰੇਟਰ ਦਾ ਇਸਤੇਮਾਲ ਵੱਖ-ਵੱਖ ਕੰਡੀਸ਼ਨਸ (conditions) ਨੂੰ ਜੋੜਨ ਲਈ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ?
1.3 int i=5 ਨੂੰ _____________ ਅਸਾਇਨਮੈਂਟ (assignment) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ?
1.4 i=5 ਨੂੰ _____________ ਅਸਾਇਨਮੈਂਟ (assignment) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ?
1.5 _________________ ਓਪਰੇਟਰ ਦੀ ਦਰਜਾਬੰਦੀ ਨਹੀਂ ਹੁੰਦੀ ?

ਪ੍ਰਸ਼ਨ 2: ਦੱਸੋ ਸਹੀ ਹੈ ਜਾਂ ਗਲਤ ?

1. ਕੀ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ++(a+b) ਉਚਿਤ ਹੈ, (ਜਿਥੇ a=y, b=5 ਹੈ) ?
2. ਕੀ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) a&&=b ਉਚਿਤ ਹੈ, (ਜੇ ਕਰ a=y, b=5 ਹੈ) ?
3. ਕੀ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) - - 10 ਉਚਿਤ ਹੈ ?
4. ਕੀ ਇਹ ਅਸਾਇਨਮੈਂਟ (assignment) ਸਟੇਟਮੈਂਟ "a+b=c;" C ਭਾਸ਼ਾ ਵਿਚ ਸਹੀ ਹੈ ?
5. ਕੀ C ਭਾਸ਼ਾ ਵਿਚ ਆਇਡੈਂਟਿਫਾਇਰ (identifier) ਦਾ ਪਹਿਲਾ ਅੱਖਰ __ ਹੋ ਸਕਦਾ ਹੈ ?

ਪ੍ਰਸ਼ਨ 3: ਸੰਖੇਪ ਉੱਤਰ ਪ੍ਰਕਾਰ ਦੇ ਪ੍ਰਸ਼ਨ

1. Modules ਓਪਰੇਟਰ ਕਿੱਥੇ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ?
2. ਲੌਜਿਕਲ (logical) ਅਤੇ relational ਓਪਰੇਟਰ ਵਿਚ ਅੰਤਰ ਪਤਾ ਕਰੋ ?
3. Unary ਓਪਰੇਟਰ ਕੀ ਹੈ ?
4. Increment ਅਤੇ decrement ਓਪਰੇਟਰ ਦਾ ਇਕ-ਇਕ ਉਦਾਹਰਣ ਦਿਉ?
5. ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਵਿਚ C ਦਾ ਇਸਤੇਮਾਲ ਕੀ ਹੁੰਦਾ ਹੈ ?
6. ਅਸਾਇਨਮੈਂਟ (assignment) ਓਪਰੇਟਰ ਦੀਆਂ ਵੱਖ-ਵੱਖ ਕਿਸਮਾਂ ਲਿਖੋ ?
7. ਓਪਰੈਂਡ (operand) ਕੀ ਹੁੰਦਾ ਹੈ ?

ਪ੍ਰਸ਼ਨ 4: ਨਿਮਨਲਿਖਤ ਪ੍ਰੋਗਰਾਮ ਦੇ ਲਾਗੂ ਕਰਨ ਦੇ ਬਾਅਦ ਕੀ ਪਰਿਣਾਮ ਹੁੰਦਾ ਹੈ?

```
a=7;
b=++a+5;
c=b--+10
printf("%d, %d, %d\n" a, b, c);
```

Answer to objective questions

ਪ੍ਰਸ਼ਨ	.1	.2	.3	.4	.5
1	**	Relational	Declaration	General	Relational
2	F	F	T	F	T

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ **5**

ਕੰਟਰੋਲ ਫਲੋ (ਭਾਗ - 1)
Control Flow (Part - I)

5.0.0 ਭੂਮਿਕਾ (Introduction):

ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮ ਹਿਦਾਇਤਾਂ ਦਾ ਇਕ ਸੈੱਟ ਹੁੰਦਾ ਹੈ ਜੋ ਕੰਪਿਊਟਰ ਨੂੰ ਵਿਸ਼ੇਸ਼ ਕੰਮ ਕਰਨ ਲਈ ਦਿਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਹਿਦਾਇਤਾਂ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਲਿਖੀ ਤਰਤੀਬ ਅਨੁਸਾਰ ਇਕ ਤੋਂ ਬਾਅਦ ਇਕ ਕਰਕੇ ਲਾਗੂ ਹੁੰਦੀਆਂ ਹਨ। ਸਟੇਟਮੈਂਟ (statement) ਲਾਗੂ ਕਰਨ ਦਾ ਇਹ ਕ੍ਰਮ ਸਧਾਰਣ ਸਮੱਸਿਆਵਾਂ ਲਈ ਚੰਗਾ ਕੰਮ ਕਰਦਾ ਹੈ, ਜਿਥੇ ਫੈਸਲਾ ਕਰਨ ਦੀ ਕੋਈ ਪ੍ਰਕ੍ਰਿਆ ਸ਼ਾਮਲ ਨਹੀ ਹੁੰਦੀ। ਪਰੰਤੂ ਅਮਲ ਵਿਵਹਾਰ ਵਿਚ, ਇਹਨਾਂ ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਲਾਗੂ ਕਰਨ ਦੇ ਕ੍ਰਮ ਨੂੰ ਬਦਲਣ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ ਜਾ ਸਟੇਟਮੇਂਟਸ (statements) ਦੇ ਸਮੂਹ ਨੂੰ ਦੋਹਰਾਇਆ ਜਾਂਦਾ ਹੈ ਤੇ ਕਈ ਵਾਰ ਅਜਿਹਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਦ ਤਕ ਵਿਸ਼ੇਸ਼ ਹਾਲਾਤਾਂ ਦੀ ਪੂਰਤੀ ਨਹੀ ਹੋ ਜਾਂਦੀ। ਅਜਿਹੇ cases ਵਿਚ ਜਿਥੇ ਇਨ੍ਹਾਂ ਸਟੇਟਮੈਂਟਸ (statements) ਵਿਚ ਕ੍ਰਮ, ਜੋ ਲਾਗੂ ਕੀਤੇ ਗਏ ਹਨ ਤੇ ਕੰਟਰੋਲ ਪਾਉਣਾ ਪੈਂਦਾ ਹੈ। ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਲਾਗੂ ਕਰਨ ਦੀ ਤਰਤੀਬ ਨੂੰ ਬਦਲਣਾ, ਕੰਟਰੋਲ-ਫਲੋ (contro flow) ਨੂੰ ਬਦਲਣਾ ਹੁੰਦਾ ਹੈ। ਇਨ੍ਹਾਂ ਵਿਚ ਨਿਮਨਲਿਖਤ ਟੋਈ ਇਕ ਸਥਿਤੀ ਸ਼ਾਮਲ ਹੁੰਦੀ ਹੈ।

- **ਸ਼ਾਖਾਵਾਂ (Branching)** ਦਾ ਭਾਵ ਹੈ ਹਿਦਾਇਤਾਂ ਦਾ ਇਕ ਸਮੂਹ ਲਾਗੂ ਕਰਨਾ ਜੋ ਇਕ ਫੈਸਲੇ ਦੇ ਨਤੀਜੇ ਤੇ ਨਿਰਭਰ ਕਰਦਾ ਹੈ।

  ਕੰਡੀਸ਼ਨਲ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਜਾਂ ਫੈਸਲਾ ਕਰਨ ਵਾਲੇ ਸਟੇਟਮੈਂਟ (Decision making statements or conditional control statements): ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਹ ਸਿਲਸਿਲੇਵਾਰ (ਅਰਥਾਤ ਕ੍ਰਮ ਵਾਲੀਆਂ) ਸਟਰਕਚਰ (structure) ਦੇ ਨਾਲ ਦਿਖਾਈ ਦਿੰਦੇ ਹਨ। ਇਹ ਦਿਤੀ ਗਈ ਸਥਿਤੀ ਦੀ ਵੈਧਤਾ (validity) ਦੀ ਜਾਂਚ ਕਰਨ ਦੇ ਲਈ ਜਿੰਮੇਵਾਰ ਹੁੰਦੇ ਹਨ ਅਤੇ ਵਿਸ਼ੇਸ਼ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸਟੇਟਮੈਂਟ (statement) ਲਾਗੂ ਕਰਦੇ ਹਨ ਬਸ਼ਰਤੇ ਕਿ ਕੰਡੀਸ਼ਨ ਸਹੀ (condition true) ਹੋਵੇ ਅਤੇ ਕੁਝ ਵਿਸ਼ੇਸ਼ ਸਟੇਟਮੈਂਟਸ (statements) ਨੂੰ ਛਡਦੀ ਹੋਵੇ, ਜੋ ਗਲਤ ਹਨ। ਉਹ ਹਨ :-

• if ਸਟੇਟਮੈਂਟ	if statement
• if - else ਸਟੇਟਮੈਂਟ	if – else statement
• else if ਸਟੇਟਮੈਂਟ	else if statement
• ਸਵਿਚ ਸਟੇਟਮੈਂਟ	switch statement

- **ਲੂਪਿੰਗ (Looping)** ਦਾ ਭਾਵ ਹੈ ਹਿਦਾਇਤਾਂ ਦੇ ਇਕ ਸਮੂਹੀ ਨੂੰ ਮੁੜ ਮੁੜ ਲਾਗੂ ਕਰਨਾ ਜਾਂ ਕੁਝ ਦਿਤੀ ਗਈ ਗਿਣਤੀ ਲਈ ਹੋਣ ਜਦ ਤਕ ਕਿ ਲੋੜੀਂਦੇ ਹਾਲਾਤਾਂ ਦੀ ਪੂਰਤੀ ਨਹੀਂ ਹੁੰਦੀ। ਲੂਪਿੰਗ ਸਟੇਟਮੈਂਟ (looping statement) ਇਹ ਉਸ ਕੰਡੀਸ਼ਨਲ ਸਟੇਟਮੈਂਟ (conditional staement) ਦੀ ਤਰ੍ਹਾਂ ਹੁੰਦਾ ਹੈ ਜੋ ਕਿਸੇ ਵਿਸ਼ੇਸ਼ ਹਾਲਤ ਦਾ ਮੁਲਾਂਕਣ ਕਰਦਾ ਹੈ। ਜੇਕਰ ਸ਼ਰਤ (condition) ਦਾ ਜਾਂ ਸਥਿਤੀ ਦਾ ਸਹੀ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਦਿਤੇ ਗਏ ਪ੍ਰੋਗਰਾਮ ਸਟੇਟਮੈਂਟ (statement) ਮੁੜ-ਮੁੜ ਲਾਗੂ ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਜਦ ਤਕ ਕਿ ਸ਼ਰਤ/condition (ਜਿਹੜੀ ਹਰੇਕ ਲੂਪ ਦੇ ਦੋਰਾਨ ਪੁਨਰ ਮੁਲਾਂਕਣ ਕੀਤੀ ਜਾਂਦੀ ਹੈ) ਦਾ ਮੁਲਾਂਕਣ ਕਰਕੇ ਉਸਨੂੰ ਗਲਤ ਨਹੀ ਕੀਤਾ ਜਾਂਦਾ। ਉਹ ਹਨ :-

• for ਸਟੇਟਮੈਂਟ	for statement
• while ਸਟੇਟਮੈਂਟ	while statement
• Do-while ਸਟੇਟਮੈਂਟ	do-while statement

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਜੰਪਿੰਗ (Jumping) ਤੋਂ ਭਾਵ ਇਕ ਪੁਆਇੰਟ ਤੋਂ ਦੂਜੇ ਪੁਆਇੰਟ ਉਤੇ ਜਾਂ ਉਪ ਯੂਨਿਟ ਉਤੇ ਪ੍ਰੋਗਰਾਮ ਵਿਚ, ਕੰਟਰੋਲ ਦੀ ਤਬਦੀਲੀ ਕਰਨੀ ਹੈ ਇਸ ਵਿਚ ਸ਼ਾਮਲ ਹਨ :

• **break ਸਟੇਟਮੈਂਟ**	**break statement**
• **continue ਸਟੇਟਮੈਂਟ**	**continue statement**
• **go to ਸਟੇਟਮੈਂਟ**	**go to statement**

ਸੀ (C) ਭਾਸ਼ਾ ਵਲੋਂ ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਲਾਗੂ ਕਰਨ ਦੇ ਕ੍ਰਮ ਤੇ ਕੰਟਰੋਲ (control) ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਨ੍ਹਾਂ ਨੂੰ ਫਲੋ-ਕੰਟਰੋਲ (flow control) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਸੀ (C) ਵਲੋਂ ਨਿਮਨਲਿਖਿਤ ਸਟੇਟਮੈਂਟਸ (statements) ਦਾ ਸੈੱਟ (set) ਪ੍ਰਦਾਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

5.1.0 ਡਿਸਿਜ਼ਨ ਮੇਕਿੰਗ ਸਟੇਟਮੈਂਟਸ (Decision making statements)

5.1.1 if ਸਟੇਟਮੈਂਟ (if statement): if statement ਦਾ ਇਸਤੇਮਾਲ ਇਕ ਸਥਿਤੀ ਨੂੰ ਸਟੇਟਮੈਂਟ ਦੇ ਲਾਗੂ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਜਾਂਚਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਸਦੀਆਂ ਦੋ ਰੂਪ ਰੇਖਾਵਾਂ ਹੁੰਦੀਆਂ ਹਨ :-

if ਐਕਸਪ੍ਰੇਸ਼ਨ
if(expression)
 	
statement;
 	
ਸਟੇਟਮੈਂਟ (statement), ਸਟੇਟਮੈਂਟਸ ਦਾ ਖੰਡ (block of statements) ਵੀ ਹੋ ਸਕਦਾ ਹੈ
	if ਐਕਸਪ੍ਰੇਸ਼ਨ [if (expression)]:
	{
	statement1;
	statement2;
	statement3;
	}
	
	ਜੇ ਪਰਿਣਾਮ ਸਹੀ ਹੈ, ਐਕਸਪ੍ਰੇਸ਼ਨ (expression), ਮੁਲਯਾਂਕਣ (evaluated) ਹੁੰਦਾ ਹੈ ਫਿਰ ਅਗਲਾ ਸਟੇਟਮੈਂਟ (statement) ਲਾਗੂ ਹੋ ਜਾਂਦਾ ਹੈ ਨਹੀ ਤੇ ਜੰਪ (skip) ਕਰ ਜਾਂਦਾ ਹੈ
		ਮਿਸਾਲ :
		int a,b, c;
		a = 10,b = 5,c = 10 ;
		if (a>b)
		c= c+5;
		printf("%d",c);
		
		Output : 15

ਇਥੇ ਜੇਕਰ **a** ਦਾ ਮੁੱਲ ਸ਼ੁਰੂ ਹੋਣ ਵੇਲੇ 5 ਤੋਂ ਘੱਟ ਹੁੰਦਾ ਹੈ, ਮੰਨ ਲਓ ਕਿ 4, ਤਦ ਪ੍ਰਿੰਟਿਡ ਮੁੱਲ 10 ਹੁੰਦਾ ਹੈ ਕਿਉਂ ਜੋ ਐਕਸਪ੍ਰੇਸ਼ਨ/expression (a>b) ਦੀ ਕੀਤੀ ਮੁਲਾਂਕਣ ਗਲਤ ਹੋਏਗੀ ਇਸ ਲਈ ਸਟੇਟਮੈਂਟ C=C+5 ਬਾਈ-ਪਾਸ (Bypass) ਹੋ ਜਾਂਦੀ ਹੈ।

5.1.2 if–else: ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) if ਸਟੇਟਮੈਂਟ (statement) ਵਿਚ ਪਹਿਲਾਂ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਜੇਕਰ ਇਹ ਠੀਕ ਹੁੰਦਾ ਹੈ ਤਾਂ ਫਿਰ ਸਟੇਟਮੈਂਟ-1 (statement-1) ਲਾਗੂ ਹੁੰਦਾ ਹੈ ਨਹੀ ਤੇ (ਅਰਥਾਤ ਜੇਕਰ ਇਹ ਗਲਤ ਹੁੰਦਾ ਹੈ) ਸਟੇਟਮੈਂਟ-2 ਲਾਗੂ ਹੁੰਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

if ਐਕਸਪ੍ਰੈਸ਼ਨ (if expression):

```
{
 statement1;
else
 statement2;
}
```

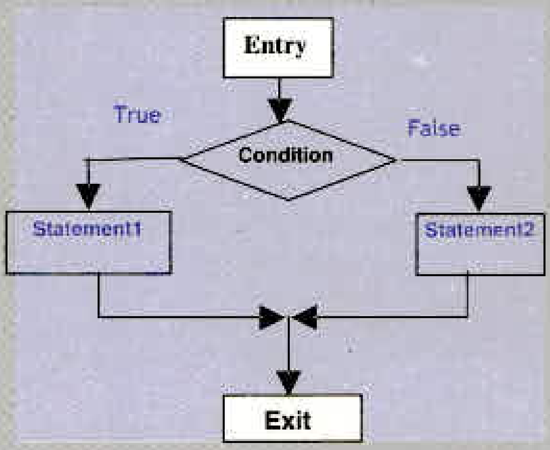
ਮਿਸਾਲ 1 :

```
int a,b,c;
a =10;
b = 5 ;
if (a>b)
    c = 5 ;
else
    c = 7;
printf("Value of c = %d\n",c);
```

Output:

Value of c = 5

however , if the initial value of a is 4 (less than the value of b) then the result printed will be 7



ਮਿਸਾਲ 2 : ਇਹ ਪ੍ਰੋਗਰਾਮ ਇਕ ਅੰਕ ਨੂੰ ਸਵੀਕਾਰ ਕਰਦਾ ਹੈ ਤੇ ਜੇਕਰ ਨੰਬਰ **odd** ਹੈ ਤਾਂ ਉਸ ਨੂੰ ਛਾਪਦਾ ਹੈ ਨਹੀਂ ਤਾਂ ਪ੍ਰੋਗਰਾਮ ਤੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ

(This program accepts a number and prints if it is an odd number other- wise exit)

```
# include <stdio.h>
void main()
{
 int numb;
 printf("Enter the number\n ");
 scanf("%d" , &numb);
 if ((numb %2) ! = 0)
 printf("%d ,is an odd numbers\n" ,numb);
}                                        /* end of main() *\
```

Output:

Enter the number
13
13, is an odd number

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ 3: ਤਿੰਨ ਅੰਕਾਂ ਵਿਚੋ ਸਭ ਤੋ ਵੱਡਾ ਅੰਕ ਲੱਭਣਾ।
(To find the largest of three numbers)

```
void main()
{
  float a,b,c ;
  printf("Enter the three numbers -> ");
  scanf("%f %f %f ", &a,&b,&c);
  printf("\n\n The largest among % 7.3f , % 7.3f , % 7.3f is " , a,b,c,);
  If (a>b && a>c)
     printf("% 7.3f\n" , a);
  else if ((b>a && b>c)
            printf("% 7.3f\n" , b);
         else
               printf("% 7.3f\n" , c);
}
```

Output:

```
Enter the three numbers -> 23 76 45
The biggest among 23.000 , 76.000 , 45.000 is 76.000
```

ਮਿਸਾਲ 4: ਇਹ ਪ੍ਰੋਗਰਾਮ ਦੋ ਫਲੋਟਿੰਗ (floating) ਪੁਆਂਇੰਟ ਅੰਕ ਪ੍ਰਵਾਨ ਕਰਦਾ ਹੈ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਅਨੁਪਾਤ ਦੀ ਗਣਨਾ ਕਰਦਾ ਹੈ। ਜੇਕਰ ਅਨੁਪਾਤ ਜ਼ੀਰੋ ਤੋ ਵੱਡਾ ਹੁੰਦਾ ਹੈ ਤਾਂ ਇਹ ਇਨ੍ਹਾਂ ਅੰਕਾਂ ਦੀ ਵਿਸ਼ਾ ਸੂਚੀ ਨੂੰ ਆਪਸ ਵਿਚ ਬਦਲ ਦਿੰਦਾ ਹੈ।

```
# include <stdio.h>
void main()
 {
  float x,y,ratio,temp;
  printf("Enter the X and Y \n");
  scanf("%f %f" , &x, &y);
  ratio = x / y;
  printf("Ratio = %f\n",ratio);
  If (ratio >0)
     {
      temp = x;                          /* exchange variables*/
      x = y;
      y = temp ;
     }
      printf("X =%f ,Y = %f\n" , x,y);
 }
```

Output: Enter X and Y

```
34.0
55.0
Ratio = 0.6181818
X = 55.000000 , Y = 34.000000
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ 5

nested if ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ ਜਿਵੇਂ ਮਿਸਾਲ 2 ਵਿਚ ਪੂਰਵ ਪੰਨੇ ਤੋਂ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਮੁਕੰਮਲ ਕਰਨ ਲਈ ਬਸ਼ਰਤੇ ਕਿ ਇਹ ਵਿਦਿਆਰਥੀ ਵਲੋਂ ਵੱਖ ਵੱਖ ਵਿਸ਼ਿਆਂ ਵਿਚ ਪ੍ਰਾਪਤ ਕੀਤੇ ਗਏ ਅੰਕਾਂ ਨੂੰ ਕੀ ਬੋਰਡ ਰਾਹੀ ਇਨਪੁਟ ਕੀਤਾ ਹੋਏ। ਵਿਦਿਆਰਥੀ ਨਿਮਨਲਿਖਿਤ ਨਿਯਮਾਂ ਦੇ ਅਨੁਸਾਰ ਇਕ ਗ੍ਰੇਡ ਪ੍ਰਾਪਤ ਕਰਨਗੇ।

Marks 75 or above	**- grade = A**
above 60 but below 75	**- grade = B**
above 50 but below 60	**- grade = C**
less than 50	**- grade = D**

Hint

```
if (marks > =75)
   grade =”A” ;
  else
   if (marks > =60)
     grade =”B” ;
       if (marks > =50)
         grade =”C” ;
       else
        printf(“grade = D”);
```

5.2.0 ਸਵਿਚ ਸਟੇਟਮੈਂਟ (switch statement)

ਅਸੀ ਦੇਖਿਆ ਹੈ ਕਿ if ਸਟੇਟਮੈਂਟ (statement) ਇਕ ਜਾਂ ਦੋ ਸਟੇਟਮੈਂਟਸ (statements) ਵਿਚੋਂ ਚੁਣਿਆ ਜਾਂਦਾ ਹੈ ਜਦ ਕਿ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਦੇ ਮੁੱਲ ਤੇ, ਸਵਿਚ-ਸਟੇਟਮੈਂਟ (switch statement) ਵਿਚ ਅਨੇਕਾਂ ਸਟੇਟਮੈਂਟਸ (statements) ਵਿਚੋਂ ਇਕ ਨੂੰ ਚੁਣਿਆ ਜਾਂਦਾ ਹੈ ਜੋ ਕਿ ਪੂਰਨ ਅੰਕ ਦੇ ਮੁੱਲ (int, byte, short or long) ਤੇ ਜਾਂ ਕੈਰ ਐਕਸਪ੍ਰੈਸ਼ਨ (char expression) ਦੇ ਮੁੱਲ ਤੇ ਆਧਾਰਿਤ ਹੁੰਦਾ ਹੈ। ਕੀ-ਵਰਡ, “case” ਦਾ ਅਨੁਸਰਣ ਪੂਰਣ ਅੰਕ ਜਾਂ ਚਿੰਨ੍ਹ-ਕਾਂਸਟੈਂਟ (character constant) ਦੁਆਰਾ ਹੁੰਦਾ ਹੈ। ਹਰ ਇਕ “case” ਮਾਮਲੇ ਵਿਚ ਇਕ ਕਾਂਸਟੈਂਟ (constant) ਦੂਜਿਆਂ ਤੋਂ ਭਿੰਨ ਹੁੰਦਾ ਹੈ।

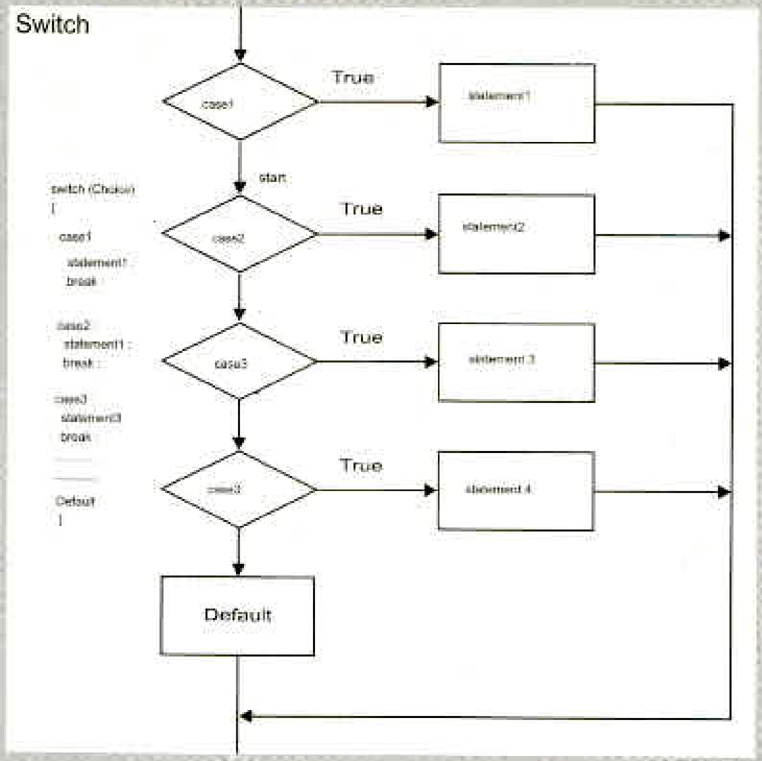
Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਲੂਪ (loop) ਓਪਰੇਸ਼ਨ ਦੇ ਦੌਰਾਨ ਕੁਝ ਹਾਲਤਾਂ ਵਿਚ ਪ੍ਰੋਗਰਾਮ ਲੂਪ ਦਾ ਕੁਝ ਭਾਗ ਛੱਡਣਾ ਜਰੂਰੀ ਹੋ ਜਾਂਦਾ ਹੈ ਅਤੇ ਉਸ ਦੇ ਇਲਾਵਾ ਅਗਲਾ ਭਾਗ ਚਲਾਉਣਾ ਵੀ ਜਰੂਰੀ ਹੁੰਦਾ ਹੈ. ਇਸ ਪ੍ਰਕ੍ਰਿਆ ਨੂੰ ਕਰਨ ਲਈ ਸੀ (C) ਵਿਚ continue statement ਵਰਤੀ ਜਾਂਦੀ ਹੈ

The syntax is:

```
switch (expression) {
   case value1 :
      statements ;
      break ;
   case value2 :
      statements ;
      break ;
   ...(more cases)...
   default :
      statements ;
     break ;
 }
```



5.3.0 ਬ੍ਰੇਕ ਸਟੇਟਮੈਂਟ (Break Statement)

ਕਈ ਵਾਰ ਇਕ ਲੂਪ (loop) ਨੂੰ ਲਾਗੂ ਕਰਦੇ ਸਮੇਂ ਇਹ ਜ਼ਰੂਰੀ ਹੋ ਜਾਂਦਾ ਹੈ ਕਿ ਲੂਪ (loop) ਦਾ ਕੁਝ ਭਾਗ ਛੱਡਿਆ (skip) ਜਾਵੇ ਅਤੇ ਲੂਪ ਨੂੰ ਜਿੰਨੀ ਜਲਦੀ ਹੋ ਸਕੇ ਕੁਝ ਵਿਸ਼ੇਸ਼ ਸ਼ਰਤਾਂ ਵਾਪਰਣ ਤੇ ਹਟਾਇਆ ਦੇਣਾ ਚਾਹੀਦਾ ਹੈ।

ਮਿਸਾਲ ਲਈ 100 ਅੰਕਾਂ ਦੇ ਸੈੱਟ ਵਿਚ ਇਕ ਵਿਸ਼ੇਸ਼ ਅੰਕ ਦੀ ਖੋਜ ਦਾ ਵਿਚਾਰ ਕਰੋ ਜਿਵੇਂ ਹੀ ਖੋਜ ਅੰਕ ਲੱਭ ਜਾਂਦਾ ਹੈ ਤਾਂ ਲੂਪ (loop) ਨੂੰ ਸਮਾਪਤ ਕਰਨਾ ਜਰੂਰੀ ਹੋ ਜਾਂਦਾ ਹੈ। ਸੀ (C) ਭਾਸ਼ਾ ਵਲੋਂ ਇਕ ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਦੂਜੇ ਲੂਪ (loop) ਦੇ ਵਿਚ ਅਤੇ ਉਸ ਤੋਂ ਬਾਹਰ ਜਾਣ ਦੀ ਆਗਿਆ ਦਿੱਤੀ ਜਾਂਦੀ ਹੈ। ਬ੍ਰੇਕ ਸਟੇਟਮੈਂਟ (break statement) ਸਾਨੂੰ ਇਸ ਕੰਮ ਨੂੰ ਪੂਰਨ ਕਰਦੀ ਹੈ। ਬ੍ਰੇਕ ਸਟੇਟਮੈਂਟ (break statement), for, while, do ਅਤੇ switch ਤੋਂ ਜਲਦੀ ਬਾਹਰ ਨਿਕਾਲ ਦਿੰਦੀ ਹੈ। ਇਕ ਬ੍ਰੇਕ (break), ਅੰਦਰੂਨੀ ਲੂਪ (inner loop) ਤੋਂ ਬਾਹਰ ਨਿਕਲਣ ਦਾ ਕਾਰਨ ਬਣਦੀ ਹੈ ਜਾਂ ਤੁਰੰਤ ਹੀ ਪ੍ਰੋਗਰਾਮ ਤੋਂ ਬਾਹਰ ਨਿਕਲ ਜਾਂਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ 6: ਪ੍ਰੋਗਰਾਮ ਬ੍ਰੇਕ ਸਟੇਟਮੈਂਟ (break statement) ਦੀ ਵਰਤੋਂ :

complete this program to find the average

```
/* A program to find the sum of the marks*/
#include < stdio.h >
void main()
{
int a, num=0;
float sum=0,average;
printf("Input the marks, -1 to end\n");
while(1)                                  /*While loop starts*/
  {
   scanf("%d",&a);
   if(a == -1)
   break;
   sum = sum+a;
   num++;
  }
}
```

5.4.0 ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ (Continue statement)

ਲੂਪ ਕੰਮਾਂ ਦੇ ਦੌਰਾਨ ਕੁਝ ਵਿਸ਼ੇਸ਼ ਹਾਲਾਤਾਂ ਦੇ ਅਧੀਨ ਲੂਪ ਦੀ ਬਾਡੀ ਦਾ ਇਕ ਭਾਗ ਤੋਂ ਅਲੱਗ ਹੋਣਾ ਜਰੂਰੀ ਹੁੰਦਾ ਹੈ। ਬ੍ਰੇਕ ਸਟੇਟਮੈਂਟ (break statement) ਦੀ ਤਰ੍ਹਾਂ, ਸੀ (c) ਵੱਲੋਂ ਅਜਿਹੇ ਹੀ ਸਟੇਟਮੈਂਟ (statement) ਦਾ ਸਮਰਥਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਸਨੂੰ ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ (Continue statement) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ (Continue statement) ਲੂਪ ਦੇ ਅਗਲੇ ਦੁਹਰਾਉ ਦੇ ਨਾਲ ਜਾਰੀ ਰਹਿਣ ਦਾ ਕਾਰਣ ਬਣਦਾ ਹੈ, ਵਿਚੋਂ ਕਿਸੇ ਵੀ ਸਟੇਟਮੈਂਟ (statement) ਤੋਂ ਅਲੱਗ ਹੋਣ ਤੋਂ ਬਾਅਦ। ਅਗਲੇ ਦੁਹਰਾਉ ਦੇ ਨਾਲ ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ (Continue statement) ਦੀ ਸਧਾਰਣ ਰੂਪ ਰੇਖਾ ਹੈ : ਨਿਰੰਤਰ (**Continue;**)

ਨਿਮਨਲਿਖਿਤ ਪ੍ਰੋਗਰਾਮ ਤੇ ਵਿਚਾਰ ਕਰੋ ਜੋ ਪੰਜਾਂ ਪੋਜੀਟਿਵ (positive) ਪੂਰਣ (integer) ਅੰਕਾਂ ਦਾ ਜੋੜ ਲਭਦਾ ਹੈ। ਜੇ ਕਰ ਨੈਗਿਟਿਵ (negative) ਨੰਬਰ ਪ੍ਰਵੇਸ਼ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਜੋੜ੍ਹ ਨਹੀਂ ਕੀਤਾ ਜਾਂਦਾ ਕਿਉਂ ਜੋ ਲੂਪ (loop) ਦਾ ਬਾਕੀ ਦਾ ਭਾਗ ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ (Continue statement) ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਲੋਪ ਹੋ ਜਾਂਦਾ ਹੈ।

```
#include < stdio.h >
void main()
{
  int a=1, num, sum=0;             /* declare and initialize the variables */
  for (a = 0; a < 5; a++)          /*for loop */
   {
     printf("Enter the integer");  /*Message to the user*/
     scanf("%d", &num);            /*read and store the number */
     if(num < 0)                   /*check whether the number is less than zero */
      {
       printf("You have entered a negative number"); // message to the user
       continue;                   /* starts with the beginning of the loop*/
      }                            /*end of for loop */
   sum+=num;                       /*add and store sum to num */
  }
     printf("The sum of positive numbers entered = %d",sum);    // print the sum.
}
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ 7: ਸਪਤਾਹ (ਸਵਿਚ ਸਟੇਟਮੈਂਟ ਦੀ ਵਰਤੋ ਨਾਲ) ਦੇ ਦਿਨ ਪ੍ਰਾਪਤ ਕਰਨ ਲਈ ਅੰਕ ਨੂੰ ਇਨਪੁਟ ਕਰੋ

```
#include <stdio.h>
#include <conio.h>
void main (void)
{
 int day;
 clrscr();
 printf(\"nEnter the day of the week as number:
");
 scanf("%d" ,&day);
 switch(day)
{
 case 0 :printf("\nDay of the week is Sunday\n");
break;
case 1 :printf("\nDay of the week is Monday\n");
break;
case 2 :printf("\nDay of the week is Tuesday\n");
break;
case 3 :printf("\nDay of the week is
Wednesday\n"); break;
case 4:printf("\nDay of the week is Thursday\n");
break;
case 5 :printf("\nDay of the week is Friday\n");
break;
case 6 :printf("\nDay of the week is Saturday\n");
break;
default :printf("\nWrong input\n") ;
}
}
```

Output:
Enter the day of week as number: 4
Day of the week is Thursday

ਮਿਸਾਲ 8:

```
// Program to evaluate simple expression of the
form
// value operator value
#include<stdio.h>
void main()
{
float value1 , value2;
char operator1 ;
printf("Type in your expression. \n") ;
scanf("%f %c %f", &value1 ,&operator1,
&value2);
switch (operator1)
{
 case '+' :
    printf ("%.2f \n" , value1+ value2);
  case '-' :
            printf ("%.2f \n" , value1 - value2);
    break ;
  case '*':
    printf ("%.2f \n" , value1 * value2);
    break ;
  case '/' :
    if (value2 == 0)
          printf ("Division by Zero. \n");
    else
          printf ("%.2f \n" , value1 / value2);
    break ;
  default:
         printf("Unknown operator.\n");
         break ;
         }
}
```

Output:
Type in your expression.
178.99– 326.8
147.81

ਮਿਸਾਲ 9

ਜੁਮੇਟਰੀਕਲ ਅੰਕੜੇ ਦੇ ਖੇਤਰ ਨੂੰ ਗਿਣਨ ਵਾਲਾ ਪ੍ਰੋਗਰਾਮ ਜਿਵੇਂ ਕਿ ਸਰਕਲ (circle), ਸਕੁਏਰ (square), ਟਰਾਈਐਂਗਲ (triangle), ਰੈਕਟੈਂਗਲ (rectangle) ਆਦਿ। ਇਸ ਵਿਚ ਵੱਖ ਵੱਖ ਅੰਕੜਿਆਂ (figure) ਦੇ ਅੰਕੜਾ-ਕੋਡਾਂ (figure codes) ਦਾ ਮੀਨੂੰ ਪ੍ਰਦਰਸ਼ਿਤ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਵਿਸ਼ੇਸ਼ ਅੰਕੜਾ-ਕੋਡ (fig_code) ਦੀ ਚੋਣ ਨਾਲ ਅੰਕੜੇ (figure) ਵੱਲੋਂ ਉਤਰਵਰਤੀ (corresponding) ਮਾਪਦੰਡ (parameter) ਜੋ ਉਸ figure ਨੂੰ ਜ਼ਰੂਰਤ ਹੈ ਮੰਜ਼ੂਰ ਕਰਦਾ ਹੈ, ਪ੍ਰੋਸੈਸ ਕਰਦਾ ਹੈ ਅਤੇ ਖੇਤਰ ਪ੍ਰਦਰਸ਼ਿਤ ਕਰਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

```
# include <stdio.h>
main()
{
  int fig_code;
  float side, base,length,breadth,height,area, radius;
printf("=================\n");
printf("1 Circle\n");
printf("2 Rectangle\n");
printf("3 triangle\n");
printf("=================\n");
printf("enter the figure code") ;
scanf("%d " ,&fig_code);
switch(fig_code) {
case 1:       printf("Enter radius\n");
         scanf("%f" ,&radius);
         area = 3.142* radius*radius;
         printf("area of the circle = %f\n"     area) ;
         break;
case 2:       printf("Enter breadth and length\n");
         scanf("%f %f",&breadth ,&length);
         area = breadth * length
         printf("area of the rectangle = %f\n",area) ;
         break;
case 3:       printf("Enter base and height");
         scanf("%f %f ",&base, &height);
           area = 0.5* base * height ;
         printf("area of the triangle = %f\n", area) ;
         break;
default:      printf("error in the figure code\n");
         break;
  }
 }
```

Output:

```
=============
 1   Circle
 2   Rectangle
 3   Triangle
=============
```

ਅੰਕੜਾ (figure) ਕੋਡ ਪ੍ਰਵੇਸ਼ ਕਰੋ 3
ਬੇਸ ਅਤੇ ਉਚਾਈ ਪ੍ਰਵੇਸ਼ ਕਰੋ 10 and 12
ਟਰਾਈਐਂਗਲ (triangle) ਦਾ ਖੇਤਰ ਹੈ 0.6

o

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਕੰਟਰੋਲ ਫਲੋ (Control Flow)

ਅਸੀਂ ਇਸ ਅਧਿਆਇ ਵਿਚ ਦੇਖਿਆ ਹੈ ਕਿ ਇਥੇ ਸਥਿਤੀਆਂ ਦੀਆਂ ਤਿੰਨ ਕਿਸਮਾਂ ਹੁੰਦੀਆਂ ਹਨ ਜਿਹੜੀ ਕੰਟਰੋਲ ਦੇ ਫਲੋਂ ਵਿਚ ਤਬਦੀਲੀ ਦੀ ਮੰਗ ਕਰਦੇ ਹਨ। ਇਹ ਹਨ:-

(1) ਪ੍ਰੋਗਰਾਮ ਦੇ ਕੁਝ ਭਾਗਾਂ ਨੂੰ ਖਿਸਕਾਉਣਾ ਜਾਂ ਉਸੇ ਭਾਗ ਨੂੰ ਮੁੜ ਲਾਗੂ-ਕਰਨ ਲਈ ਪਿੱਛੇ ਮੁੜਨਾ।

(2) ਦੋ ਜਾਂ ਵਧੇਰੇ ਭਾਗਾਂ ਵਿਚੋਂ ਇਕ ਭਾਗ ਲਾਗੂ ਕਰਨਾ ਜੋ ਕਿ ਕਿਸੇ ਹਾਲਾਤ ਦੇ ਸਿੱਟੇ ਤੇ ਨਿਰਭਰ ਕਰਦਾ ਹੈ।

(3) ਦਿਤੇ ਗਏ ਸਮੇਂ ਲਈ ਜਾ ਤਾਂ ਇਕ ਭਾਗ ਨੂੰ ਦੁਹਰਾਉਣ ਲਈ ਲਾਗੂ ਕਰਨਾ ਜਾਂ ਜਦ ਤਕ ਕੁਛ ਸਥਿਤੀ ਦੀ ਪੂਰਤੀ ਨਹੀਂ ਹੁੰਦੀ।

ਇਨ੍ਹਾਂ ਸਥਿਤੀਆਂ ਲਈ (c) ਕੋਲ ਤਾਰ ਟਾਈਪ ਦੇ ਸਟੇਟਮੈਂਟ ਹੁੰਦੇ ਹਨ ਜਿਵੇ ਡਿਕਲੇਅਰ ਸਟੇਟਮੈਂਟ, ਇਨਪੁਟ-ਪਰਿਣਾਮ ਸਟੇਟਮੈਂਟ, ਗਣਿਤ, ਲੌਜੀਕਲ ਸਟੇਟਮੈਂਟ ਅਤੇ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਆਦਿ।

ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਦੀ ਪਰਿਭਾਸ਼ਾ ਹੈ, ਪ੍ਰੋਗਰਾਮ ਸਟੇਟਮੈਂਟ ਦੇ ਲਾਗੂ ਕਰਣ ਦਾ ਕ੍ਰਮ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਤਿੰਨ ਟਾਈਪ ਦੀਆਂ ਹੁੰਦੀਆਂ ਹਨ : ਸ਼ਰਤੀਆ, ਲੂਪਿੰਗ ਅਤੇ ਅਨਕੰਡੀਸ਼ਨਲ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ।
ਚਾਰ ਬੁਨਿਆਦੀ ਕੰਡੀਸ਼ਨਲ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਹਨ if, if-else, if-else-if ਅਤੇ switch ਸਟੇਟਮੈਂਟ।
ਇਨ੍ਹਾਂ ਨੂੰ ਚੋਣਵੇਂ ਜਾਂ ਫੈਸਲਾ ਕਰਨ ਵਾਲੇ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। goto ਕੰਡੀਸ਼ਨਲ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ ਹੈ ਜਿਹੜੀ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਨਿਸ਼ਚਿਤ ਸਟੇਟਮੈਂਟ ਦਾ ਕੰਟਰੋਲ ਬਦਲਦੀ ਹੈ।

ਲੂਪਿੰਗ ਨੂੰ ਇਟੇਰੇਟਿਵ ਜਾਂ ਰੇਪੀਟਵ (iterative or repetive) ਕੰਟਰੋਲ ਡਿਵਾਈਸ ਸਟਰਕਚਰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜਿਥੇ int, ਸਟੇਟਮੈਂਟਸ ਦਾ ਸੈੱਟ ਬਾਰ ਬਾਰ ਲਾਗੂ ਹੁੰਦਾ ਹੈ ਚਾਹੇ ਨਿਸਚਿਤ ਸਮੇਂ ਲਈ ਜਾ ਲੌਜੀਕਲ ਕੰਡੀਸ਼ਨਲ ਲਈ ਸਹੀ ਹੋਵੇ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1: ਬਹੁ ਪਸੰਦੀ ਪ੍ਰਸ਼ਨ

1.1 ਸੀ (c) ਮਿਸਾਲ ਹੁੰਦੀ ਹੈ ....
a) object oriented language
b) object based language
c) Structural programming language
d) None of these

1.2 ਸਟਰਕਚਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਅਸੈਸ (structure programming access) ਲਾਭ ਦਿੰਦੀ ਹੈ।
a) modules b) Control structures c) both a and b d) None of these

1.3 , ਨੂੰ .... ਵਜੋਂ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
a) a separator in C b) an operator in C c) terminator in C d) a delimiter in C

1.4 ਇਕ ਖਾਲੀ ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ .... ਦੁਆਰਾ ਪ੍ਰਗਟਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।
a) new line b) blank space c) semicolon d) colon

1.5 **Infinite** (ਅਨੰਤ) ਲੂਪ (loop) ਹੁੰਦਾ ਹੈ ....
a) useful for time delay b) useless c) used to terminate execution d) not possible

1.6 ਬਰੇਕ ਸਟੇਟਮੈਂਟ ਪੋਜੀਟਿਵ (break statement positive) ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ....
a) selective control structures only
b) Loop control Structures only
c) both option a and b
d) switch-case control structures only

1.7 ਨਿਰੰਤਰ ਸਟੇਟਮੈਂਟ (continue statement) ਦੀ ਵਰਤੋਂ .... ਕੀਤੀ ਜਾਂਦੀ ਹੈ
a) continue the next iteration of the loop statement
b) Exit the block of loop statement
c) continue execution of the program even errors occurs
d) Exit from the outmost block even it is used in the innermost block

ਪ੍ਰਸ਼ਨ 2: ਦੱਸੋ ਸਹੀ ਹੈ ਜਾਂ ਗਲਤ ?

1. ਹਰ continue statement ਦੀ ਸਮਾਪਤੀ ; ਨਾਲ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ?
2. else ਕੀ-ਵਰਡ ਹਮੇਸ਼ਾ if ਸਟੇਟਮੈਂਟ (if statement) ਦੇ ਨੇੜੇ ਮੇਲ ਖਾਂਦਾ ਹੈ
3. ਇਕ switch ਹਦਾਇਤ ਵਿਚ break ਸਟੇਟਮੈਂਟ (break statement) ਦੀ ਵਰਤੋਂ ਹਰ case ਵਿਚ ਹੁੰਦੀ ਹੈ ਸਿਵਾਏ ਅੰਤਿਮ ਕੇਸ ਦੇ
4. (switch statement ਵਿਚ, case ਮੁੱਲ ਹਮੇਸ਼ਾਂ ਉਪਰ ਜਾਣ ਦੇ ਕ੍ਰਮ ਵਿਚ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ
5. Continue statement ਨੂੰ ਫੈਸਲਾ ਕਰਨ ਵਾਲੇ ਸਟੇਟਮੈਂਟਸ ਦੇ ਨਾਲ ਨਹੀ ਵਰਤਿਆ ਜਾਣਾ ਚਾਹੀਦਾ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰਸ਼ਨ 3: ਸੰਖੇਪ ਪ੍ਰਕਾਰ ਦੇ ਪ੍ਰਸ਼ਨ

1. Loop ਕੀ ਹੁੰਦਾ ਹੈ ?

2. ਅਗਲੇ ਅਤੇ ਪਿਛਲੇ ਜੰਪ (jump) ਤੋਂ ਕੀ ਸਮਝਦੇ ਹੋ? ਇਨ੍ਹਾਂ ਵਿਕਲਪਾਂ ਲਈ ਕਿਹੜਾ loop ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ?

3. If-else statement ਨੂੰ ਤੁਸੀਂ ਕਦੋਂ ਪਹਿਲ ਦਿੰਦੇ ਹੋ ?

4. ਚਾਰ ਬੁਨਿਆਦੀ ਸ਼ਰਤ ਵਾਲੇ continue statement ਕਿਹੜੇ ਹਨ ?

ਪ੍ਰਸ਼ਨ 4: ਪ੍ਰਸ਼ਨਾਂ ਦੇ ਉੱਤਰ ਦਿਓ

1. Conditions ਦੇ ਨਾਲ ਲਾਗੂ ਕਰਨ ਕੀ ਹੁੰਦਾ ਹੈ? ਇਹ ਕਿਥੇ ਲਾਭਦਾਇਕ ਹੁੰਦੇ ਹਨ ?

2. ਲੋਕਾਂ ਨੂੰ ਬੱਚੇ, ਨੌਜਵਾਨ ਅਤੇ ਬੁੱਢਿਆਂ ਦੀ ਸ਼੍ਰੇਣੀ, ਜੋ ਕਿ ਉਮਰ ਗਰੁੱਪ ਤੇ ਨਿਰਭਰ ਕਰਦੀ ਹੈ ਦੀ ਗਰੁਪਿੰਗ ਦੀ ਪ੍ਰਕ੍ਰਿਆ ਵਿਚ ਤੁਸੀ ਕਿਹੜਾ control statement) ਚਾਹੋਗੇ ਅਤੇ ਕਿਉਂ ?

3. ਅਸੀਂ for loop ਕਦੋਂ ਚੁਣਦੇ ਹਾਂ ? ਇਹ while loop ਨਾਲੋ ਕਿਵੇਂ ਭਿੰਨ ਹੈ ?

4. Case label ਕੀ ਹਨ ਤੇ ਉਹ ਕਿਵੇਂ ਵਰਤੇ ਜਾਂਦੇ ਹਨ ?

ਪ੍ਰਸ਼ਨ 5: ਨਿਮਨਲਿਖਿਤ ਦੇ ਪਰਿਣਾਮਾਂ ਦੀ ਪਰਖ ਲਈ ਪ੍ਰੋਗਰਾਮ ਤਕ ਮੁਕੰਮਲ ਕਰੋ

<table>
<tr>
<td>
<pre>
void main (void)
        {
         int a,b ;
         for (a =0 ;a <10 ; a++)
         {
           switch (a % 5)
        {
         case 0 : b = 0;
         case 1 : b++ ;
         case 2 : b + = 2;
         case 3 : b + = 3;
         case 4 : b + = 4;
         default : b + = 10;
        } }
        printf("%d" , b); }
</pre>
</td>
<td>Output :
90</td>
</tr>
</table>

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰਸ਼ਨ 6: ਨਿਮਨਲਿਖਿਤ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸੰਪੂਰਣ ਕਰੋ ਅਤੇ ਪਰਿਣਾਮ ਪਤਾ ਕਰੋ

(a)
```
if (x > 10), then
        x + = 1:
    else
        x - =1;
```

(b)
```
If (a <>b)
printf("a and b are unequal):
else
printf("a and b are equal):
```

c)
```
if (value = ! 1100)
        value - -:
        printf ("value = %d\n", value)
```

(d)
```
if (x > 0)
        y = 1
else if (x == 0)
        y = 0
else if (x < 0)
        y = 1
        printf("y = %d\n", y):}
```

(e)
```
main( )
{
int a = 0, b = 0: if (a == 0)
if (b == 0)
printf("Hello Mr"):
else /'
        printf("Hello Mrs.): }
```

(f)
```
main ( )
float p = 3.2, q = 6.0;
int r = 25:
if (r)
{ p = p + 2.3;
q = p + q:}
printf("%f %f\n", p,q
```

(g)
```
main ( )
{  char ch = '*'
switch (ch)
{
case' *' : printf ("%C", ch);
case '?' : printf("%C", ch);
default: printf("%C", *):}}
```

Answer to objective questions

Q	.1	.2	.3	.4	.5	.6	.7
Q1	c	c	d	b	a	c	a
Q2	F	T	T	F	F		

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਕੰਟਰੋਲ ਫਲੋ (ਭਾਗ - 2)
Control Flow (Part - 2)

ਅਧਿਆਇ 6

6.0.0 ਭੂਮਿਕਾ (Introduction)

ਕਿਸੇ ਵੀ ਪ੍ਰੋਗਰਾਮ ਦੀ ਪ੍ਰਕਿਰਿਆ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਦੁਆਰਾ ਨਿਰਧਾਰਿਤ ਹੁੰਦੀ ਹੈ ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਇਹ ਫੈਸਲਾ ਕਰਦੀਆਂ ਹਨ ਕਿ ਇਕ ਪ੍ਰੋਗਰਾਮ ਦੇ ਕਿਸ ਭਾਗ ਨੇ ਚਲਣਾ ਹੈ ਅਤੇ ਕਿੰਨੀ ਵਾਰ ਚਲਣਾ ਹੈ। ਪਿਛਲੇ ਅਧਿਆਇ ਕੰਟਰੋਲ ਫਲੋ ਭਾਗ -1 ਵਿਚ ਅਸੀ ਇਹ ਪੜ੍ਹ ਚੁਕੇ ਹਾਂ ਕਿ ਕਿਵੇਂ ਪ੍ਰੋਗਰਾਮ ਦਾ ਕੋਈ ਵੀ ਖੰਡ, counter ਦੀ ਮਦਦ ਨਾਲ ਬਾਰ-ਬਾਰ ਚਲਾਇਆ ਜਾ ਰਕਦਾ ਹੈ ਅਤੇ ਇਸ ਦੀ ਜਾਂਚ if ਸਟੇਟਮੈਂਟ ਦੁਆਰਾ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ, ਲੇਕਿਨ ਇਹ ਤਰੀਕਾ ਸਿਰਫ ਵਿਵਹਾਰਿਕ ਕਾਰਜਾਂ ਲਈ ਹੀ ਭਰੋਸੇਯੋਗ ਹੰਦਾ ਹੈ। ਪ੍ਰੋਗਰਾਮ ਦੇ ਲੂਪ ਦੀ ਸਹੀ ਸਮਾਪਤੀ ਲਈ counter ਨੂੰ initialize, ਬੜੋਤਰੀ (increment) ਕਰਨ ਅਤੇ ਉਸ (counter) ਦੇ ਮੁਲਾਂ ਦੀ ਓਚਿਤ ਜਗ੍ਹਾ ਤੇ ਜਾਂਚ ਕਰਨ ਦੀ ਜਰੂਰਤ ਹੁੰਦੀ ਹੈ। ਆਮ ਤੌਰ ਤੇ, if ਸਟੇਟਮੈਂਟ ਨਾਲ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਪਹਿਲਾਂ ਤੋਂ ਇਹ ਨਿਸਚਿਤ ਨਹੀ ਹੁੰਦਾ ਕਿ ਇਕ ਲੂਪ ਨੇ ਕਿੰਨੀ ਵਾਰ ਚਲਣੰਾ ਹੈ।

ਇਸ ਅਧਿਆਇ ਵਿਚ ਅਸੀ for ਅਤੇ while loop ਨੂੰ ਇਸਤੇਮਾਲ ਕਰਕੇ ਬਾਰ-ਬਾਰ ਦੋਹਰਾਉਣ ਦੀ ਪ੍ਰਕਿਰਿਆ ਨੂੰ ਪਹਿਲਾਂ ਤੋਂ ਹੀ ਸਿਖਾਂਗੇ ਅਤੇ ਆਮ ਵਰਤੋ ਵਿਚ ਆਉਣ ਵਾਲੇ ਪ੍ਰੋਗਰਾਮ ਬਨਾਉਣੇ ਸਿਖਾਂਗੇ ।

6.1.0 ਕੰਟਰੋਲ ਲੂਪ ਸਟਰਕਚਰ (Control loop structures)

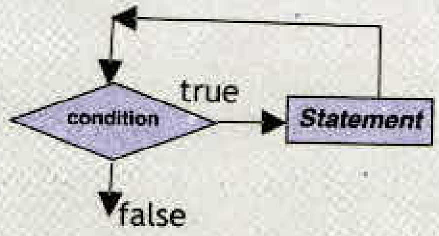
ਬਾਰ-ਬਾਰ ਲਾਗੂ ਕਰਨ ਲਈ ਬਹੁਤ ਸਾਰੇ ਪ੍ਰੋਗਰਾਮਾਂ ਨੂੰ ਹਿਦਾਇਤਾਂ ਦੇ ਇਕ ਸਮੂਹ (set) ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ ਜਦ ਤਕ ਕਿ ਕੋਈ ਲੋਜਿਕਲ (logical) ਸਥਿਤੀ ਦੀ ਸੰਤੁਸ਼ਟੀ ਨਾ ਹੋ ਜਾਵੇ।

ਇਹ ਲੂਪ ਸਟਰਕਚਰਸ ਹਨ

6.1.1 While ਸਟੇਟਮੈਂਟ (while statement):

While Statement ਨੂੰ While Loop ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਟੇਟਮੈਂਟਸ ਦੇ ਬਲਾਕ ਨੂੰ ਓਨੀ ਦੇਰ ਲਾਗੂ ਕਰਦੀ ਹੈ ਜਦੋਂ ਤਕ ਇਕ ਨਿਸ਼ਚਿਤ ਸਥਿਤੀ ਸਹੀ ਹੁੰਦੀ ਹੈ। While statement ਦੇ ਹੇਠ ਲਿਖੇ ਰੂਪ ਹੋਣਗੇ ।

```
while (condition)
  { statement1;
    statement2;
    ...........  }
```



ਕੰਡੀਸ਼ਨ, ਕੋਈ ਵੀ ਸੀ (C) ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਹੈ, ਅਤੇ ਸਟੇਟਮੈਂਟ (statement) ਇਕਲਾ ਹੈ ਜਾਂ ਮਿਸ਼ਰਤ ਸੀ (C) ਸਟੇਟਮੈਂਟ। ਜਦੋਂ ਪ੍ਰੋਗਰਾਮ ਲਾਗੂਕਰਨ while statement ਤੇ ਪਹੁੰਚ ਜਾਂਦਾ ਹੈ ਤਾਂ ਹੇਠ ਲਿਖੀਆ ਪ੍ਰਤਿਕ੍ਰਿਆਵਾਂ ਵਾਪਰਦੀਆਂ ਹਨ :

1. ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਸਥਿਤੀ ਦਾ ਮੁਲਾਂਕਣ ਹੁੰਦਾ ਹੈ।
2. if condition ਦਾ ਗਲਤ ਮੁਲਾਂਕਣ ਹੁੰਦਾ ਹੈ (ਉਹ ਹੈ ਜ਼ੀਰੋ) ਤਾਂ while statement ਸਮਾਪਤ ਹੁੰਦਾ ਹੈ ਅਤੇ ਲਾਗੂਕਰਨ ਨਿਮਨਲਿਖਤ ਸਟੇਟਮੈਂਟ (statement) ਦੇ ਪਹਿਲੇ ਸਟੇਟਮੈਂਟ (statement) ਵੱਲ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

3. if condition ਸਹੀ (ਉਹ ਗੈਰ ਜ਼ੀਰੋ ਹੁੰਦੀ ਹੈ) ਦਾ ਮੁਲਾਂਕਣ ਕਰਦੀ ਹੈ ਤਾਂ ਸੀ (C) ਸਟੇਟਮੈਂਟਸ (statements) ਦਾ ਲਾਗੂ ਕਰਨ ਹੁੰਦਾ ਹੈ।
4. ਲਾਗੂ ਕਰਨ ਕਦਮ-1 ਵਿਚ ਮੁੜ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ: ਪਹਿਲੇ n ਪੂਰਨ ਅੰਕ ਦਾ ਜੋੜ ਲਭਣ ਲਈ ਪ੍ਰੋਗਰਾਮ **(use of while loop)**

```
#include <stdio.h>
void main()
 { int n ,a ,sum = 0 ;
   printf("Enter the value of N\n");
   scanf("%d" , &n);
  a =1;
while (a<n)
{
  sum+ =a;
  a++ ;
}                         /* end of while */
  printf("sum = %4d\n" , sum);
}
```

Output:

Enter the value of N
10
sum = 55

ਮਿਸਾਲ : ਇਕ ਆਸਾਨ while ਸਟੇਟਮੈਂਟ (A simple while statement).

```
/* Demonstrates a simple while statement  to display first ten number*/

#include <stdio.h>
int count;                          /* global declaration */
int main()
{                                   /* print the numbers 1 through 10 */
 count = 1;
 while (count <= 10)
 {
   printf("%d/t", count);
   count++;                         /* or count = count +1*/
 }
return 0;
}
```

Output:
1 2 3 4 5 6 7 8 9 10

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

6.1.2 Do while:

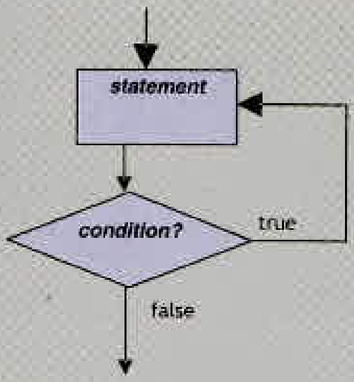
- **The syntax for the do-while is:**

```
do {

    statement1 ;
    statement2 ; ...

} while (expression);
```

ਪਹਿਲਾਂ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਦਾ ਮੁਲਾਂਕਣ ਹੁੰਦਾ ਹੈ। ਸਟੇਟਮੈਂਟਸ (statements) ਦੇ ਬਲਾਕ ਪਹਿਲਾਂ ਲਾਗੂ ਹੁੰਦੇ ਹਨ। ਫਿਰ ਸਟੇਟਮੈਂਟ (statement) ਬਾਰ ਬਾਰ ਜਦ ਤਕ ਮੁੱਲ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਗਲਤ ਨਹੀਂ ਹੋ ਜਾਂਦਾ ਲਾਗੂ ਕਰਨ ਵਿਚ ਆਉਂਦੇ ਹਨ।

ਮਿਸਾਲ : ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਲੂਪ 9 ਵਾਰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਜਦੋਂ x ਦਾ ਮੁੱਲ 10 ਬਣ ਜਾਂਦਾ ਹੈ ਤਾਂ ਕੰਟਰੋਲ (control) ਲੂਪ (loop) ਵਿਚੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ।

```
#include<stdio.h>
void main()
 {
 int x;
 x =1 ;
 do
   {
    printf("x value is %d\n" ,x);
    x+ = 1;
   } while (x <10) ;
 }
```



do execution ਤੋਂ ਬਾਅਦ ਸਟੇਟਮੈਂਟਸ statements) ਦੇ ਬਲਾਕ ਨੂੰ ਘਟੋ ਘਟ ਇਕ ਵਾਰ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਫਿਰ ਕੰਟਰੋਲ ਦੀ ਪਰਖ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਤਾਂ ਜੋ ਲੂਪ 9 ਵਾਰ ਲਾਗੂ ਹੋ ਸਕੇ।

ਮਿਸਾਲ : ਮਿਸਾਲ ਵਿਚ ਜੇਕਰ ਅਸੀਂ while (x>10) ਲਿਖਦੇ ਹਾਂ ਤਾਂ ਇਨਪੁਟ ਵਿਚ ਨਿਮਨ ਲਿਖਤ ਅੰਤਰ ਦੇਖੋ:

x is 1

- **x is assigned with 1.**
- **prints as x is 1.**
- **x is incremented by 1. so x is 2 now**
- **Tests for condition x>10**
- **since it is false, it comes out of the loop**

So ,this program prints once as x is 1

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ : n ਅੰਕਾਂ ਦੀ ਔਸਤ ਕੱਢੋ

```
#include <stdio.h>
void main ()
{
 int n ,count =1;
 float x ,average ,sum = 0;          /*initialize */
 printf("how many numbers ? "); /* read in a value for n*/
 scanf("%d" ,&n);    /* read in the numbers */
 do
   {   printf("x =");
       scanf("%f" ,&x);
       sum+ = x;
   }            /*To calculate the average & write the answer */
 while (count < = n);
  average = sum/ n;
  printf("\n the average is %f\n ", average);
}
```

ਮਿਸਾਲ : The program prints the sum of all odd integers between 1 to 50

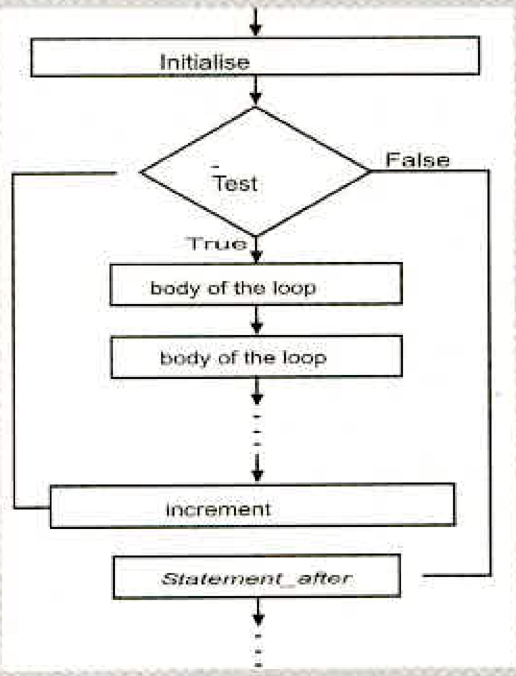
```
#include <stdio.h>
void main ()
{
  int odnum ,sum = 0;
  odnum =1;
  do
 {
   sum+ = odnum;
   odnum+ = 2;
  } while(odnum<=50);
 printf("sum = %4d\n" ,sum);
}
```

Output:
Sum = 625

6.1.3 For Statement (loop): for statement ਦਾ ਉਪਯੋਗ ਉਹਨਾ ਸਮੱਸਿਆ ਵਿਚ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਹਨਾਂ ਵਿਚ ਇਕ ਸਟੇਟਮੈਂਟ ਇਕ ਤੋਂ ਵਧ ਪੂਰਵ ਨਿਰਧਾਰਿਤ ਵਾਰ ਕਿਸੇ ਸਟੇਟਮੈਂਟ-ਬਲਾਕ (ਇਕ ਤੋਂ ਵਧ ਸਟੇਟਮੈਂਟ) ਨੂੰ ਚਲਾਂਉਦੀ ਹੈ।

```
for (expression1 ; expression2 ;expression3)
{
 action-statements;
}
```

ਐਕਸਪ੍ਰੈਸ਼ਨ 1 (expression1) ਦਾ ਪਹਿਲਾਂ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਫਿਰ ਐਕਸਪ੍ਰੈਸ਼ਨ 2 (expression2) ਦਾ ਮੁਲਾਂਕਣ (evaluation) ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜੋ ਇਹ ਜਾਂਚ ਕਰਦਾ ਹੈ ਕਿ ਲੂਪ (loop) ਦੋਹਰਾਇਆ ਗਿਆ ਹੈ। ਸਟੇਟਮੈਂਟ (statement) ਨੂੰ ਲਾਗੂ ਕਰਨ ਤੋਂ ਬਾਅਦ ਐਕਸਪ੍ਰੈਸ਼ਨ 3 (expression3) ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਪ੍ਰਕਾਰ ਐਕਸਪ੍ਰੈਸ਼ਨ 1 (expression1) ਆਮ ਕਰਕੇ ਲੂਪ ਕਾਊਂਟਰ ਤੋ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ ਅਤੇ ਐਕਸਪ੍ਰੈਸ਼ਨ 3 (expression3) ਲੂਪ ਕਾਊਂਟਰ ਦੇ ਵਧਾਉਣੇ/ਘਟਾਉਣੇ (increment/decrement) ਨਾਲ।



ਮਿਸਾਲ : for (count = 1; count <= 10; count = count +1)

- ਜਦੋਂ for ਸਟੇਟਮੈਂਟ (for statement) ਪਹਿਲੀ ਵਾਰ ਲਾਗੂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਗਿਣਤੀ ਦਾ ਮੁੱਲ ਮੁੱਢਲਾ ਮੁੱਲ 1 ਨਿਸ਼ਚਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

- ਹੁਣ ਸਥਿਤੀ ਦੀ ਗਿਣਤੀ ਹੁੰਦੀ ਹੈ <=10 ਜੋ ਪਰਖੀ ਗਈ ਹੈ। ਕਿਉਂ ਜੋ ਗਿਣਤੀ 1 ਹੈ ਇਸ ਲਈ ਸਥਿਤੀ ਤਸੱਲੀ ਬਖਸ਼ ਹੈ ਅਤੇ ਲੂਪ ਦੀ ਬਾਡੀ ਪਹਿਲੀ ਵਾਰ ਲਾਗੂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
- **for** ਦੇ ਕਲੋਜ਼ਿੰਗ ਬਰੇਸ (closing brace) ਤੇ ਪਹੁੰਚਣ ਤੇ ਕੰਪਿਊਟਰ, ਕੰਟਰੋਲ ਨੂੰ **for ਸਟੇਟਮੈਂਟ** (for statement) ਵੱਲ ਵਾਪਿਸ ਭੇਜ ਦਿੰਦਾ ਹੈ ਜਿਥੇ ਗਿਣਤੀ ਮੁੱਲ 1 ਦੁਆਰਾ ਵਧ ਜਾਂਦਾ ਹੈ।
- ਮੁੜ ਪਰਖ ਇਸ ਗੱਲ ਦੀ ਜਾਂਚ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ ਗਿਣਤੀ ਦਾ ਨਵਾਂ ਮੁੱਲ ਵਧ ਗਿਆ ਹੈ ਤੇ 10 ਹੋ ਗਿਆ ਹੈ।
- ਜੇਕਰ ਗਿਣਤੀ ਦਾ ਮੁੱਲ ਅਜੇ ਵੀ 1 ਤੋਂ 10 ਤਕ ਹੈ ਤਾਂ ਬਰੇਸਿਸ (braces) ਦੇ ਵਿਚ ਸਟੇਟਮੈਂਟ (statement) ਮੁੜ ਲਾਗੂ ਹੁੰਦੇ ਹਨ।
- **for ਲੂਪ** (for loop) ਦੀ ਬਾਡੀ ਉਨੀ ਦੇਰ ਲਾਗੂ ਰਹਿੰਦੀ ਹੈ ਜਦ ਤਕ ਗਿਣਤੀ ਅੰਤਿਮ ਮੁੱਲ 10 ਤੋਂ ਵਧ ਨਹੀਂ ਜਾਂਦਾ।
- ਜਦੋਂ ਗਿਣਤੀ 11 ਮੁੱਲ ਤੇ ਪਹੁੰਚ ਜਾਂਦੀ ਹੈ ਤਾਂ ਕੰਟਰੋਲ, ਲੂਪ (loop) ਚੋਂ ਨਿਕਲ ਜਾਂਦਾ ਹੈ ਅਤੇ ਕੰਟਰੋਲ for ਦੀ ਬਾਡੀ ਦੇ ਤੁਰੰਤ ਬਾਅਦ ਕੋਈ ਸਟੇਟਮੈਂਟ (statement) ਜੋ ਹੋਵੇ ਉਸਤੇ ਚਲਾ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ : 10 ਪ੍ਰਕ੍ਰਿਤਕ ਅੰਕਾਂ ਦੇ ਜੋੜ ਛਾਪਣ ਲਈ
ਪ੍ਰੋਗਰਾਮ **(use of for loop)**

```
#include <stdio.h>
void main()
{
int sum , number ;
 sum=0;            /*initialize sum to 0*/
   for (number =1; number <=10; number ++)
   sum + =number ;
   printf("sum = %d \n" , sum);
}
```

Output:

Sum =55

ਸਪਸ਼ਟੀਕਰਨ (Explanation): ਪਹਿਲੀ ਵਾਰ for ਸਟੇਟਮੈਂਟ (for statements) ਲਾਗੂ ਹੁੰਦਾ ਹੈ ਇਹ ਜੋੜ ਦਾ ਮੁੱਲ, 0 ਤੋਂ ਸ਼ੁਰੂ ਕਰਦਾ ਹੈ ਅਤੇ ਅੰਕ ਦਾ ਮੁੱਲ 1 ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। ਸੀ (C) ਵਲੋਂ ਫਿਰ ਇਹ ਪਰਖ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ ਇਸਨੇ for ਦੀ ਬਾਡੀ ਦਾ ਮੁਲਾਂਕਣ ਕਰਨਾ ਹੈ ਜੋ ਕਿ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਅੰਕ ਹੈ <=0 ਜਿਵੇਂ ਕਿ ਅੰਕ ਦਾ ਮੁੱਲ 1 ਹੈ। ਇਹ for ਦੀ ਬਾਡੀ ਨੂੰ ਲਾਗੂ ਕਰਦਾ ਹੈ। ਸੀ (C) ਵਲੋਂ ਫਿਰ ਤੀਜਾ ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਹੜਾ 1 ਅੰਕ ਤੋਂ ਵੱਧ ਹੁੰਦਾ ਹੈ ਜੋ ਲੂਪ ਦੇ ਅਗਲੇ ਅੰਤਰ ਸੰਬੰਧ ਨੂੰ ਤਿਆਰ ਕਰਦਾ ਹੈ। ਸੀ (C) ਵਲੋਂ ਮੁੜ ਪਰਖ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ ਇਸ ਵਲੋਂ ਐਕਸਪ੍ਰੈਸ਼ਨ2 (expression 2) ਦੇ ਲਾਗੂ ਕਰਨ ਨਾਲ for ਦੀ ਬਾਡੀ ਲਾਗੂ ਕੀਤੀ ਗਈ ਹੈ। ਅੰਕ <=10 ਜਿਵੇਂ ਕਿ ਅੰਕ ਦਾ ਮੁੱਲ ਹੁਣ 2 ਹੈ, ਸੀ (C), for ਦੀ ਬਾਡੀ ਮੁੜ ਲਾਗੂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਜਦੋਂ ਬਾਡੀ ਨੂੰ 10 ਗੁਣਾ ਤਿਆਰ ਕਰ ਦਿੰਦੀ ਹੈ ਅਤੇ ਅਗਲੇ ਅੰਤਰ ਸੰਬੰਧ ਨੂੰ ਤਿਆਰ ਕਰਦੀ ਹੈ ਤਾਂ ਅੰਕ ਦਾ ਮੁੱਲ 11 ਹੋ ਜਾਂਦਾ ਹੈ। ਇਸ ਵਾਰ ਐਕਸਪ੍ਰੈਸ਼ਨ2 (expression 2) ਵਿਚ ਪਰਖ ਗਲਤ ਹੁੰਦੀ ਹੈ ਅਤੇ ਕੰਟਰੋਲ printf ਸਟੇਟਮੈਂਟ (statements) ਕੋਲ ਚਲਾ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ ਪ੍ਰਿੰਟ ਲਈ ਪ੍ਰੋਗਰਾਮ (Program to print)

```
/*for loop header*/
#include<stdio.h>
void main()
{
 int i;          /* In this example for loop braces not required as body contains only */
 for (i = 1; i <= 5; i = i +1)          /* one statement */
 printf("i = %d", i);
}
```

ਇਸ ਮਿਸਾਲ ਵਿਚ i ਦਾ ਮੁੱਢਲਾ ਮੁੱਲ 1 ਹੈ। ਇਸ ਗੱਲ ਤੇ ਕੰਟਰੋਲ (control) ਕਰਨ ਲਈ ਕਿ ਜੇਕਰ i, 5 ਤੋਂ ਘੱਟ ਹੁੰਦਾ ਹੈ ਜਾਂ 5 ਦੇ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ। ਸਟੇਟਮੈਂਟ (statement) i= i+1 ਭਾਵ ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਚੋ ਹਰ ਵਾਰ ਗੁਜਰਦਿਆਂ ਕੰਟਰੋਲ-ਵੇਰੀਏਬਲ (control variable) ਹਰ ਵਾਰ i ਤੋਂ ਵਧ ਚਲੇਗਾ ਜਦ ਤਕ ਕਿ ਇਹ 6 ਤੇ ਨਹੀ ਪੁੱਜ ਜਾਂਦਾ (ਅਤੇ ਇਹ ਮੁੱਲ ਪਰਖ ਕੰਟਰੋਲਿੰਗ ਵਿਚ ਲੂਪ ਦੇ ਗਲਤ ਹੋਣ ਦਾ ਕਾਰਣ ਬਣਦਾ ਹੈ।)

for ਲੂਪ ਹੈਡਰ (loop header) ਤੇ ਬਾਅਦ ਲੂਪ ਦੀ ਬਾਡੀ ਹੁੰਦੀ ਹੈ

- ਵੇਰੀਏਬਲ i ਇਕ (1) ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦੀ ਹੈ, ਜੋ ਕਿ ਆਪਸੀ ਮੁੱਲ 5 ਨਾਲ ਤੁਲਨਾ ਕਰਦਾ ਹੈ ਕਿ i ਦਾ ਮੁੱਲ 5 ਤੋਂ ਘੱਟ ਜਾਂ ਬਰਾਬਰ ਹੈ, ਪ੍ਰੋਗਰਾਮ ਲੂਪ ਵਿਚ ਜਾਂਦਾ ਹੈ ਅਤੇ printf( ) ਹਦਾਇਤ ਨੂੰ ਲਾਗੂ ਕਰਦਾ ਹੈ। ਕੁਦਰਤੀ ਹੈ ਕਿ **printf** ਹਿਦਾਇਤ ਮੁੱਲ 1 ਨੂੰ ਛਾਪਦੀ ਹੈ। ਇਹ ਲੂਪ ਲਈ ਬਾਡੀ ਰਾਹੀ ਪਹਿਲੀ ਵਾਰ ਗੁਜਰਨ ਨੂੰ ਮੁਕੰਮਲ ਕਰਦੀ ਹੈ।
- ਫਿਰ ਅਸੀ i ਵਿਚ ਵਾਧਾ ਕਰਕੇ 1 ਤੋਂ 2 ਬਣਾਉਦੇ ਹਾਂ ਅਤੇ ਇਸ ਨਵੇਂ ਮੁੱਲ ਨੂੰ 5 ਨਾਲ ਤੁਲਨਾ ਕਰਦੇ ਹਾਂ। ਜਿਵੇਂ ਕਿ 2 ਦਾ ਮੁੱਲ 5 ਨਾਲੋਂ ਘੱਟ ਹੈ ਜਾਂ ਬਰਾਬਰ ਹੈ। ਅਸੀ ਲੂਪ ਦੀ ਬਾਡੀ ਨੂੰ ਲਾਗੂ ਕਰਦੇ ਹਾਂ । ਇਸ ਵਾਰ printf ( ) ਵਲੋਂ 2 ਛਾਪਿਆ ਜਾਂਦਾ ਹੈ ਜੋ ਨਵੀ ਰੇਖਾ ਤੇ ਹੁੰਦਾ ਹੈ। ਇਹ ਦੂਜੇ ਗੁਜਰਨ ਨੂੰ ਸੰਪੂਰਣ ਕਰਦਾ ਹੈ।
- ਫਿਰ i ਵਧ ਕੇ 3 ਹੋ ਜਾਂਦਾ ਹੈ ਜਿਹੜਾ ਕਿ ਕੰਟਰੋਲ ਵੇਰਿਏਬਲ (control variable) 5 ਦੇ ਸੀਮਿਤ ਮੁੱਲ ਦੇ ਬਰਾਬਰ ਜਾਂ ਘੱਟ ਹੁੰਦਾ ਹੈ। ਇਸ ਲਈ ਅਸੀਂ 3 ਛਾਪਦੇ ਹਾਂ।
- ਇਸੇ ਤਰ੍ਹਾਂ i ਵਧ ਕੇ 4 ਹੋ ਜਾਂਦਾ ਹੈ ਅਤੇ ਅਸੀ ਛਾਪਦੇ ਹਾਂ 4
- ਫਿਰ i ਵਧਕੇ 5 ਹੋ ਜਾਂਦਾ ਹੈ ਜਿਸਦੀ ਆਗਿਆ ਹੁੰਦੀ ਹੈ ਅਸੀ ਪਰਖ ਕਰਦੇ ਹਾਂ ਕਿ ਚਾਲੂ ਮੁੱਲ ਘੱਟ ਹੈ ਜਾਂ ਅੰਤਿਮ ਤੋਂ ਵੱਧ ਹੈ। printf ਛਾਪਦਾ ਹੈ 5
- ਫਿਰ i ਵਧ ਕੇ 6 ਹੋ ਜਾਂਦਾ ਹੈ। ਇਸ ਨੁਕਤੇ ਦੇ ਮੁਤਾਬਿਕ, <=5 ਠੀਕ ਨਹੀ ਹੁੰਦੀ ਇਸ ਲਈ ਅਸੀਂ i ਜਿਸਦਾ ਮੁੱਲ 6 ਹੁੰਦਾ ਹੈ ਦੇ ਨਾਲ ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਵਿਚ ਪ੍ਰਵੇਸ਼ ਨਹੀ ਕਰਦੇ ਇਸਦੀ ਬਜਾਏ for ਲੂਪ (for loop) ਮੁਕੰਮਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਅਸੀ ਅਗਲੇ ਸਟੇਟਮੈਂਟ (statement) ਦੇ ਨਾਲ ਜਾਰੀ ਕਰਦੇ ਹਾਂ। ਸਾਡੀ ਮਿਸਾਲ ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਵਿਚ ਇਕ ਸਟੇਟਮੈਂਟ (statement) ਹੁੰਦਾ ਹੈ ਜੋ i ਦਾ ਮੁੱਲ ਛਾਪਦਾ ਹੈ, ਸਮੁੱਚਾ for ਲੂਪ (for loop) ਇਕ ਵਖਰੀ ਰੇਖਾ ਤੇ 1,2,3,4 ਅਤੇ 5 ਮੁੱਲਾਂ ਨੂੰ ਛਾਪਦਾ ਹੈ।

ਮਿਸਾਲ : 4 ਤੋ 9 ਤਕ ਅੰਕਾਂ ਨੂੰ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਵਰਗ ਮੂੱਲ (square root) ਛਾਪੋ

```
#include <stdio.h>
 void main( )
 {
  int number , sqnumber ;
  for (number = 4; number <=9; number =
number +1)
   {
    sqnumber = number * number ;
    printf("%d %d\n" , number,sqnumber);
   }
 }
```

Output:

4	16
5	25
6	36
7	49
8	64
9	81

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਪਸ਼ਟੀਕਰਨ (Explanation)

- ਜਦੋਂ **for** ਲੂਪ ਸਟੇਟਮੈਂਟ (statement) ਲਾਗੂ ਹੁੰਦਾ ਹੈ ਤਾਂ ਅੰਕ 4 ਮੁੱਲ ਦੇ ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦੇ ਹਨ। (ਇਸ ਨੁਕਤੇ ਵਰਗ-ਮੁੱਲ (sqnumber) ਕੀ ਹੁੰਦਾ ਹੈ।)
- ਜਿਵੇਂ ਕਿ ਇਹ ਮੁੱਲ ਸੀਮਤ ਜਾਂ ਅੰਕ (ਇਸ ਕੇਸ ਵਿਚ 9) ਦੇ ਮੁੱਲ ਦੇ ਬਰਾਬਰ ਜਾਂ ਘੱਟ ਹੁੰਦਾ ਹੈ, ਅਸੀਂ ਲੂਪ (loop) ਦੀ ਬਾਡੀ (body) ਲਾਗੂ ਕਰ ਸਕਦੇ ਹਾਂ।
- ਲੂਪ ਦੀ ਬਾਡੀ ਲਈ ਬਰੈਕਟ ਖੋਲ੍ਹਣ ਦੇ ਅੰਦਰ sqnumber ਨੂੰ number * number ਨਾਲ ਤੈਅ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਜਿਹੜਾ ਇਸ ਕੇਸ ਵਿਚ 4 * 4 = 16 ਹੈ। printf ਫੰਕਸ਼ਨ (function) 4 ਅਤੇ 16 ਮੁੱਲਾਂ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ।
- ਅਸੀਂ ਜਦੋਂ ਅੰਤਿਮ ਬਰੈਕਟ ਤੇ ਪਹੁੰਚਦੇ ਹਾਂ ਤਾਂ ਪ੍ਰੋਗਰਾਮ ਆਪਣੇ ਆਪ ਲੂਪ ਹੈਡਰ (loop header) ਵਲ ਬਦਲ ਜਾਂਦਾ ਹੈ ਜਿਥੇ ਅੰਕ 1 ਨਾਲ ਵਧਦਾ ਹੈ ਇਸਨੂੰ 5 ਦੇ ਬਰਾਬਰ ਬਣਾ ਦਿੰਦਾ ਹੈ।
- ਕਿਉਂਜ 5 ਸੀਮਤ ਮੁੱਲ 9 ਨਾਲੋਂ ਘੱਟ ਜਾਂ ਬਰਾਬਰ ਹੈ ਤਾਂ ਅਸੀਂ ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਲਾਗੂ ਕਰਦੇ ਹਾਂ।
- ਇਸ ਵਾਰ sqnumber 5 * 5 = 25 ਤੇ ਨਿਰਧਾਰਤ ਕੀਤੇ ਜਾਂਦਾ ਹੈ। printf 5 ਅਤੇ 25 ਨਵੀਂ ਲਾਈਨ ਤੇ ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ।
- ਅੰਕ ਮੁੜ 1 ਨਾਲ ਵਧ ਜਾਂਦੇ ਹਨ ਤਾਂ ਜੋ ਇਸਦਾ ਮੁੱਲ ਹੁਣ 6 ਹੋ ਜਾਵੇ।
- ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਲਾਗੂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ sqnumber ਹੁੰਦਾ ਹੈ 6* 6 = 36 ਅਤੇ 6 ਅਤੇ 36 ਪ੍ਰਿੰਟ ਹੁੰਦੇ ਹਨ।
- ਜਦੋਂ ਅੰਕ 7 ਅਤੇ 8 ਹੁੰਦਾ ਹੈ ਅਸੀਂ ਪ੍ਰਿੰਟ ਕਰਦੇ ਹਾਂ 7 49 ਅਤੇ 8 64
- ਪ੍ਰਿੰਟਿੰਗ ਤੋਂ ਬਾਅਦ, ਅਸੀਂ ਹੈਡਰ (header) ਵੱਲ ਵਾਪਸ ਜਾਂਦੇ ਹਾਂ ਜਿਥੇ ਅੰਕ 9 ਤਕ ਵਧ ਜਾਂਦਾ ਹੈ।
- ਕਿਉਂ ਜੋ 9 ਘੱਟ ਹੈ ਜਾਂ 9 ਦੇ ਬਰਾਬਰ ਹੈ ਅਸੀਂ ਇਕ ਵਾਰ ਮੁੜ ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਲਾਗੂ ਕਰਦੇ ਹਾਂ।
- ਇਸ ਵਾਰ Sqnumber ਨੂੰ 9 * 9 = 81 ਤੇ ਨਿਰਧਾਰਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਮੁਲ 9 81 ਪ੍ਰਿੰਟ ਹੁੰਦੇ ਹਨ।
- ਜਦੋਂ ਅੰਕ 10 ਤਕ ਵਧ ਜਾਂਦਾ ਹੈ, ਲੂਪ (loop) ਤੇ ਕੰਟਰੋਲ (control) ਕਰਨ ਵਾਲੀ ਸ਼ਰਤ ਗਲਤ ਹੋ ਜਾਂਦੀ ਹੈ।
- ਇਸ ਲਈ ਅਸੀਂ ਮੁੜ ਲੂਪ (loop) ਦੀ ਬਾਡੀ ਵਿਚ ਪ੍ਰਵੇਸ਼ ਨਹੀਂ ਕਰਦੇ। ਇਸ ਦੀ ਬਜਾਏ ਅਸੀਂ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਅਗਲੇ ਸਟੇਟਮੈਂਟ (statement) ਤੇ ਜਾਰੀ ਰਹਿੰਦੇ ਹਾਂ।

ਮਿਸਾਲ : 1 ਤੋਂ 5 ਤਕ ਅੰਕ ਛਾਪਣ ਲਈ ਬਿਨਾ ਐਕਸਪ੍ਰੇਸ਼ਨ1 (expression1) ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ 3 (expression3)

```
#include <stdio.h>
void main( )
{ int I =0; limit =5;
  printf("Value of i \n");
  for (; i <limit ;)
  {
    i++;
    printf("%d\n", i);
  }                    /* end of for loop *\
}
```

Output:

Value of i

```
1
2
3
4
5
```

ਇਥੇ ਹਰ ਲਾਂਘੇ ਤੋਂ ਬਾਅਦ i ਦਾ ਮੁੱਲ ਲੂਪ (loop) ਲਈ ਬਾਡੀ ਦੇ ਵਿਚ-ਵਿਚ ਵਧ ਜਾਂਦਾ ਹੈ ਅਤੇ i ਦਾ ਪਹਿਲਾ ਮੁੱਲ ਲੂਪ ਲਈ ਬਾਹਰ ਜ਼ੀਰੋ ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ।

o

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਕੰਟਰੋਲ ਫਲੋ (Control Flow)

while, do-while ਅਤੇ for loop ਸਟੇਟਮੇਂਟ ਤਿੰਨ ਸ਼ਕਤੀਸ਼ਾਲੀ ਲੂਪ ਹਨ ਜੋ C ਵਿਚ ਉਪਲੱਬਧ ਸਟਰਕਚਰ ਤੇ ਕੰਟਰੋਲ ਕਰਦੇ ਹਨ। while ਸਟੇਟਮੇਂਟ ਪ੍ਰੀ-ਟੈਸਟ ਸਟੇਟਮੇਂਟ ਹੈ (ਇਹ ਤਾਂ ਹੀ ਲਾਗੂ ਹੁੰਦੀ ਹੈ ਜਦੋਂ ਐਕਸਪ੍ਰੇਸ਼ਨ ਸਹੀ ਹੋਵੇ) ਅਤੇ do-while ਇਕ ਪੋਸਟ-ਟੈਸਟ ਲੂਪ ਹੈ (ਇਹ ਇਕੋ ਵਾਰ ਲਾਗੂ ਹੁੰਦੀ ਹੈ ਫਿਰ ਏਕਸਪ੍ਰੇਸ਼ਨ-ਲੂਪ ਲਈ ਦੁਹਰਾਏ ਜਾਣ ਤੇ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ)। for ਲੂਪ ਦੀ ਵਰਤੋਂ ਉਦੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜਦੋਂ ਯੂਜ਼ਰ ਜਾਣਦਾ ਹੋਵੇ ਕਿ number of ਇਟੀਰੇਸ਼ਨ ਕਿੰਨੇ ਬਣਾਏ ਜਾਣੇ ਹਨ। ਕਈ ਵਾਰ ਲੂਪ ਨੂੰ ਬਾਹਰ ਕੱਢਣ ਦੀ ਇੱਛਾ ਹੁੰਦੀ ਹੈ ਜਾਂ ਇਟੇਰੇਟਿਵ ਵਿਚ ਲੂਪ ਦੇ ਭਾਗ ਨੂੰ ਖਿਸਕਾਉਣਾ ਪੈਂਦਾ ਹੈ। ਅਜਿਹਾ ਕਰਨ ਲਈ ਬਰੇਕ break; ਅਤੇ ਨਿਰੰਤਰ continue; ਸਟੇਟਮੇਂਟ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1: ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ (Fill in the blanks)

1.1	while ਲੂਪ (while loop) ਨੂੰ ਘੱਟ ਤੋ ਘੱਟ ______ ਵਾਰ ਲਾਗੂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ ?
1.2	do-while ਸਟੇਟਮੈਂਟ (do while statement) ਦਾ ਆਮ ਰੂਪ ਹੈ ....?
1.3	Do while loop ਨੂੰ ________ ਵੀ ਕਿਹਾ ਜਾ ਸਕਦਾ ਹੈ ?

ਪ੍ਰਸ਼ਨ 2: ਦੱਸੋ ਸਹੀ ਹੈ ਜਾਂ ਗਲਤ ?

1.	While do loop ਵਿਚ expression ਪਹਿਲਾਂ ਚਲਦਾ ਹੈ ?
2.	Do while loop ਵਿਚ block of statements ਪਹਿਲਾਂ ਚਲਦਾ ਹੈ ?
3.	ਕਿ while (i=20) {----} ਸਟੇਟਮੈਂਟ infinite loop ਵਿਚ ਪਰਵਰਤਿਤ ਹੋ ਜਾਏਗੀ ?

ਪ੍ਰਸ਼ਨ 3: ਸੰਖੇਪ ਉੱਤਰ ਪ੍ਰਕਾਰ ਦੇ ਪ੍ਰਸ਼ਨ

1.	For statement ਐਕਸਪ੍ਰੈਸ਼ਨ (expression) ਦੇ ਤਿੰਨ ਵੱਖ-ਵੱਖ ਢੰਗ ਕੀ ਹਨ ?
2.	ਜਦੋਂ ਲੂਪ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਬਾਡੀ do while ਦੇ ਵਿਚ ਦੀ ਘੱਟ ਘੱਟ ਗਿਣਤੀ ਕਿੰਨੀ ਹੁੰਦੀ ਹੈ ?
3.	Loop ਵਿਚ null ਸਟੈਟਮੈਂਟ ਨੂੰ ਕਿਵੇਂ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ ?
4.	while loop ਦਾ ਕੰਮ ਕਰਨਾ ਕਿਵੇਂ ਅੰਤ ਹੁੰਦਾ ਹੈ ?
5.	

ਪ੍ਰਸ਼ਨ 4: ਨਿਮਨਲਿਖਿਤ ਦੇ ਪਰਿਣਾਮਾਂ ਦੀ ਪਰਖ ਲਈ ਪ੍ਰੋਗਰਾਮ (1 ਤੋ 5) ਤਕ ਮੁਕੰਮਲ ਕਰੋ ?

i)

```
for ( a =1;a <=6; a++)
  {     if (a %2)
          continue;
        else
          printf("%d\n", a);
          printf("end of loop\n");
  }
```

Output:

```
2
end of loop
4
end of loop
6
end of loop
```

ii)

```
a= 0;
while ( a< 5)
{   if ( a <2) {
 a += 2;
 continue;
 }
else
 printf { "%d\n", ++a);
 printf("end of loop\n");}
```

Output:

```
3
end of loop
4
end of loop
5
end of loop
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

iii)

```
a =0
do {
   if ( a < 3) {
      a += 2;
      printf("%d\n",a);
      continue; }
   else {
      printf( "%d\n " ,++a);
      break; }
   } While ( a<5);
```

Output:

2
4
5

v)

```
int x;
x =7;
While (x > =0)
x = x-2;
printf("%d\n", x);
```

Output

-1

iv)

```
int x;
x = 7;
while (x >= 0)
{ x = x -2;
printf("%d\n", x); }
```

Output

7
5
3
1

ਪ੍ਰਸ਼ਨ 6: ਨਿਮਨਲਿਖਿਤ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਸੰਪੂਰਨ ਕਰੋ ਅਤੇ ਪਰਿਣਾਮ ਪਤਾ ਕਰੋ ?

i)

```
main ( )
  {
  int a ;
  for(a = 0; a < 10; a++)
  {
     ;
     } }
```

ii)

```
int x;
x =7;
While (x > =0)
x = x-2;
printf("%d\n", x);
```

Output

-1

iii

```
main( )
{
 int j;
 j = 0;
while (++j < 10)
printf("GOOD\n") ;
printf("j=%d\n", j);
}
```

output

GOOD
GOOD
GOOD?
GOOD
GOOD
GOOD
GOOD
GOOD
GOOD
J =10

iv

```
main( )
{
 int a,b,c ;
 b = 10;
  while (b < 100)
{   a = b++;
   c = ++b }
   printf("a = %d\n", a);
   printf("a = %d\n", b);
   printf("c = %d\n", c);}
```

output

a = 98
b = 100
c = 100

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

v) **Using For loop** : To increment or decrement a variable by more than one number.(correct the program so that the output is as shown)

```
#include <stdio.h>
main( )
{
  int num;
  for(num =1;num<=100;num+=20)
/*increment*/
   {
    printf("%d\n", num);
  printf("\n");}
  for (num =100;num>=1;num-=20) /*
decrement*/
  printf("%d\n",num);
}
```

Output : 1 21 41 61 81 100 80 60 40 20

vi) **Use of commas** in the for loop (multiple initialization with comma expressions) ,correct the program for given output

```
#include <stdio.h>
 main( )
{
  int num, total, square, cube;
  for (total =0,square =0,cube=0 ,num
=1;num<=5;++num)
   {
     total = total + num ;
     square = square + num*num;
     Cube = cube + num*num*num;
}
 printf("Total          = %d\n", total);
 printf("Square        = %d\n", square);
 printf(("Cube          =  %d\n", cube); }
}
```

Output
Total = 15
Square = 55
cube = 225

Answer to objective questions

Question	.1	.2	.3
1	1	do statement while (expression);	post test loop
2	T	T	T

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ

7

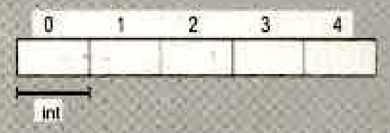
ਐਰੇਸ (ਭਾਗ–1)
ARRAYS (Part – 1)

7.0.0 ਭੂਮਿਕਾ (Introduction)

ਐਰੇ : ਇਕ ਤਰਤੀਬਵਾਰ ਐਲੀਮੈਂਟਸ (elements) ਦਾ ਸਮੂਹ ਹੈ ਜੋ ਇਕੋ ਨਾਮ ਨੂੰ ਸਾਂਝੇ ਵਰਤਦੇ ਹਨ।

ਐਰੇ (array) ਡਾਟਾ ਟਾਈਪਸ ਹੁੰਦੀਆਂ ਹਨ ਅਤੇ ਇਹ ਸਾਨੂੰ ਇਕੋ ਡਾਟਾ ਟਾਈਪ ਦੇ ਨਾਲ ਅਨੇਕਾਂ ਮੁੱਲ ਸਟੋਰ ਕਰਨ ਲਈ ਇਕਹਰੇ ਵੇਰੀਏਬਲ (variable) ਵਰਤਣ ਦੀ ਆਗਿਆ ਦਿੰਦੀਆਂ ਹਨ। ਐਰੇ (array) ਦੇ ਹਰ ਇਕ ਨਿਰਮਿਤ ਮੁੱਲ ਨੂੰ ਐਲੀਮੈਂਟ (element) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਐਰੇ (array) ਦੇ ਹਰ ਇਕ ਐਲੀਮੈਂਟ (element) ਦਾ ਆਪਣਾ ਹੀ ਸਟੋਰੇਜ ਸਥਾਨ ਹੁੰਦਾ ਹੈ, ਜਿਸਦਾ ਕਿਸੇ ਵੀ ਵੇਰੀਏਬਲ (variable) ਦੇ ਨਾਮ ਨਾਲ ਹਵਾਲਾ ਦੇ ਕੇ ਉਸ ਸਥਾਨ ਤਕ ਪਹੁੰਚਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਐਰੇ (array) ਦੇ ਐਲੀਮੈਂਟ (element) ਮੈਮੋਰੀ ਵਿਚ ਲਗਾਤਾਰ ਸਟੋਰੇਜ਼ ਸਥਾਨ ਵਿਚ ਸਟੋਰ ਕੀਤੇ ਹੋਏ ਹੁੰਦੇ ਹਨ ਅਤੇ ਬਰੈਕਟਾਂ ਵਿਚ ਅੰਕਾਂ ਰਾਹੀਂ ਐਰੇ (array) ਦੇ ਨਾਮ ਦੁਆਰਾ ਪਛਾਣੇ ਜਾਂਦੇ ਹਨ। ਇਹ ਅੰਕ 0 ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦੇ ਹਨ ਅਤੇ ਹਰ ਵਾਰ ਇਕ-ਇਕ ਅੰਕ ਨਾਲ ਵਧਦੇ ਹਨ। ਕਹਿ ਲਓ ਕਿ ਹਰ ਇਕ ਅੰਕ 0,1,2,3,4 .... ਨੂੰ ਇਕ ਸੂਚੀ ਦਾ ਸਬਸਕਰਿਪਟ/ਇਨਡੈਕਸ (subscript/index) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਐਰੇ (array) ਦੇ ਵਿਅਕਤੀਗਤ ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਹੇਠਾਂ ਦਿਖਾਏ ਗਏ ਵਜੋਂ ਸਬਸਕਰਿਪਟ ਵੇਰੀਏਬਲ (variable) ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਮਿਸਾਲ : ਚਿੱਤਰ ਵਿਚ ਦਰਸਾਏ ਗਏ Array Age ਨੂੰ 5 ਪੂਰਣ-ਅੰਕ (int age [5]) ਮੁੱਲ ਵਿਚ ਵੰਡਿਆ ਗਿਆ ਹੈ ਜੋ ਇੰਟ ਟਾਈਪ ਦੇ ਹਨ। ਜਿਥੇ ਹਰੇਕ ਖਾਲੀ ਪੈਨਲ ਐਰੇ (array) ਦੇ ਇਕ ਐਲੀਮੈਂਟ (element) ਦੀ ਪ੍ਰਤਿਨਿਧਤਾ ਕਰੇਗਾ ਜੋ ਕਿ ਇਸ ਕੇਸ ਵਿਚ ਟਾਈਪ ਇੰਟ (type Int) ਦਾ ਪੂਰਣ-ਅੰਕ (integer) ਮੁੱਲ ਹੈ। ਇਨ੍ਹਾਂ ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਸੰਖਿਆ 0 ਤੋਂ 4 ਹੁੰਦੀ ਹੈ ਜਿਵੇਂ ਕਿ ਐਰੇ (array) ਵਿਚ ਪਹਿਲੀ ਸੂਚੀ ਹਮੇਸ਼ਾ 0 ਹੁੰਦੀ ਹੈ ਜੋ ਇਸਦੀ ਲੰਬਾਈ ਤੇ ਸੁਤੰਤਰ ਹੁੰਦੀ ਹੈ।



ਜੇਕਰ Age array ਟਾਈਪ ਹੈ ਜਿਸ ਵਿਚ n elements ਹਨ, ਤੇ
Age array ਐਲੀਮੈਂਟਸ ਇਸ ਤਰ੍ਹਾਂ ਹੋਣਗੇ : age [0], age[1], age[2], age[3], age[4]...... age[n]

7.1.0 ਐਰੇ ਦੀ ਡਿਕਲੇਅਰਿੰਗ ਅਤੇ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (Declaring and initialization of array):

ਵੇਰੀਏਬਲ (variable) ਦੀ ਤਰ੍ਹਾਂ ਐਰੇ (array) ਨੂੰ ਇਸ ਦੀ ਵਰਤੋਂ ਤੋਂ ਪਹਿਲਾਂ ਡਿਕਲੇਅਰ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।

```
ਡਾਟਾ_ਟਾਈਪ     ਐਰੇ_ਨਾਮ [ਐਲੀਮੈਂਟਸ ਦੇ ਨੰਬਰ ਐਰੇਸ ਵਿਚ]
Data_type     array_name [number of elements in array]
```

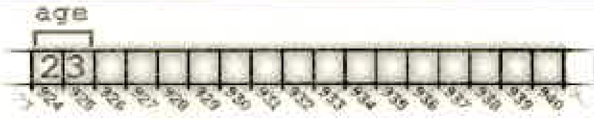
ਜਿਥੇ ਡਾਟਾ ਟਾਈਪ (data type), ਇਕ ਵੇਲਿਡ-ਟਾਈਪ (valid type) ਹੈ (ਜਿਵੇਂ int, float...), ਨੇਮ ਵੇਲਿਡ-ਆਈਡੇਂਟੀਫਾਈਰ (valid identifier) ਹੈ ਅਤੇ ਐਲੀਮੈਂਟ (element) ਖੇਤਰ (field) ਹੈ (ਜੋ ਹਮੇਸ਼ਾ ਚੌਰਸ ਬਰੈਕਟਾਂ [ ] ਵਿਚ ਹੁੰਦੇ ਹਨ) ਨਿਸ਼ਚਿਤ ਕਰਦੀ ਹੈ ਕਿ ਐਰੇ (array) ਵਿਚ ਇਨ੍ਹਾਂ ਐਲੀਮੈਂਟਸ (elements) ਚੋਂ ਕਿੰਨੇ ਸ਼ਾਮਲ ਕੀਤੇ ਜਾਣੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਆਓ ਇਕ ਵਿਅਕਤੀ ਦੀ age ਸਟੋਰ ਕਰਨ ਲਈ ਇਕਹਰੇ ਵੇਰੀਏਬਲ (variable) ਤੇ ਗੌਰ ਕਰੀਏ।

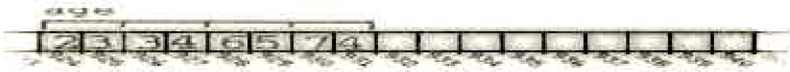
```
#include <stdio.h>

 int main()
 {
  short age;
  age=23;
  printf("%d\n", age);
  return 0;
 }
```



ਰੇਖਾ 5 ਤੇ age variable ਬਣਾਇਆ ਗਿਆ ਹੈ ਜੋ short int type ਦਾ ਹੈ ਅਤੇ ਮੁੱਲ age ਨੂੰ ਸਮਰਪਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਅੰਤ ਵਿਚ, age ਸਕਰੀਨ ਤੇ ਪ੍ਰਿੰਟ ਹੋ ਜਾਂਦੀ ਹੈ।

ਆਓ ਹੁਣ ਕੇਵਲ ਇਕ ਦੀ ਬਜਾਏ 4 ਉਮਰਾਂ ਦੇ ਮਾਰਗ ਤੇ ਗੌਰ ਕਰੀਏ। ਇਸ ਲਈ 4 ਵਖਰੇ ਵੇਰੀਏਬਲਜ (variables) ਪੈਦਾ ਕਰ ਸਕਦੇ ਹਾਂ ਪਰੰਤੂ 4 ਵਖਰੇ ਵੇਰੀਏਬਲਜ (variables) ਦੀ ਸੀਮਿਤ ਵਰਤੋਂ ਹੁੰਦੀ ਹੈ। 4 ਵਖਰੇ ਵੇਰੀਏਬਲ (variable) ਇਸਤੇਮਾਲ ਕਰਨ ਦੀ ਬਜਾਏ, ਅਸੀ ਐਰੇ (array) ਹੀ ਵਰਤਾਂਗੇ । ਆਓ ਤੁਹਾਨੂੰ ਦਸੀਏ ਕਿ ਐਰੇ (array) ਕਿਵੇ ਬਣਾਉਣੀ ਹੈ ਅਤੇ ਐਰੇ (array) ਨੂੰ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਦਾ ਢੰਗ ਕੀ ਹੈ:

```
1: #include <stdio.h>
2:
3: int main()
4: {
5:   short age[4];
6:   age[0]=23;
7:   age[1]=34;
8:   age[2]=65;
9:   age[3]=74;
10:  return 0;
11: }
```



ਰੇਖਾ (5) ਤੇ 4 short data type ਦੇ ਐਰੇ (array) ਡਿਕਲੇਅਰ ਕਿਤੇ ਗਏ ਹਨ। ਰੇਖਾ (6) ਤੋਂ ਰੇਖਾ (9) ਤਕ ਐਰੇ (array) ਵਿਚ ਹਰ ਇਕ ਵੇਰੀਏਬਲ (variable) ਨੂੰ ਮੁੱਲ ਸਮਰਪਣ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਪ੍ਰੋਗਰਾਮ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਐਰੇ (array) ਦਾ ਪਹਿਲਾ ਐਲੀਮੈਂਟ (element) ਪਰਖਣ ਲਈ 0 ਬਜਾਏ ਅੰਕ 1 ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਬਰੈਕਟਾਂ [ ] ਦੇ ਵਿਚ ਐਲੀਮੈਂਟ (element) ਖੇਤਰ (field) ਜਿਹੜੇ ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਗਿਣਤੀ ਦਾ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਕਰਦੇ ਹਨ, ਜੋ ਐਰੇ (array) ਵਿਚ ਰਖੇ ਜਾਣੇ ਹੁੰਦੇ ਹਨ, ਕਾਨਸਟੈਂਟ ਮੁੱਲ (constant value) ਦੇ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ।

The following assignment statements are valid in arrays

a) x[0]= 15; c.) x[5]+ = 1; e) char c[2]; g) b = age[a+2];
b) x[6]= x[6]+6+ x[5] ; d.) x[5] = x[12]+ 6 ; f) float b[4]; h) age[age[a]] = age[2] + 5

7.1.1 ਇਨਿਸ਼ਿਯਲਾਈਜ਼ਿੰਗ ਐਰੇਸ (Initializing arrays):

ਇਨਿਸ਼ਿਯਲਾਈਜ਼ (initialize) ਕਰਨਾ ਕੁੱਝ ਹੋਰ ਨਹੀ ਹੁੰਦਾ ਬਲਕਿ ਵੇਰੀਏਬਲ (variable) ਨੂੰ ਕੁਝ ਮੁੱਲ ਸਮਰਪਣ ਕਰਨੇ ਹੁੰਦੇ ਹਨ ਜਿਹੜੇ ਪ੍ਰੋਸੈਸਿੰਗ ਕਰਦੇ ਹਨ। ਸਾਧਾਰਣ ਵੇਰੀਏਬਲ (variable) ਦੇ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਦੀ ਤਰ੍ਹਾਂ ਐਰੇ (array) ਦੇ ਵਿਅਕਤੀਗਤ ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ (initialize) ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਸਮੂਹ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ਡ (initialized) ਮੁੱਲ (value) ਨਿਰੰਤਰ (continues) ਜਾਂ ਕਾਨਸਟੈਂਟ (constant) ਹੋਣੇ ਚਾਹਿਦੇ ਹਨ। ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization), ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਦੇ ਸਮੇਂ ਦੌਰਾਨ ਸਮੂਹ ਐਰੇ (array) ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਵੀ ਕਿਤੀ ਜਾ ਸਕਦੀ ਹੈ।

Data type array_name[size] = {element1, element2, element3, element4, ... element n}

ਮਿਸਾਲ : **int age [5] = { 16, 2, 77, 40, 71 };**
ਇਹ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਐਰੇ (array) ਇਸ ਤਰਾਂ ਹੋਵੇਗਾ :-

	0	1	2	3	4
Age	16	2	77	40	71

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

7.2.0 ਕੁਝ ਵਿਸ਼ੇਸ਼ ਨਿਯਮ (Some special rules):

1. ਇਨਿਸ਼ਿਯਲਾਈਜ਼ਰ (initializer) ਦੀ ਸੰਖਿਆ ਐਰੇ (array) ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਸੰਖਿਆ ਤੋਂ ਘਟ ਹੋ ਸਕਦੀ ਹੈ।

```
int ary[5] = {2,5,8} ;
```

/* ਇਥੇ ਕੇਵਲ ਐਰੇ ਦੇ ਪਹਿਲੇ 3 ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਹੀ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ ਕਿਤਾ ਗਿਆ ਹੈ */

2. ਇਨਿਸ਼ਿਯਲਾਈਜ਼ਰਸ (initilizers) ਦੀ ਸੰਖਿਆ ਜੇਕਰ ਐਲੀਮੈਂਟਸ (elements) ਤੋਂ ਵੱਧ ਹੋਵੇ ਤਾਂ ਪਰਿਣਾਮ ਗਲਤ ਹੁੰਦਾ ਹੈ।

3. **int limit [ ] = {0,1,2,3,8};** ਉਹੋ ਹੀ ਹੈ ਜੋ ਕੀ **int limit[4] = {0,1,28};** ਹੈ

ਇਥੇ ਐਰੇ (array) ਦੇ ਆਕਾਰ ਨੂੰ ਕੋਈ ਸੰਖਿਆ ਸਪਲਾਈ ਨਹੀ ਕੀਤੀ ਜਾਂਦੀ ਇਸ ਮਾਮਲੇ ਵਿਚ ਕੰਪਾਈਲਰ (compiler) ਵਲੋ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਸੂਚੀ ਵਿਚ ਤਿੰਨ ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਸੰਖਿਆ ਗਿਣ ਲਈ ਜਾਂਦੀ ਹੈ। ਅਤੇ ਇਸ ਨੂੰ ਐਰੇ (array) ਆਕਾਰ ਨਿਸਚਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ

ਮਿਸਾਲ : ਇਹ ਪ੍ਰੋਗਰਾਮ ਸਟੇਟਮੈਂਟ (statement) ਲਈ ਐਰੇ (array) ਦੇ ਉੱਤਰਵਰਤੀ ਐਲੀਮੈਂਟਸ (elements) ਨਾਲ ਚਾਰ ਪੂਰਣ ਅੰਕਾਂ ਦੇ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।

```
#include <stdio.h>
void main()
{
 int marks [ ] = {10,25,20,35} ;
 int x ;
 printf( "Element of the array are\n");
 for (x = 0 ; x < 4 ; x++)
 printf("marks [%d]  =%d\n " , x ,marks[x] ) ;
}                              /* End of main */
```

Output :

```
Element of the array are
marks [0] = 10
marks [1] = 25
marks [2] = 20
marks [3] = 35
```

Problem: What does this loop do?

```
for(x = 0 ; x <3 ;x++)
num[x] = x * 3;
```

ਇਹ ਹਰ ਇਕ ਐਰੇ (array) ਦੇ ਐਲੀਮੈਂਟ ਨੂੰ num ਮੁੱਲ ਸਮਰਪਣ ਕਰਦਾ ਹੈ ਜੋ ਇਸ ਦੇ ਐਲੀਮੈਂਟ ਦਾ ਹਰ ਵਾਰ 3 ਗੁਣਾ ਹੋਏਗਾ। ਜੋ ਨਿਮਨਲਿਖਿਤ ਅਨੁਸਾਰ ਹੈ :-

```
num [0] gets the value 0
num [1] gets the value 3
num [2] gets the value 6
```

7.2.1 ਐਰੇ ਵਿਚ ਡਾਟਾ ਪ੍ਰਵੇਸ਼ ਕਰਨਾ (Entering Data into an Array)

ਕਿਉਂ ਜੋ ਐਰੇ (array) ਮੁਲਾਂ (values) ਦਾ ਸੰਗ੍ਰਹਿ ਕਰਦੀ ਹੈ, ਇਸ ਲਈ ਐਰੇ (array) ਦੇ ਵਿਅਕਤੀਗਤ ਐਲੀਮੈਂਟਸ (elements) ਵਿਚ ਡਾਟਾ ਇਨਪੁਟ (input) ਕਰਨ ਲਈ ਅਸੀ ਲੂਪ ਸਟੇਟਮੈਂਟਸ (loop statements) ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਜੇਕਰ ਐਰੇ (array) ਦੇ ਅਸਲ ਆਕਾਰ ਦਾ ਪਹਿਲਾਂ ਹੀ ਪਤਾ ਲਗ ਜਾਵੇ ਤਾਂ while loop ਜਾਂ do-while-loop ਦੀ ਜਗ੍ਹਾ for loop ਨੂੰ ਪਹਿਲ ਦਿਤੀ ਜਾਂਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਕੋਡ ਦਾ ਉਹ ਭਾਗ ਜੋ ਡਾਟਾ ਨੂੰ ਐਰੇ (array) ਵਿਚ ਰਖਦਾ ਹੈ

```
#include <stdio.h>
#include <conio.h>
void main( )
{
clrscr();
int i;
printf("Enter marks for 5 students:");
for(i=0;i<5;i++)
 {
   printf("\nMarks of student %d;", i+1);
   scanf("%d",&marks[i]);
 }
}
```

Output:

```
Enter marks for 5 students
Marks of student 1: 45  (press enter)
Marks of student 2: 49  (press enter)
Marks of student 3: 67  (press enter)
Marks of student 4: 98  (press enter)
Marks of student 5: 76  (press enter)
```

ਇਥੇ for loop, data ਨੂੰ ਐਂਟਰ ਕਰਨ ਲਈ 5 ਵਾਰ ਪੁਛੇਗਾ ਅਤੇ 5 ਵਾਰ ਹੀ ਐਂਟਰ ਕਰਨ ਲਈ ਕਹੇਗਾ। ਇਸ ਲਈ scanf() ਸਟੇਟਮੈਂਟ ਮੁੱਲ ਨੂੰ ਐਰੇ ਐਲੀਮੈਂਟ marks[0] ਵਿਚ ਰਖੇਗੀ ਜੋ ਕਿ ਐਰੇ (array) ਦਾ ਪਹਿਲਾ ਐਲੀਮੈਂਟ (element) ਹੈ। ਲੂਪ (loop) ਉਨੀ ਦੇਰ ਲਾਗੂ ਕਰਨ ਲਈ ਜਾਰੀ ਰਹੇਗਾ ਜਦ ਤਕ i ਸਟੇਟਮੈਂਟ (statement) **scan**() ਵਿਚ 4 (i<5) ਨਹੀ ਬਣ ਜਾਂਦਾ ।

scanf() ਸਟੇਟਮੈਂਟ ਵਿਚ ਐਰੇ marks[i] ਦੇ ਨਾਲ address of the operator (&) ਦਾ ਇਸਤੇਮਾਲ ਕੀਤਾ ਗਿਆ ਹੈ। ਇਸ ਤਰ੍ਹਾਂ ਅਸੀ scanf() ਫੰਕਸ਼ਨ ਨੂੰ ਐਰੇ ਦੇ ਐਲੀਮੈਂਟ ਦਾ ਐਡਰੇਸ ਪਾਸ ਕਰ ਰਹੇ ਹਾਂ ਨਾ ਕਿ ਉਸਦਾ ਮੁੱਲ।

clrscr( ); ਇਹ ਸਕਰੀਨ ਤੇ ਲਿਖੇ ਡਾਟਾ ਨੂੰ ਸਾਫ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਇਸ ਲਈ conio.h ਨੂੰ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

ਸਾਰੇ ਮੁਲਾਂ (values) ਦੀ ਅਲਗ-ਅਲਗ ਪ੍ਰਿੰਟਿਗ ਲਈ, ਕੋਸ਼ਿਸ਼ ਕਰੋ :

```
#include <stdio.h>

int main()
{
  short age[4];
  age[0]=23;
  age[1]=34;
  age[2]=65;
  age[3]=74;
  printf("%d\n", age[0]);
  printf("%d\n", age[1]);
  printf("%d\n", age[2]);
  printf("%d\n", age[3]);
  return 0;
}
```

ਇਹ ਪ੍ਰੋਗਰਾਮ ਦੀਆ ਦਿਤੀਆ ਗਈਆਂ ਕੀਮਤਾਂ (values) 23,34,65,74 ਨੂੰ ਐਰੇ (array) age ਦੁਆਰਾ ਸਟੋਰ ਕਰੇਗਾ ਅਤੇ ਇਹ ਸਕਰੀਨ ਤੇ ਹੇਠਾਂ ਵਾਂਗ ਦਿਖਾਈ ਦੇਣਗੀਆਂ :

```
23
34
65
74
```

ਟਿੱਪਣੀ: ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਕੋਈ ਵੀ ਅਜਿਹਾ ਇਕਲੀ ਸਟੇਟਮੈਂਟ ਨਹੀ ਹੈ ਜੋ ਇਹ ਕਹੇ ਕਿ 'print an entire array to the screen' (ਸਕਰੀਨ ਤੇ ਸਾਰੇ ਐਰੇ (array) ਪ੍ਰਿੰਟ ਕਰੋ) ਐਰੇ (array) ਵਿਚ ਹਰੇਕ ਐਲੀਮੈਂਟ (element) ਸਕਰੀਨ ਤੇ ਵਿਅਕਤੀਗਤ ਤੋਰ ਤੇ ਪ੍ਰਿੰਟ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

7.3.0 ਐਰੇ ਕਾਪੀ ਕਰਨੀ (Copying arrays)

ਆਓ ਐਰੇਸ (arrays) ਦੀ ਕਾਪੀ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕਰੀਏ ਇਕ ਸਮਾਨ ਤਕਨੀਕ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਜੋ ਐਰੇਸ (ਜੋ ਕਿ ਇਕ ਸਮੇਂ ਇਕ ਐਲੀਮੈਂਟ (element) ਹੈ) ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

```
1: #include <stdio.h>
2:
3: int main()
4: {
5:    short age[4];
6:    short same_age[4];
7:
8:    age[0]=23;
9:    age[1]=34;
10:   age[2]=65;
11:   age[3]=74;
12:
13:   same_age[0]=age[0];
14:   same_age[1]=age[1];
15:   same_age[2]=age[2];
16:   same_age[3]=age[3];
17:
18:   printf("%d\n", same_age[0]);
19:   printf("%d\n", same_age[1]);
20:   printf("%d\n", same_age[2]);
21:   printf("%d\n", same_age[3]);
22:   return 0;
23: }
```

ਇਸ ਮਿਸਾਲ ਵਿਚ ਐਰੇ (array) ਕਾਪੀ ਕਰਨ ਦੀ ਤਕਨੀਕ ਵਧੀਆ ਕੰਮ ਕਰਦੀ ਹੈ। ਦੋ ਐਰੇਸ age ਅਤੇ same_age ਡਿਕਲੇਅਰ ਕੀਤੀਆਂ ਗਈਆਂ ਹਨ। age array ਦੇ ਹਰ ਇਕ ਐਲੀਮੈਂਟ (element) ਨੂੰ ਇਕ ਮੁੱਲ ਸਮਰਪਣ ਕੀਤਾ ਗਿਆ ਹੈ। ਹੁਣ ਜੇ ਕਰ age array ਚਾਰ ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ same_age array ਉਪਰ ਕਾਪੀ ਕਰਨਾ ਹੋਵੇ ਤਾਂ ਅਸੀਂ ਇਹ ਕੰਮ ਐਲੀਮੈਂਟ-ਤੋਂ-ਐਲੀਮੈਂਟ (element by element) ਦੁਆਰਾ ਕਰਾਂਗੇ । ਮਿਸਾਲ ਲਈ :

copy first element of the array

copy second element of the array

ਟਿੱਪਣੀ : ਪ੍ਰਿੰਟ ਹੋਏ ਐਰੇ ਦੀ ਤਰ੍ਹਾਂ ਸੀ (C) ਭਾਸ਼ਾ ਵਿਚ ਕੋਈ ਅਜਿਹੀ ਇਕਲੀ ਸਟੇਟਮੇਂਟ ਨਹੀ ਜੋ ਕਹੇ ਕਿ ਸਮੂਚੀ ਐਰੇ ਨੂੰ ਕਿਸੇ ਹੋਰ ਐਰੇ ਤੇ ਕਾਪੀ ਕਰੋ ("copy an entire array into another array") ਇਸ ਲਈ ਐਰੇ ਐਲੀਮੈਂਟ (element) ਵਿਅਕਤੀਗਤ ਤੌਰ ਤੇ ਕਾਪੀ ਕੀਤੇ ਜਾਣੇ ਚਾਹੀਦੇ ਹਨ।

ਇਸ ਲਈ ਇਕ ਐਰੇ (array) ਨੂੰ ਕਿਸੇ ਹੋਰ ਵਿਚ ਕਾਪੀ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਤਕਨੀਕ ਬਿਲਕੁਲ ਉਹੀ ਹੁੰਦੀ ਹੈ ਜੋ ਅਸੀਂ 4 ਵਖਰੇ ਵੇਰੀਏਬਲਜ (variables) ਨੂੰ 4 ਹੋਰ ਵੇਰੀਏਬਲਜ ਵਿਚ ਕਾਪੀ ਕਰਨ ਲਈ ਵਰਤਦੇ ਹਾਂ। ਇਸ ਲਈ ਵਖਰੇ ਵੇਰੀਏਬਲਜ (variables) ਦੀ ਜਗ੍ਹਾ ਐਰੇ (array) ਵਰਤਣ ਦਾ ਲਾਭ ਕੀ ਹੁੰਦਾ ਹੈ ?

ਐਰੇ (array) ਦਾ ਵਖਰੇ ਵੇਰੀਏਬਲਜ (variables) ਉਪਰ ਇਕ ਮਹੱਤਵ ਪੂਰਨ ਲਾਭ ਉਸਦਾ ਨਾਮ ਹੁੰਦਾ ਹੈ। ਸਾਡੀ ਮਿਸਾਲ ਵਿਚ ਚਾਰ ਵਖਰੇ ਵੇਰੀਏਬਲਜ ਦੀ ਵਰਤੋਂ ਲਈ 4 ਖਾਸ ਨਾਵਾਂ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਇਸ ਐਰੇ ਵਿਚ 4 short ਵੇਰੀਏਬਲਜ ਦਾ ਨਾਮ ਅਤੇ age ਉਹੀ ਹੈ। ਲਾਈਨ (5) ਅਤੇ ਲਾਈਨ (6) ਦੇ ਉਪਰ ਦੋ ਐਰੇ ਵੇਰੀਏਬਲਜ short ਟਾਈਪ ਦੇ age ਅਤੇ same_age ਦਰਸਾਏ ਗਏ ਹਨ। ਇਸ ਐਰੇ ਦੇ 4 ਵੇਰੀਏਬਲਜ short ਟਾਈਪ ਇਕੋ ਹੀ ਤਰ੍ਹਾਂ ਦੇ ਹਨ, ਪਰ ਇਹਨਾਂ ਦੋਵੇਂ ਐਰੇ ਦੀ ਸਬਸਕਰਿਪਟ (subscript) ਜਾਂ ਸੂਚੀ (index) ਅਲੱਗ-ਅਲੱਗ ਹੁੰਦੀ ਹੈ। ਇਹ ਵੱਖਰਤਾ ਸਾਨੂੰ ਸਾਡੇ ਕੋਡ ਨੂੰ ਇਸ ਤਰ੍ਹਾਂ ਸੰਖੇਪ ਕਰਨ ਦੀ ਆਗਿਆ ਦਿੰਦੀ ਹੈ ਜੋ ਕਿ 4 ਵੇਰੀਏਬਲਜ (variables) ਦੇ ਨਾਲ ਹਰ ਇਕ ਦੇ ਖਾਸ ਨਾਮ ਨਾਲ ਮੁਸ਼ਕਲ ਹੋਵੇਗਾ। ਕਿਉਂ ਜੋ ਇਹਨਾਂ 4 ਐਰਜ਼ (arrays) ਦੇ ਵਿਚ ਉਹਨਾਂ ਦਾ ਸਬਸਕਰਿਪਟ (subscript) ਦਾ ਹੀ ਫਰਕ ਹੈ ਇਸ ਲਈ ਇਕ ਲੂਪ (loop) ਅਤੇ ਇਕ ਕਾਊਂਟਰ ਸਮੂਹ ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਵਧੇਰੇ ਆਰਾਮ ਨਾਲ ਕਾਪੀ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਇਹੀ ਤਕਨੀਕ ਉਸ ਕੋਡ ਨੂੰ ਸੰਖੇਪ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਜਿਹੜੀ ਐਰੇ ਨੂੰ ਸਕਰੀਨ ਤੇ ਛਾਪਦੀ ਹੈ।

```
1: #include <stdio.h>
2:
3: int main()
4: {
5:    short age[4];
6:    short same_age[4];
7:    int i, j;
8:    age[0]=23;
9:    age[1]=34;
10:   age[2]=65;
11:   age[3]=74;
12:
13:   for(i=0; i<4; i++)
14:   same_age[i]=age[i];
15:
16:   for(j=0; j<4; j++)
17:   printf("%d\n", same_age[j]);
18:   return 0;
19: }
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

7.4.0 ਐਰੇ ਦੇ ਮੁੱਲਾਂ ਦੀ ਪਹੁੰਚ-ਕਰਨੀ (Accessing the values of an array):

ਇਕ ਪ੍ਰੋਗਰਾਮ ਦੇ ਕਿਸੇ ਵੀ ਬਿੰਦੂ ਤੇ ਜਿਥੇ ਐਰੇ ਪ੍ਰਤੱਖ (visible) ਹੁੰਦਾ ਹੈ ਅਸੀ ਇਨ੍ਹਾਂ ਦੇ ਵਿਅਕਤੀਗਤ ਐਲੀਮੈਂਟਸ (elements) ਵਿਚੋਂ ਕਿਸੇ ਦੇ ਵੀ ਮੁੱਲ ਤਕ ਪਹੁੰਚ (access) ਕਰ ਸਕਦੇ ਹਾਂ ਜਿਵੇਂ ਕਿ ਇਹ ਆਮ ਵੇਰੀਏਬਲ (variable) ਹੁੰਦਾ ਹੈ। ਇਸ ਪ੍ਰਕਾਰ ਪੜ੍ਹਣ ਅਤੇ ਇਸਦੇ ਮੁੱਲ ਨੂੰ ਸੋਧਣ ਦੇ ਕਾਬਿਲ ਹੋ ਜਾਂਦੇ ਹਾਂ। ਇਸ ਦੀ ਬਣਤਰ ਬਹੁਤ ਸਧਾਰਣ ਹੁੰਦੀ ਹੈ।

```
name[index];
```

ਪੂਰਵ ਮਿਸਾਲਾਂ ਨੂੰ ਅਪਣਾਉਂਦੇ ਹੋਏ ਜਿਸ ਵਿਚ age ਦੇ 5 ਐਲੀਮੈਂਟ (element) ਸਨ ਅਤੇ ਉਨ੍ਹਾਂ ਐਲੀਮੈਂਟਸ (elements) ਵਿਚੋਂ ਹਰ ਇਕ int ਟਾਈਪ ਦਾ ਸੀ, ਹਰ ਇਕ ਐਲੀਮੈਂਟ (element) ਦੇ ਹਵਾਲੇ ਲਈ ਅਸੀਂ ਜਿਹੜਾ ਨਾਮ ਵਰਤ ਸਕਦੇ ਹਾਂ ਉਹ ਨਿਮਨ ਅਨੁਸਾਰ ਹੁੰਦਾ ਹੈ:

Array name age	age[0]	age[1]	age[2]	age[3]	age[4]

ਮਿਸਾਲ, age ਦੇ ਤੀਜੇ ਐਲੀਮੈਂਟ (element) ਵਿਚ ਮੁਲ 75 ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਅਸੀਂ ਨਿਮਨਲਿਖਤ ਸਟੇਟਮੈਂਟ (statement) ਲਿਖ ਸਕਦੇ ਹਾਂ; **age[2] = 75;**

ਅਤੇ, ਵੇਰੀਏਬਲ (variable) age ਦੇ ਤੀਜੇ ਐਲੀਮੈਂਟ (element) ਦੇ ਮੁੱਲ ਨੂੰ ਵੇਰੀਏਬਲ a ਤਕ ਭੇਜਣ ਲਈ ਅਸੀ ਲਿਖ ਸਕਦੇ ਹਾਂ **a=age[2];**

ਇਸ ਲਈ, ਸਮੂਹ ਮੰਤਵਾਂ ਲਈ ਐਕਸਪ੍ਰੇਸ਼ਨ (expression), age[2] ਟਾਈਪ int ਦੇ ਵੇਰੀਏਬਲ (variable) ਸਮਾਨ ਹੈ।

ਸੂਚਨਾ ਕਿ, age ਦਾ ਤੀਜਾ ਐਲੀਮੈਂਟ (element) ਨਿਸ਼ਚਿਤ age[2] ਹੁੰਦਾ ਹੈ, ਜਿਵੇਂ ਕਿ ਪਹਿਲੀ age[0] ਅਤੇ ਦੂਜੀ age[1] ਹੈ ਅਤੇ ਇਸ ਲਈ ਤੀਜੀ age[2] ਹੈ। ਉਸੇ ਕਾਰਨ ਨਾਲ, ਅੰਤਿਮ ਐਲੀਮੈਂਟ (element) age[4] ਹੈ। ਇਸ ਲਈ ਜੇਕਰ ਅਸੀ age[5] ਲਿਖਦੇ ਹਾਂ ਤਾਂ ਅਸੀਂ age ਦੇ ਛੇਵੇਂ ਐਲੀਮੈਂਟ (element) ਤਕ ਪਹੁੰਚ ਰਹੇ ਹੋਵਾਂਗੇ ਅਤੇ ਇਸ ਲਈ ਐਰੇ (array) ਦੇ ਆਕਾਰ ਤੋਂ ਅੱਗੇ ਵਧ ਜਾਵਾਂਗੇ। ਐਰੇ (array) ਐਲੀਮੈਂਟਸ ਤਕ ਐਰੇ (array) ਦਾ ਨਾਮ ਲਿਖ ਕੇ ਅਤੇ ਐਰੇ (array) ਵਿਚ ਮੱਦ (item) ਦੀ ਸੂਚੀ ਦੀ ਸਥਿਤੀ ਤਕ ਪਹੁੰਚਿਆ ਜਾ ਸਕਦਾ ਹੈ।

ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਸੂਚੀ ਵਿਚ ਐਰੇ (array) ਦੇ ਮੁੱਲ ਦੇ ਸਮਾਨ ਹੁੰਦਾ ਹੈ।

```
int data[2];
data[0] = 30;
data[1] = 20;
printf("value at data[1] is %d\n" ,data[1]);
```

ਇਥੇ ਇਹ ਦੱਸਣਾ ਜਰੂਰੀ ਹੈ ਕਿ ਐਰੇ ਦੀ ਬਰੈਕਟਾਂ [ ] ਦੇ ਦੋ ਮੁੱਖ ਕੰਮ ਹਨ। ਪਹਿਲਾ, ਐਰੇ ਦੇ ਆਕਾਰ ਨੂੰ ਦਰਸਾਉਣਾ, ਅਤੇ ਦੂਜਾ ਐਰੇ ਐਲੀਮੈਂਟਸ ਲਈ ਸਬਸਕਰਿਪਟ (subscript) ਜਾਂ ਸੂਚੀ (index) ਨਿਸ਼ਚਿਤ ਕਰਨਾ।

int age[5];	// declaration of a new array
age[2] = 75;	// access to an element of the array

ਹੋਰ ਐਰੇਸ ਦੇ ਕੁਛ ਵੇਲਿਡ (valid) ਓਪਰੇਸ਼ਨ :

```
age[0] = a;
age[a] = 75;
b = age [a+2];
age[age[a]] = age[2] + 5;
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

7.5.0 ਐਰੇ ਐਲੀਮੈਂਟਸ ਦਾ ਪ੍ਰਬੰਧਨ (Manipulation of array elements)

ਸੀ (C) ਵਲੋਂ ਸਮੁੱਚੀ ਐਰੇ (array) ਤੇ ਕਿਸੇ ਵੀ ਕੰਮ ਨੂੰ ਨਿਭਾਉਣ ਲਈ ਸਹਾਇਤਾ ਨਹੀਂ ਦਿਤੀ ਜਾਂਦੀ। ਇਸ ਲਈ ਸਮੁੱਚੀ ਐਰੇ (array) ਨੂੰ ਇਕਲੇ ਐਲੀਮੈਂਟ (element) ਵਜੋਂ ਪ੍ਰਕ੍ਰਿਆ ਵਿਚ ਨਹੀਂ ਲਿਆਂਦਾ ਜਾ ਸਕਦਾ। ਪਰੰਤੂ, ਇਹ ਪ੍ਰੋਗਰਾਮਰ ਨੂੰ Element by Element ਅਧਾਰ ਤੇ ਵਿਸ਼ੇਸ਼ ਕੰਮ ਕਰਨ ਦੀ ਆਗਿਆ ਦੇਂਦਾ ਹੈ।

ਐਰੇ (array) ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਵਰਤੋਂ ਗਣਿਤ ਦੇ ਕੰਮਾਂ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ ਜਿਵੇਂ ਕਿ ਐਰੇ (array) ਐਲੀਮੈਂਟ (element) ਦੇ ਵਿਭਿੰਨ ਮੁੱਲਾਂ ਨੂੰ ਜੋੜ ਕੇ, ਐਰੇ ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਉਪਜ ਲਭ ਕੇ ਇਹ ਕੰਮ ਕਰਨ ਦੇ ਦੋ ਢੰਗ ਹੁੰਦੇ ਹਨ। ਇਕ ਐਰੇ ਦੀ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਲਈ ਦਿਤੇ ਗਏ ਮੁੱਲਾਂ ਦੀ ਵਰਤੋਂ ਕਰ ਕੇ ਅਤੇ ਦੂਜੇ ਯੂਜ਼ਰਸ (ਕੀ-ਬੋਰਡ) ਰਾਹੀਂ ਮੁੱਲ ਸਪਲਾਈ ਕਰਕੇ।

ਇਥੇ array age ਪੰਜ ਮੁੱਲਾਂ ਦੁਆਰਾ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ (initialize) ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਇਹਨਾ ਦਾ ਜੋੜ ਕਰਨ ਲਈ for loop ਨੂੰ ਪੰਜ ਵਾਰ ਅੱਗੇ ਚਲਾਇਆ ਗਿਆ ਹੈ ਅਤੇ ਇਸ ਰਾਹੀਂ age array ਦਾ ਜੋੜ result ਵੇਰੀਏਬਲ ਨੂੰ ਸਮਰਪਨ ਕੀਤਾ ਗਿਆ ਹੈ।

```
// arrays example
#include <stdio.h>

int age[ ] = {16, 2, 77, 40, 10};
int n, result=0;
int main ( )
 {
  for ( n=0 ; n<5 ; n++ )
    {
     result += age[n];
    }
  printf("\n the sum of ages is = %d", result);
  return 0;
 }
```

Output: the sum of ages is =145

ਐਲੀਮੈਂਟਸ ਦਾ ਜੋੜ (Sum of Eiements): ਆਓ ਕੀ-ਬੋਰਡ ਰਾਹੀਂ ਯੂਜ਼ਰ ਦੇ ਜ਼ਰੀਏ ਐਰੇ (array) ਐਲੀਮੈਂਟ (element) ਸਪਲਾਈ ਜਮ੍ਹਾਂ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ਲਿਖੀਏ।

ਪ੍ਰੋਗਰਾਮ 1 : ਐਰੇ ਵਿਚ ਐਲੀਮੈਂਟਸ ਦਾ ਜੋੜ (Sum of the elements in an array)

```
#include<stdio.h>
void main()
{
  int a[5],i, sum=0;        /* input the array*/
  for (i=0;i<5;i++)
    {
    printf("Enter the value of element%d;", i+1);
    scanf("%d",&a[i]);
    }               /* sum all the elements in the array */
 for (i=0;i<5;i++)
 sum=sum+a[i];           /* print the sum */
printf( "The sum of all the elements of the array is %d",sum);
}
```

Output:
Let's say the following numbers are inputted in the array a

Enter the value of element1 : 10
Enter the value of element2: 12
Enter the value of element3: 14
Enter the value of element4: 12
Enter the value of element5: 30
The sum of all the element of the array is 78

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰੋਗਰਾਮ 2: ਇਕ ਐਰੇ ਵਿਚ ਸਮੂਹ ਐਲੀਮੈਂਟਸ ਦਾ ਗੁਣਾਂਕ-ਮੁੱਲ (Product of all the elements in an array)

```
#include<stdio.h>
void main()
{
  int a[10],i, prod=1;                    /* input the array*/
  for (i = 0; < = 9;i++)
    {
      printf("enter the element value");
      scanf("%d",&a[i]);
    }                                      /* sum all the elements in the array */
 for (i = 0; i< = 9;i++)
 prod=prod * a[i];                         /* print the product */
 printf( "The product of all the elements of the array is %d",prod);
}
```

ਪਰਿਣਾਮ (output)

ਆਓ ਮੰਨ ਲਈਏ ਕਿ ਨਿਮਨਲਿਖਿਤ ਨੰਬਰਾਂ ਨੂੰ ਇਕ ਐਰੇ (array) ਵਿਚ ਇਨਪੁਟ ਕੀਤਾ ਗਿਆ ਹੈ ਜਿਵੇਂ 2 2 2 2 2 3 3 3 3 3

ਅੰਤ ਵਿਚ ਸਮੂਹ ਐਲੀਮੈਂਟਸ (elements) ਦੇ **ਗੁਣਾਂਕ-ਮੁੱਲ** ਨੂੰ variable prod ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਹੜੀ ਕਿ 7776 ਦੇ ਬਰਾਬਰ ਹੋਵੇਗੀ।

ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਦਿਖਾਈ ਦੇਣ ਵਾਲਾ ਆਊਟਪੁਟ ਹੋਵੇਗਾ 7776

The product of all the elements of the array is 7776

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰੋਗਰਾਮ 3 : ਦੋ ਇਕ ਡਾਈਮੈਂਨਸ਼ਨਲ ਐਰੇਸ ਵਿਚ ਸੰਬੰਧਤ ਐਲੀਮੈਂਟਸ ਦਾ ਪ੍ਰੋਡਕਟ

Product of the respective elements in two one dimensional arrays

```
#include<stdio.h>
void main()
{
 int a[10],b[10],c[10],i, prod=1;            /* input the a array*/
 for (i = 0;i< = 9;i++)
    {
      printf("enter the element value"};
      scanf("%d",&a[i]);
    }                                         /* input the b array*/
    for (I = 0; i< = 9;i++)
     {
       printf("enter the element value");
       scanf("%d",&b[i]);
     }                                 /* product of all the elements in both the arrays */
for (i = 0; i<= 9;i++)
c[i]= a[i] * b[i];          /* print the array where product of both a & b arrays is stored  i.e. array c*/
printf("The product of both the arrays a and b are stored in c whose values are\n");
for (i =0; i< = 9;i++)
      {
       printf( "%d\t",c[i]);
      }
}
```

ਪਰਿਣਾਮ Output:

ਆਓ ਮੰਨੀਏ ਕਿ ਐਰੇ (array) a ਵਿਚ : 22222 33333 ਅਤੇ ਐਰੇ (array) b ਵਿਚ : 33333 22222 ਇਨਪੁੱਟ (input) ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

ਸਮੂਹ ਐਲੀਮੈਂਟਸ (elements) ਦਾ ਆਊਟਪੁਟ ਐਰੇ (array) ਸੀ (C) ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਜਾਵੇਗਾ।

ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਨਜ਼ਰ ਆਉਣ ਵਾਲਾ ਆਊਟਪੁਟ ਹੋਵੇਗਾ।

The product of both the arrays a and b are stored in c whose values are: 6666666666

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰੋਗਰਾਮ 4 : ਐਰੇ ਵਿਚ ਸਮੂਹ ਐਲੀਮੈਂਟਸ ਦਾ ਔਸਤਨ
(Average of all the elements in an array)

```
#include<stdio.h>
void main()
{
 int a[10],i, sum=0,n;
 float avg;
 printf("How many element array is required\n");
 scanf("%d",&n);                                  /* input the array*/
 for (i=0;i<n;i++)
    {
     printf("enter the element value");
     scanf("%d",&a[i]);
    }
for (i=0;i<n;i++)                          /* sum all the elements in the array */
sum=sum+a[i];
avg=sum/n;                                   /*average of all the elements is (sum/n) */
                                             /* print the average */
printf( "The average of all the elements of the array is %d",avg);
 }
```

ਪਰਿਣਾਮ Output:

ਆਓ ਮੰਨ ਲਈਏ ਕਿ ਨਿਮਨਲਿਖਿਤ ਅੰਕ ਇਕ ਐਰੇ (array) a ਵਿਚ ਇਨਪੁਟ (input) ਕੀਤੇ ਜਾਂਦੇ ਹਨ ਮਿਸਾਲ ਲਈ : **1 2 3 4 5 6 7 8 9 10**

ਅੰਤ ਵਿਚ ਸਮੂਹ ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਔਸਤ variable **avg** ਵਿਚ ਸਟੋਰ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜੋ ਕਿ 5.5 ਦੇ ਬਰਾਬਰ ਹੋਵੇਗੀ।

ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਨਜ਼ਰ ਆਉਣ ਵਾਲਾ ਆਊਟਪੁਟ ਹੋਵੇਗਾ।

The average of all the elements of the array is 5.5

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰੋਗਰਾਮ 5: ਲਿਨੀਅਰ ਸਰਚ, ਐਰੇ ਵਿਚ ਮੁੱਲ ਲਈ ਤਲਾਸ਼
(Linear Search, Searching for a value in the array)

```
#include<stdio.h>
void main()
{
 int a[10],i, sum=0,val,c=0;                  /* input the array*/
 for (i=0;i<=9;i++)
     {
      printf("enter the element value");
      scanf("%d",&a[i]);
     }                                        /*input the value to be searched*/
printf("enter the value to be searched");
scanf("%d",&val);                             /*search in all the elements in the array */
i=0;
while (i<=9)
      {
       if (val == a[i])
         {
          c=1;
          break;
         }
        else
           c=0;
       i++;
      }                                       /* print */
if (c==1)
  printf( "The Value found is %d at location a[%d]",val,i+1)
else
  printf("The Value does not exist in the array");
}
```

ਪਰਿਣਾਮ Output:

ਮੰਨ ਲਉ ਕਿ ਇਕ ਐਰੇ (array) a ਵਿਚ ਨਿਮਨਲਿਖਿਤ ਅੰਕ ਇਨਪੁਟ (input) ਕੀਤੇ ਗਏ ਹਨ ਮਿਸਾਲ ਲਈ **1 2 3 4 5 6 7 8 9 10**

ਇਹ ਜਦੋ ਲੱਭੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲ ਨੂ ਪ੍ਰਵੇਸ਼ ਕਰਨ ਲਈ ਕਹੇਗਾ ਤਾਂ ਜੇ ਤੁਸੀ 5 ਪ੍ਰਵੇਸ਼ ਕਰੋਗੇ, ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਦੇਖਿਆ ਜਾਣ ਵਾਲਾ ਆਊਟਪੁਟ ਹੋਵੇਗਾ :

The Value found is 5 at location a[5]

ਪਰੰਤੂ ਇਹ ਜਦੋ ਲੱਭੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲ ਨੂੰ ਪ੍ਰਵੇਸ਼ ਕਰਨ ਲਈ ਕਹੇਗਾ ਅਤੇ ਜੇ ਤੁਸੀ 90 ਪ੍ਰਵੇਸ਼ ਕਰੋਗੇ, ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਦੇਖਿਆ ਜਾਣ ਵਾਲਾ ਆਊਟਪੁਟ ਹੋਵੇਗਾ :

The value does not exist in the array

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰੋਗਰਾਮ 6: ਐਰੇ ਵਿਚ ਉੱਤਮ ਅੰਕ ਲਭਣ ਲਈ
(To find the maximum number in the array)

```
#include<stdio.h>
void main()
{
 int a[10],i, max;              /* input the array*/
 for (i=0;i<=9;i++)
   {
    printf("enter the element value");
    scanf("%d",&a[i]);
   }
max=a[0];                       /* assume that the first element is the maximum */
for (i=1;i<=9;i++)              /* check max with the other elements and if some element
                                   is found greater than the assumed value stored in max,
                                   max is changed to the new value */
   {
    if ( max< a[i])
    max=a[i];
   }                            /* print the maximum number */
printf( "The maximum of all the elements of the array is %d",max);
}
```

ਪਰਿਣਾਮ : Output :

ਮੰਨ ਲਓ ਕਿ ਇਕ ਐਰੇ (array) a ਵਿਚ ਨਿਮਨਲਿਖਿਤ ਅੰਕ ਇਨਪੁਟ (input) ਕੀਤੇ ਜਾਂਦੇ ਹਨ :
1 2 3 4 5 6 7 8 9 10

ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਦੇਖਿਆ ਜਾਣ ਵਾਲਾ ਆਉਟਪੁਟ ਹੋਵੇਗਾ :

The maximum of all the elements of the array is 10

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰੋਗਰਾਮ 7: ਐਰੇ ਵਿਚ ਨਿਊਨਤਮ ਅੰਕ ਲੱਭਣ ਲਈ
(To find the minimum number in the array)

```
#include<stdio.h>
void main()
{
 int a[10],i, min;                    /* input the array*/
 for (i=0;i<=9;i++)
    {
     printf("enter the element value");
     scanf("%d",&a[i]);
    }
min=a[0];                   /* assume that the first element is the minimum */
for (i=1;i<=9;i++)          /* check min with the other elements and if some element
                           is found lesser than the assumed value stored in min,
                           min is changed to the new value */
   {
    if ( min> a[i])
    min=a[i];
   }                       /* print the minimum number */

printf( "The minimum of all the elements of the array is %d",min);
}
```

ਪਰਿਣਾਮ Output:

ਮੰਨ ਲਓ ਕਿ ਇਕ ਐਰੇ (array) a ਵਿਚ ਨਿਮਨਲਿਖਿਤ ਅੰਕ ਇਨਪੁਟ (input) ਕੀਤੇ ਜਾਂਦੇ ਹਨ :

7 8 9 10 1 2 3 4 5 6

ਇਸ ਲਈ ਸਕਰੀਨ ਤੇ ਦੇਖਿਆ ਜਾਣ ਵਾਲਾ ਆਊਟਪੁਟ ਹੋਵੇਗਾ:

The minimum of all the elements of the array is 1

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਆਓ ਐਰੇ (array) ਦੀ ਇਕ ਮਿਸਾਲ ਤੇ ਵਿਚਾਰ ਕਰੀਏ ਜਿਥੇ age ਦੇ 10 ਮੁੱਲ ਯੂਜ਼ਰ ਵਲੋਂ ਇਨਪੁਟ (input) ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਉਮਰਾਂ ਦਾ ਜੋੜ ਅਤੇ ਔਸਤ ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਵੇਗਾ, ਵੱਧ ਤੋਂ ਵੱਧ ਅਤੇ ਘਟੋ ਘੱਟ ਉਮਰ ਦਾ ਪ੍ਰਦਰਸ਼ਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

```
#include<stdio.h>
int main ()
{
  int age[10];
  int i,sum=0, avg=0;
  int max=0,min=100;
  for(i=0;i<10;i++)
    {
     printf("Enter the age of student%d: " , i+1);
    scanf("%d",&age[i]);
    }
     for(i=0;i<10;i++)
       {
         sum=sum+age[i];
         if(age[i]>max)
           {
             max=age[i];
           }
          if(age[i]<min)
           {
            min=age[i];
           }
       }
      avg=sum/10;
printf("Average age of the students of the class : " ,"avg" );
printf("Maximum age of the student of the class : " ,"max");
printf("Minimum age of the student of the class : " ,"min");
return(0);
}
```

ਪ੍ਰੋਗਰਾਮ **Output** ਦਾ ਨਤੀਜਾ:-

ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਐਰੇ (array) ਦੀ ਡਿਕਲੇਅਰ ਇਸ ਸਟੇਟਮੈਂਟ (statement) ਦੁਆਰਾ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

int age[10];

age ਐਰੇ ਦਾ ਆਈਡੈਂਟੀਫਾਇਰ (identifier) ਹੈ ਜੋ int type ਹੁੰਦਾ ਹੈ ਅਤੇ ਜਿਸਦਾ ਅਕਾਰ 10 ਹੈ। ਫਿਰ ਲੂਪ (loop) ਵਿਚ ਯੂਜ਼ਰ ਨੂੰ ਵਿਦਿਆਰਥੀ ਦੀ age ਪ੍ਰਵੇਸ਼ ਕਰਨ ਲਈ ਕਿਹਾ ਜਾਂਦਾ ਹੈ, ਅਤੇ ਇਸ (statement) ਦੁਆਰਾ ਐਰੇ ਦੇ ਐਲੀਮੈਂਟ (element) ਵਿਚ age ਸਟੋਰ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

scanf("%d",&age[i]);

ਇਥੇ **i** ਐਰੇ (array) ਦੀ ਸੂਚੀ ਹੈ। ਸੂਚੀ 0 ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦੀ ਹੈ। **for loop** ਅਗੇ ਔਸਤਨ ਉਮਰ, ਨਿਊਨਤਮ age ਅਤੇ ਉੱਚਤਮ age ਜੋ ਵਿਦਿਆਰਥੀ ਦੀ ਹੁੰਦੀ ਹੈ ਦੀ ਗਿਣਤੀ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

ਹਰ ਇਕ ਐਲੀਮੈਂਟ (element) ਦੀ ਪਹੁੰਚ ਸੂਚੀ i ਦੁਆਰਾ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਐਰੇ ਦੇ ਐਲੀਮੈਂਟਸ (elements) ਦਾ ਜੋੜ ਹੇਠ ਲਿਖੀ ਸਟੇਟਮੈਂਟ (statement) ਦੁਆਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ :-

sum=sum+age[i];
age[i] is the (i+1)th element of the array.

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਉੱਚਤਮ age ਸਟੇਟਮੈਂਟ (statement) **if** ਦੁਆਰਾ ਗਿਣੀ ਜਾਂਦੀ ਹੈ।

```
if(max>age[i])
{
 max=age[i];
}
```

ਨਿਊਂਨਤਮ age ਇਕ ਹੋਰ ਸਟੇਟਮੈਂਟ (statement) **if** ਦੁਆਰਾ ਗਿਣੀ ਜਾਂਦੀ ਹੈ।

```
if(min<age[i])
{
  min=age[i];
}
```

ਔਸਤਨ ਦੀ ਗਿਣਤੀ ਸਟੇਟਮੈਂਟ (statement) ਦੁਆਰਾ ਕੀਤੀ ਜਾਂਦੀ ਹੈ

```
avg=sum/10;
```

ਇਕ ਐਰੇ (array) ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਬਾਈਟਾਂ ਦੀ ਕੁਲ ਗਿਣਤੀ

```
Total bytes=size of(base type)xsize of array.
```

o

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਐਰੇਜ਼ *(Arrays)*

ਐਰੇ ਇਕ ਡਾਟਾ ਸਟਰਕਚਰ ਹੁੰਦੀ ਹੈ ਜਿਹੜੀ ਇਕ ਦਿਤੇ ਗਏ ਨਾਮ ਤੇ ਅਜਿਹੇ ਮੁੱਲਾਂ ਦੇ ਸੰਗ੍ਰਹਿ ਦਾ ਸਟੋਰ ਕਰਦੀ ਹੈ, ਇਨ੍ਹਾਂ ਨੂੰ ਨਿਮਨਲਿਖਿਤ ਵੇਰੀਏਬਲ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਐਰੇ ਦੇ ਅੰਸ਼ ਮੈਮੋਰੀ ਸਥਿਤੀਆਂ ਵਿਚ ਸਟੋਰ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਇਨ੍ਹਾਂ ਅੰਸ਼ਾਂ ਵਿਚ ਐਰੇ ਵਿਚ ਉਨ੍ਹਾਂ ਦੇ ਨਾਵਾਂ ਨਾਲ ਪਹੁੰਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਜੋ ਕਿ ਬਰੈਕਟਾਂ ਵਿਚ ਨਿਮਨਲਿਖਤ ਅੰਕ ਦੁਆਰਾ ਅਪਣਾਏ ਜਾਂਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1: ਬਹੁਪੱਖੀ ਪਸੰਦ ਦੇ ਪ੍ਰਸ਼ਨ

1.1 ਐਰੇ **(array)** ਨਾਮ ਹੈ ?
(ਏ) ਐਰੇ (array) ਵੇਰੀਏਬਲ (variable), (ਬੀ) ਇਕ ਕੀ-ਵਰਡ,
(ਸੀ) ਸਮੂਹ ਅੰਸ਼ਾਂ ਵਲੋਂ ਵੰਡਿਆ ਜਾਣ ਵਾਲਾ ਇਕ ਸਾਂਝਾ ਨਾਮ (ਡੀ) ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਕੋਈ ਵੀ ਨਹੀਂ।

1.2 **C** ਵਿਚ ਐਰੇ **(array)** ਸੱਬਸਕਰਿਪਟ **(subscript)** ਹਮੇਸ਼ਾ ਜਿਸ ਤੋਂ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ ?
a) -1 b) 0 c) 1 d) any value

1.3 **C** ਵਿਚ, ਚਕੋਰ ਬਰੈਕਟਾਂ ਦੀ ਵਰਤੋਂ **[ ]** ਜਿਸ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
a) functions b) arrays c) statements d) all of these

1.4 ਡਿਕਲੇਅਰ (declare) **int a [4] [5]** ਵਿਚ ਅੰਸ਼ਾਂ ਦੀ ਵੱਧ ਤੋਂ ਵੱਧ ਗਿਣਤੀ ?
a) 28 b) 32 c) 20 d) 9

1.5 ਜੇ ਕਰ ਡਿਕਲੇਅਰੇਸ਼ਨ **(declaration) int a [10]** ਹੈ ਤਾਂ ਸ਼ਨਾਖਤ ਕਰੋ ਕਿਹੜੀ ਗਲਤ ਹੈ ?
a) a[-1] b) a[0] c) a[10] d) ++a

ਪ੍ਰਸ਼ਨ 2: ਦੱਸੋ ਸਹੀ ਹੈ ਜਾਂ ਗਲਤ ਹੈ

1. ਇਕ ਐਰੇ (array) ਇਕੋ ਪ੍ਰਕਾਰ ਦੇ ਮੁੱਲਾਂ ਦਾ ਸੰਗ੍ਰਹਿ ਹੁੰਦਾ ਹੈ ?
2. ਐਰੇ (array) ਦੇ ਅੰਸ਼ ਮੈਮੋਰੀ ਵਿਚ ਇਕ ਦੂਜੇ ਤੋਂ ਅੱਗੇ ਰਖੇ ਜਾਂਦੇ ਹਨ ?
3. ਐਰੇਸ ਆਪਣੇ ਆਪ ਡਿਕਲੇਅਰੇਸ਼ਨ ਸਟੇਟਮੈਂਟ (declaration statement) ਵਿਚ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ (initialize) ਕੀਤੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ ?
4. ਜੇਕਰ ਐਰੇ (array) ਦਾ ਆਕਾਰ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਦੀ ਗਿਣਤੀ ਤੋਂ ਘੱਟ ਹੁੰਦਾ ਹੈ ਤਾਂ ਇਹ ਗਲਤੀ ਹੁੰਦੀ ਹੈ ?

ਪ੍ਰਸ਼ਨ 3: ਨਿਮਨਲਿਖਤ ਲਈ ਉਚਿਤ ਸ਼ਬਦ ਜਾਂ ਪਰਿਣਾਮ ਲਿਖੋ।

1. ਜੇਕਰ ਅਸੀ **x** ਐਰੇ (array) ਦੀ ਡਿਕਲੇਅਰੇਸ਼ਨ (declaration) ਤੋਂ **x [size]** ਵਿਚ ਐਰੇ (array) ਦੇ ਅਕਾਰ ਨੂੰ ਮਿਟਾ ਦਿੰਦੇ ਹਾਂ ਤਾਂ ਸੰਗ੍ਰਹਿ ਵਲੋਂ **x** ਨੂੰ ਕਿਸ ਪ੍ਰਕਾਰ ਦਾ ਵੇਰੀਏਬਲ ਸੋਚਿਆ ਜਾਵੇਗਾ ?
2. ਇਕ ਐਰੇ (array) ਡਿਕਲੇਅਰੇਸ਼ਨ (declaration) ਵਿਚ **[ ]** ਦਾ ਅਕਾਰ ਨਿਸ਼ਚਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ?
3. ਜਦੋਂ ਐਰੇ (array) ਵਿਚ ਲੂਪ ਨੂੰ ਪਹਿਲ ਦਿਤੀ ਜਾਂਦੀ ਹੈ, ਤਾਂ ਐਰੇ (array) ਦਾ ਆਕਾਰ ਹਮੇਸ਼ਾ ਹੀ ਕੀ ਹੁੰਦਾ ਹੈ ?

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰਸ਼ਨ 4: ਸੰਖੇਪ ਉੱਤਰ ਪ੍ਰਕਾਰ ਦੇ ਪ੍ਰਸ਼ਨ

1. ਦਸੋ ਐਰੇ (array) ਦੇ ਸੈੱਲਾਂ ਵਿਚ ਕੀ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਅਸੀ ਲਿਖਦੇ ਹਾਂ ?
   int x[5] = {8 , -2,20,26, -1000);

2. 50 ਅੰਸ਼ਾਂ ਦੇ ਨਾਲ ਡਾਟਾ ਨਾਮਕ ਅਸਲ ਐਰੇ (array) ਕਿਵੇਂ ਡਿਕਲੇਅਰ ਕੀਤੀ ਜਾਵੇਗੀ? ਅੰਤਿਮ ਅੰਸ਼ ਦਾ ਸਬਸਕਰਿਪਟ ਕੀ ਹੁੰਦਾ ਹੈ ?

3. ਇਹ ਸਟੇਟਮੇਂਟ (statement) ਕਿਨ੍ਹਾਂ ਲਈ ਅਰਥ ਰਖਦੇ ਹਨ ?
   a =4; **num [a] = 7;**

4. ਕੀ ਦੋ ਪਰਿਭਾਸ਼ਾਵਾਂ ਨੂੰ ਸੈੱਲਾਂ ਦਾ ਇਕੋ ਨੰਬਰ ਦਿਤਾ ਜਾਂਦਾ ਹੈ ?
   char characters[1000] ;
   int numbers [10][100]; **justify your answer ?**

5. ਇਕ ਐਰੇ (array) ਨੂੰ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਦਾ ਕੀ ਮਨੋਰਥ ਹੁੰਦਾ ਹੈ?

ਪ੍ਰਸ਼ਨ 5: ਪ੍ਰਸ਼ਨਾਂ ਦੇ ਉੱਤਰ ਦਿਓ

1. ਇਕ ਐਰੇ (array) ਅਤੇ ਸਧਾਰਣ-ਵੇਰੀਏਬਲ (simple variable) ਵਿਚ ਫਰਕ ਦਸੋਂ ?

2. ਐਰੇਸ (arrays) ਨੂੰ ਨਾਮ ਦੇਣ ਦੇ ਬੁਨਿਆਦੀ ਨਿਯਮ ਕੀ ਹਨ ?

3. ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਦੇ ਬਾਰੇ ਨਿਮਨਲਿਖਤ ਦੇ ਉੱਤਰ ਦਿਓ ? int amount[25];
   (ਏ) ਐਰੇ (array) ਵਿਚ ਕਿੰਨੇ ਅੰਸ਼ ਹੁੰਦੇ ਹਨ ?
   (ਬੀ) ਐਰੇ (array) ਦੇ ਉਪਰਲੇ ਅਤੇ ਹੇਠਲੇ ਬੰਧਨ ਕਿਹੜੇ ਹਨ ?
   (Hint upper bound =24, lower bound = 0 why?)

4. ਜਦੋ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਦੇ ਮੁੱਲ ਦੀ ਗਿਣਤੀ ਐਰੇ (array) ਲਈ ਨਿਸ਼ਚਿਤ ਵਿਸਥਾਰ ਨਾਲੋਂ ਵੱਧ ਹੁੰਦੀ ਹੈ ਤਾਂ ਕੀ ਹੁਦਾ ਹੈ ?
   int item [25], amount [25], a;

   a) if a= 8 , what array element is referred to by the item [a-3] ?
   b) If a is 24, what element is amount [a+1] ?

   Ans i) item [5] ii) amount [25]

5. ਨਿਮਨਲਿਖਤ ਦੇ ਉੱਤਰ ਦਿਉ ਜੇ ਕਰ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਹੋਏ
   int item(25), **amount(25),a;**

Answer to objective questions

Q	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
Q1	c	B	b	c	a					
Q2	T	T	T	F						
Q3	Simple variable	Array	known							

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ **8**

ਐਰੇਸ (ਭਾਗ-2)
ARRAYS (Part – 2)

8.0.0 ਭੂਮਿਕਾ (Introduction)

ਸਾਧਾਰਣ ਐਰੇਸ *(arrays)* ਦੀ ਸੂਚੀ ਇਕਹਰੇ ਪੂਰਣ-ਅੰਕ *(integer)* ਦੁਆਰਾ ਦਿਖਾਈ ਜਾਂਦੀ ਹੈ। ਮਲਟੀ-ਡਾਇਮੈਂਨਸ਼ਨਲ *(multi-dimentional)* ਐਰੇ *(array)* ਦੇ ਸੰਕਲਪ ਵਿਚ, ਜਿਸ ਵਿਚ ਜਿਥੇ ਅਸੀ ਪੂਰਣ-ਅੰਕਾਂ *(integers)* ਦੀ ਕ੍ਰਮ ਅਨੁਸਾਰ ਸੂਚੀ ਜਿਵੇਂ a[3,1,5], ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਐਰੇ *(array)* ਦੀ ਸੂਚੀ ਬਣਾਉਂਦੇ ਹਾਂ। ਮਲਟੀ-ਡਾਇਮੈਂਨਸ਼ਨਲ *(multi-dimentional)* ਐਰੇ *(array)* ਵਿਚ ਸੂਚੀ ਬਣਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਸੂਚੀ ਵਿਚ ਪੂਰਣ-ਅੰਕਾਂ *(integers)* ਦੀ ਗਿਣਤੀ ਹਮੇਸ਼ਾ ਉਹੀ ਰਹਿੰਦੀ ਹੈ ਅਤੇ ਇਸਨੂੰ ਐਰੇ *(array)* ਦੀ ਡਾਈਮੈਂਨਸ਼ਨਲਿਟੀ *(dimensionality)* ਦਾ ਹਵਾਲਾ ਦਿਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਨ੍ਹਾਂ ਵਿਚੋਂ ਹਰੇਕ ਤੇ ਸੀਮਾਵਾਂ ਨੂੰ ਐਰੇ *(array)* ਦੀਆਂ ਡਾਈਮੈਂਨਸ਼ਨਸ *(dimensions)* ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਡਾਈਮੈਂਨਸ਼ਨਲਿਟੀ *(dimensionality)* k ਦੇ ਨਾਲ ਐਰੇ *(array)* ਨੂੰ ਅਕਸਰ k - ਡਾਇਮੈਂਨਸ਼ਨਲ *(dimensional)* ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਕ ਡਾਇਮੈਂਨਸ਼ਨਲ *(dimensional)* ਐਰੇ *(array)* ਸਾਧਾਰਣ ਐਰੇ *(array)* ਦੀ ਉੱਤਰਵਰਤੀ ਹੁੰਦੀ ਹੈ, ਦੋ-ਡਾਇਮੈਂਨਸ਼ਨਲ *(two dimensional)* ਐਰੇਸ *(arrays)* ਵਿਸ਼ੇਸ਼ ਤੌਰ ਤੇ ਮੈਟਰਿਸਿਸ *(matrices)* ਲਈ ਸਾਝੀ ਪ੍ਰਤਿਨਿਧਿਤਾ ਕਰਦੇ ਹਨ। ਵਿਵਹਾਰ ਵਿਚ, ਐਰੇ *(array)* ਦਾ ਡਾਈਮੈਂਨਸ਼ਨ *(dimension)* ਬਹੁਤ ਘੱਟ ਤਿੰਨਾਂ ਤੋ ਵੱਧ ਹੁੰਦਾ ਹੈ। ਮੈਮਰੀ ਵਿਚ ਇਕ ਡਾਇਮੈਂਨਸ਼ਨਲ *(dimensional)* ਐਰੇ *(array)* ਦਾ ਨਕਸ਼ਾ ਬਣਾਉਣਾ ਸ਼ਾਇਦ ਹੀ ਹੁੰਦਾ ਹੋਵੇਗਾ। ਕਿਉਂਜੋ ਮੈਮਰੀ ਆਪਣੇ ਆਪ ਵਿਚ ਹੀ ਲੌਜਿਕਲ *(logical)* ਇਕ ਡਾਈਮੈਂਨਸ਼ਨਲਿਟੀ *(dimensionality)* ਐਰੇ *(array)* ਹੁੰਦੀ ਹੈ।

ਜੇ ਕਰ ਸਬਸਕ੍ਰਿਪਟ (subscript) ਦੀ ਗਿਣਤੀ ਇਕ ਤੋ ਵਧੇਰੇ ਹੋਵੇ, ਤਾਂ ਇਹਨਾ ਐਰੇਜ਼ ਨੂੰ ਮਲਟੀ ਡਾਇਮੈਂਸ਼ਨਲ (multi dimensional) ਤਰਤੀਬ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਸ ਲਈ, ਅਸੀ ਕਹਿ ਸਕਦੇ ਹਾਂ 2 ਡਾਇਮੈਂਸ਼ਨਲ (dimensional), 3 ਡਾਇਮੈਂਸ਼ਨਲ (dimensional) ਐਰੇ। ਕਿਸੇ ਵੀ ਐਰੇ ਦੇ ਆਕਾਰ ਦਾ ਪਤਾ ਉਸ ਦੇ ਨਾਮ ਤੋਂ ਬਾਅਦ ਦਿੱਤੇ ਗਏ ਚਕੋਰ ਬਰੈਕਟ (square bracket) ਵਿਚ ਨੰਬਰਾਂ ਤੋਂ ਲਗਾਇਆ ਜਾ ਸਕਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਦਿਤਾ ਗਿਆ ਹੈ

ਮਿਸਾਲ :

1. array 1 [ ] → one-dimensional array
2. array 2 [ ] [ ] → two-dimensional array
3. array 3 [ ] [ ] [ ] → three-dimensional array

ਪ੍ਰੋਗਰਾਮਰ ਬਰੈਕਟਾਂ ਦੇ ਜੋੜ ਵਿਚਕਾਰ ਐਲੀਮੈਂਟਸ ਦੇ ਨੰਬਰਾਂ ਨੂੰ ਲਿਖੇਗਾ ਜੋ ਕਿ ਕਾਰਜ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਣਗੇ। ਹੇਠਾਂ ਲਿਖੇ ਹਿੱਸੇ ਵਿਚ 2 ਡਾਇਮੈਂਸ਼ਨਲ (dimensional) ਐਰੇ ਦੀ ਵਿਸਥਾਰ ਨਾਲ ਵਿਆਖਿਆ ਕੀਤੀ ਗਈ ਹੈ।

8.1.0 ਦੋ - ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ (Two dimensional array)

ਇਹ ਇਕੋ ਕਿਸਮ ਦੇ ਮੁੱਲਾਂ ਦੀ ਕਤਾਰਬੰਦੀ ਹੈ। ਇਹਨਾਂ ਨੂੰ ਆਮ ਤੌਰ ਤੇ ਮੈਟਰਿਕਸ (matrix) ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਜਿਹਨ੍ਹਾਂ ਵਿਚ ਕਾਲਮ (colum) ਅਤੇ ਰੋ (row) ਹਨ। ਇਹਨਾਂ ਨੂੰ 2 ਸਬਸਕ੍ਰਿਪਟ (subscript) ਵੇਰਿਏਬਲਸ (variables) ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

8.1.1 ਦੋ - ਡਾਇਮੈਂਸ਼ਨਲ ਐਰੇ ਦੀ ਡਿਕਲੇਰੇਸ਼ਨ (declaration of Two dimensional array)

C ਭਾਸ਼ਾ ਵਿਚ ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ (dimensional) ਐਰੇ ਨੂੰ ਡਿਕਲੇਰੇਸ਼ਨ (declaration) ਕਰਨ ਲਈ ਵਾਕ ਰਚਨਾ :-

data type array _ name [row] [columns]:

ਜਿਥੇ ਰੋ (row) --> ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਸੰਖਿਆ ਜੋ ਸਬਸਕ੍ਰਿਪਟ (subscript) 1 ਦੇ ਵਿਚ (under) ਪ੍ਰੋਸੈਸ ਹੁੰਦੇ ਹਨ।

ਕਾਲਮ (column) --> ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਸੰਖਿਆ ਜੋ ਸਬਸਕ੍ਰਿਪਟ (subscript) 2 ਦੇ ਵਿਚ (under) ਪ੍ਰੋਸੈਸ ਹੋਣੇ ਹਨ।

ਮਿਸਾਲ

1. int marks [5] [3];
2. float matrix [3] [3];
3. char page [25] [80];

ਪਹਿਲੀ ਮਿਸਾਲ **int marks [5] [3];** ਇਹ ਦਰਸਾਉਂਦੀ ਹੈ ਕਿ marks, ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ-ਐਰੇ ਹੈ, ਜਿਸਦੇ 5 ਰੋ (rows) ਅਤੇ 3 ਕਾਲਮ (column) ਹਨ।

ਦੂਸਰੀ ਮਿਸਾਲ **float matrix [3] [3];** ਇਸ ਵਿਚ ਮੈਟਰਿਕਸ (matrix) ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ-ਐਰੇ (dimensional array) ਹਨ ਜਿਸ ਵਿਚ 3 ਰੋ (rows) ਅਤੇ 3 ਕਾਲਮ (column) ਹਨ।

ਤੀਜੀ ਮਿਸਾਲ **(char page [25] [80];** ਇਸ ਵਿਚ 25 ਰੋ (rows) ਅਤੇ 80 ਕਾਲਮ (column) ਹਨ।

8.1.2 ਆਉ ਹੁਣ ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ (dimensional) ਐਰੇ ਦੀ ਇਕ ਬਣਤਰ ਦੇਖੀਏ

ਐਰੇ ਦੇ ਵਿਚ 3 ਰੋ (rows) ਅਤੇ 4 ਕਾਲਮ (column) ਹਨ ਇਸ ਲਈ ਇਸ ਨੂੰ 3X4 ਐਰੇ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।ਆਮ ਤੌਰ ਤੇ ਜੇ m ਰੋ (row) ਤੇ n ਕਾਲਮ (column) ਹੋਣ ਤੇ ਇਸ ਨੂੰ ਅਸੀਂ m x n ਐਰੇ ਕਹਿੰਦੇ ਹਾਂ।

	Column 0	Column 1	Column 2	Column 3
Row 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
Row 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
Row 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

ਇਸ ਐਰੇ ਵਿਚ A[1][2] ਦਾ ਮਤਲੱਬ ਹੈ ਕਿ [1] ਰੋ (row) ਸਬਸਕ੍ਰਿਪਟ (subscript) ਹੈ ਤੇ [2] ਕਾਲਮ (column) ਸਬਸਕ੍ਰਿਪਟ (subscript) ਹੈ ਤੇ ਇਸ ਐਰੇ ਦੀ ਹਰ ਐਲੀਮੈਂਟ (element) ਦੀ ਪਹਿਚਾਣ A[i][j] ਨਾਲ ਹੁੰਦੀ ਹੈ ਜਿਥੇ A ਐਰੇ ਦਾ ਨਾਮ ਹੈ ਅਤੇ i ਤੇ j ਸਬਸਕ੍ਰਿਪਟ (subscript) ਹਨ ਜੋ ਕਿ ਐਰੇ ਵਿਚ ਹਰ ਐਲੀਮੈਂਟ (element) ਨੂੰ ਦਰਸਾਉਂਦੀਆਂ ਹਨ।

8.1.3 ਦੋ ਡਾਇਮੇਨਸ਼ਨਲ (2-dimensional) ਐਰੇ ਦਾ ਇਨੀਸ਼ਿਯਲਾਈਜੇਸ਼ਨ (initialization)

ਇਕ ਡਾਇਮੇਨਸ਼ਨਲ (dimensional) ਐਰੇ ਦੇ ਆਰੰਭੀਕਰਣ ਦੀ ਹੀ ਤਰ੍ਹਾਂ ਦੋ ਡਾਇਮੇਨਸ਼ਨਲ (dimensional) ਐਰੇ ਦੇ ਐਲੀਮੈਂਟ (element) ਦਾ ਆਰੰਭੀਕਰਣ ਵੀ ਘੋਸ਼ਣਾ ਦੇ ਦੌਰਾਨ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ। ਦੋ ਡਾਇਮੇਨਸ਼ਨਲ (dimensional) ਐਰੇ ਦੇ ਆਰੰਭੀਕਰਣ ਦੀ ਵਾਕ ਰਚਨਾ ਹੇਠ ਅਨੁਸਾਰ ਹੈ :-

ਮਿਸਾਲ 1 ਹੇਠ ਲਿਖੀ ਘੋਸ਼ਣਾ ਨੂੰ ਦੇਖਿਏ

int matrix [3] [3] = {1, 2, 3, 4, 5, 6, 7, 8, 9}

Then, the first 9 elements of the matrix will be,

matrix [0] [0] = 1;	matrix [0] [1] = 2;	matrix [0] [2] = 3;
matrix [1] [0] = 4;	matrix [1] [1] = 5;	matrix [1] [2] = 6;
matrix [2] [0] = 7;	matrix [2] [1] = 8;	matrix [2] [2] = 9;

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਜੇ ਕਰ ਸੌਂਪੇ ਹੋਏ (assigned) ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਗਿਣਤੀ ਕੁੱਲ ਐਲੀਮੈਂਟਸ (elements) ਦੀ ਗਿਣਤੀ ਨਾਲੋਂ ਘਟ ਹੋਵੇ ਤਾਂ ਬਕਾਇਆ ਸਾਰੇ ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਜ਼ੀਰੋ ਮੁੱਲ ਪ੍ਰਦਾਨ ਕਰ ਦਿੰਦੀ ਹੈ।

ਮਿਸਾਲ 2 **Float xy[2] [2] = {1.0, 1.5, 2.0}**

ਇਥੇ xy ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ (dimensional) ਐਰੇ ਹੈ ਜਿਸ ਵਿਚ 2X2 ਐਲੀਮੈਂਟ (element) ਹਨ ਜੋ ਕਿ ਫਲੋਟ ਡਾਟਾ ਟਾਈਪ ਦੇ ਹਨ, ਲੇਕਿਨ ਪਹਿਲੇ 3 ਐਲੀਮੈਂਟਸ (elements) ਦਾ ਹੀ ਆਰੰਭੀਕਰਣ ਕਿਤਾ ਗਿਆ ਹੈ ਤੇ ਚੌਥਾ ਐਲੀਮੈਂਟ (element) ਆਪਣੇ ਆਪ ਹੀ ਜ਼ੀਰੋ (ZERO) ਮੁੱਲ ਲੈ ਲੈਂਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਥੱਲੇ ਦਰਸਾਇਆ ਗਿਆ ਹੈ

xy [0] [0] = 1.0	**xy [0] [1] = 1.5**
xy [1] [0] = 2.0	**xy [1] [1] = 0.0**

ਮਿਸਾਲ 3 **int num [2] [2] = {{1,2}, {3,4}} ;**

ਇਥੇ ਅੰਦਰੂਨੀ ਸਮੂਹ {1,2} ਐਰੇ Num ਦੀ ਪਹਿਲੀ ਰੋ (row) ਦੇ 2 ਐਲੀਮੈਂਟ (element) ਨੂੰ 1 ਤੇ 2 ਮੁੱਲ ਪ੍ਰਦਾਨ ਕਿੱਤਾ ਗਿਆ ਹੈ **num [0] [0] =1 num [0] [1] =2**

ਇਸੇ ਤਰ੍ਹਾਂ ਦੂਸਰਾ ਸਮੂਹ {3,4} ਐਰੇ Num ਦੀ ਦੂਸਰੀ ਰੋ (row) ਦੇ ਐਲੀਮੈਂਟ (element) ਨੂੰ 3 ਤੇ 4 ਮੁੱਲ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ **num [1] [0] = 3 num [1] [1] =4**

8.1.4 ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ ਐਲੀਮੈਂਟਸ ਦੇ ਇਨੀਸ਼ਿਯਲਾਈਜੇਸ਼ਨ (initialization)

ਦੌਰਾਨ ਨਿਮਲਿਖਤ ਮੁੱਖ ਪੁਆਇੰਟਸ (points) ਯਾਦ ਰੱਖੋ:

ਸ਼ੁਰੂਆਤੀ ਮੁੱਲਾਂ ਦੇ ਸੈਟਾਂ ਦੀ ਗਿਣਤੀ ਐਰੇ (array) ਵਿਚ ਰੋ (rows) ਦੀ ਗਿਣਤੀ ਦੇ ਬਰਾਬਰ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। ਇਕ ਤੇ ਇਕ mapping ਸੰਭਾਲੀ ਗਈ ਹੈ। ਜਿਵੇਂ ਸ਼ੁਰੂਆਤੀ ਮੁੱਲਾਂ ਦਾ ਪਹਿਲਾ ਸੈਟ (set) ਪਹਿਲੀ ਰੋ (row) ਦੇ ਐਲੀਮੈਂਟ (element) ਨੂੰ ਦਿੱਤਾ ਗਿਆ ਹੈ ਅਤੇ ਦੂਜੇ ਸੈਟ (set) ਦੇ ਸ਼ੁਰੂਆਤੀ ਮੁੱਲਾਂ ਨੂੰ ਦੂਜੀ ਰੋ (row) ਦੇ ਐਲੀਮੈਂਟ (element) ਨੂੰ ਦਿੱਤਾ ਗਿਆ ਹੈ ਅਤੇ ਇਹ ਪ੍ਰਕਿਰਿਆ ਇਸੇ ਤਰ੍ਹਾਂ ਅੱਗੇ ਚਲਦੀ ਹੈ।

ਜੇਕਰ ਮੁੱਲ ਐਲੀਮੈਂਟ (element) ਦੀ ਗਿਣਤੀ ਹਰ ਇਕ ਸ਼ੁਰੂਆਤੀ ਸੈਟ ਸੰਬੰਧਤ ਰੋ (row) ਐਲੀਮੈਂਟ (element) ਦੀ ਗਿਣਤੀ ਨਾਲੋਂ ਘੱਟ ਹੁੰਦੀ ਹੈ ਤਾਂ ਉਸ ਰੋ (row) ਦੇ ਸਾਰੇ ਐਲੀਮੈਂਟ (element) ਆਪਣੇ ਆਪ ਜ਼ੀਰੋ (zero) ਹੋ ਜਾਣਗੇ ।

ਜੇਕਰ ਮੁੱਲ ਐਲੀਮੈਂਟ (element) ਦੀ ਗਿਣਤੀ ਹਰ ਇਕ ਸ਼ਰੂਆਤੀ ਸੈਟ ਵਿਚ ਸੰਬੰਧਤ ਐਲੀਮੈਂਟ (element) ਦੀ ਗਿਣਤੀ ਨਾਲੋਂ ਵੱਧ ਹੁੰਦੀ ਹੈ ਤਾਂ ਕੰਪਾਈਲੇਸ਼ਨ (compilation) ਗਲਤ ਹੋਵੇਗੀ।

ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ ਐਰੇ (Two dimension array) ਜਾਂ ਮਲਟੀ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ (multi dimension array) ਦੇ ਢੰਗ ਦੀ ਚਰਚਾ ਕਰਨ ਤੋ ਪਹਿਲਾ ਜਰਾ nested ਜਾਂ ਲੂਪ (loop) ਦੀ ਵਰਤੋਂ ਦੇਖੀਏ।

ਲੂਪ (loop) ਦੀ body ਵਿਚ ਅੰਦਰ ਹੋਰ ਵੀ ਲੂਪ ਹੋ ਸਕਦੇ ਹਨ। ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਲੂਪ (loop) ਨੂੰ nested loop ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

<pre>for (i= 1;i<5;++i)   // outer loop
{
  printf("\n") ;
     for (j=i;j<=1;- - j)   // inner loop
        printf("*") ;
}</pre>
	The sample output will be:
	<pre>*
	* *
	* * *
	* * * *</pre>

Inner for loop ਹਰ ਇਕ i ਦੇ ਮੁੱਲ ਲਈ ਦਰਸਾਇਆ ਗਿਆ ਹੈ। ਜੋ ਵੇਰੀਏਬਲ (variable) 1, 2, 3 ਅਤੇ 4 ਮੁੱਲ ਲੈਂਦਾ ਹੈ। inner loop ਇਕ ਵਾਰੀ ਦਰਸਾਇਆ ਗਿਆ ਹੈ ਜਦੋ i =1 ਹੋਵੇ, ਕੰਡੀਸ਼ਨ j= i ਦੇ ਆਧਾਰ ਤੇ (i.e. 1<=1 means once) ਦੋ ਵਾਰ ਜਦੋਂ i=2 ਹੋਵੇ, ਤਿੰਨ ਵਾਰ ਜਦੋਂ i=3 ਅਤੇ ਚਾਰ ਵਾਰ ਜਦੋ i=4 ਹੋਵੇ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Nested loops ਨਾਲ ਕੰਮ ਕਰਨ ਦੁਰਾਨ, ਤੁਹਾਨੂੰ ਇਹ ਸਮਝਣ ਦੀ ਜ਼ਰੂਰਤ ਹੈ ਕਿ inner loop ਦੇ ਪੂਰੀ ਤਰ੍ਹਾਂ ਖਤਮ ਜਾਂ ਵਿਗਨ ਹੋਣ ਤੇ ਬਾਅਦ ਹੀ external loop ਵੇਰਿਏਬਲ (variable) ਦਾ ਮੁੱਲ ਬਦਲੇਗਾ। ਇਸ ਨੂੰ ਸਮਝਣ ਲਈ ਨਿਮਨਲਿਖਤ ਕੋਡ ਲਾਈਨਾਂ ਤੇ ਗੌਰ ਕਰੋ।

```
for (outer =1 ; outer <10 ; outer + =4)
  {
     for (inner =1 ; inner <=outer ; inner + =2)
       printf("%d %4d ", outer, inner);
```

ਉਪਰਲਾ ਕੋਡ ਇਸ ਤਰ੍ਹਾਂ ਆਉਟਪੁਟ ਦੀ ਉਤਪਤੀ ਕਰੇਗਾ

	1	1	← Inner loop got over after value 1's iteration
Change in outer's value	5 5 5	1 3 5	← Inner loop got over after value 5's iteration
Change in outer's value	9 9 9 9 9	1 3 5 7 9	← Inner loop got over after value 1's iteration

ਦੋ ਡਾਈਮੈਨਸ਼ਨਲ ਐਰੇ ਨੂੰ **ਮੈਟਰਿਕਸ ਐਰੇ (matrix array)** ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਹੇਠ ਲਿਖੀ ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ ਤੇ ਗੌਰ ਕਰੀਏ ਜੋ ਸਟੂਡੈਂਟਸ (students) ਦੇ ਰੋਲ ਨੰਬਰ ਅਤੇ ਲਏ ਗਏ ਅੰਕ ਸਟੋਰ ਕਰਦਾ ਹੈ।

ਸੱਜੇ ਪਾਸੇ ਵਾਲੇ ਪ੍ਰੋਗਰਾਮ ਦੇ ਦੋ ਹਿੱਸੇ ਹਨ, ਪਹਿਲਾ ਹਿੱਸਾ for loop ਜੋ ਰੋਲ ਨੰਬਰ ਅਤੇ ਅੰਕਾ ਦੇ ਮੁੱਲਾਂ ਨੂੰ ਪੜ੍ਹਦਾ ਹੈ ਅਤੇ ਦੂਜੇ ਹਿੱਸੇ ਵਿਚ for loop ਇਹਨਾਂ ਮੁੱਲਾਂ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ।

	Col no. 0	Col no. 1
Row no 1	5555	88
Row no 2	6666	76
Row no 3	7777	78
Row no 4	8888	72

The scanf( ) statement :
scanf("%d %d , &stud[i][0], &stud[i][1]) ;
having the subscript &stud[i][0] ,here the first

```
main( )
{
  int stud[4][2];
  int i, j;

 for ( i = 0 ; i <=3; i ++)        //*inputting data
    {
     printf("\n Enter roll no. and marks") ;
     scanf("%d %d , &stud[i][0], &stud[i][1]) ;
    }
     for (i = 0 ; i <=3; i ++)     //* printing data
     printf("\n %d %d , stud[i][0], stud[i][1]);
 }
```

ਵੇਰੀਏਬਲ (Variable) stud [i] ਦੇ ਥੱਲ੍ਹਾਂ ਹੇਠਾਂ ਰੋ (row) ਨੰਬਰ ਹੁੰਦਾ ਹੈ ਜੋ ਹਰ student ਲਈ ਬਦਲਦਾ ਹੈ &stud [0], zero ਕਾਲਮ (column) ਬਾਰੇ ਦੱਸਦਾ ਹੈ ਜਿਸ ਵਿਚ rollno ਹੁੰਦਾ ਹੈ। ਇਸੇ ਤਰ੍ਹਾਂ &stud [1], ਪਹਿਲੇ ਕਾਲਮ (column) ਬਾਰੇ ਦੱਸਦਾ ਹੈ ਜਿਸ ਵਿਚ ਅੰਕ ਹੁੰਦੇ ਹਨ।

stud [0] [0] = 5555 ਅਤੇ stud [0] [1] = 88.

ਅਸੀ ਇਹ ਕਹਿ ਸਕਦੇ ਹਾ ਕਿ ਐਰੇ ਐਲੀਮੈਂਟਸ (elements) ਰੋ (row) ਬੱਧੀ ਨਾਲ ਸਟੋਰ ਹੋ ਸਕਦੇ ਹਨ ਅਤੇ ਰੋ (row) ਬੱਧੀ ਨਾਲ ਪਹੁੰਚ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

8.2.0 ਮੈਮਰੀ ਵਿਚ ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ ਐਲੀਮੈਂਟਸ : ਐਰੇ ਦੇ ਐਲੀਮੈਂਟ (element) ਇਕ ਲਗਾਤਾਰ ਚੇਨ ਵਿਚ ਸਟੋਰ ਹੁੰਦੇ ਹਨ। ਇਸ ਨੂੰ ਹੇਠਾਂ ਦੇ ਡਾਇਮੈਂਸ਼ਨਲ ਐਰੇ (two dim array) ਵਿਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ।

Stud[0][0]	stud[0][1]	stud[1][0]	Stud[1][1]	stud[2][0]	stud[2][1]	stud[3][0]	stud[3][1]
5555	88	6666	33	7777	78	8888	72
1002	1004	1006	1008	1010	1012	1014	1016

ਮਿਸਾਲ printf("marks of second student = %d",stud[1] [1]) ;

output : marks of second student = 33

8.3.0 ਮਲਟੀ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇਸ - ਕੈਰ ਟਾਈਪ (Multi dimensional Arrays) Character type

ਇਕ ਤੋ ਵੱਧ ਸਬਸਕ੍ਰਿਪਟ (subscript) ਜਾਂ ਖਾਸ ਸਾਈਜ਼ ਦੇ char ਦੇ ਕਿਸੇ ਵੀ ਐਰੇ ਨੂੰ ਮਲਟੀ ਡਾਇਮੈਂਸ਼ਨਲ ਐਰੇਸ (Multi dimensional Arrays) ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਵਾਕ ਰਚਨਾ ਹੈ **storage –class character data – name [{ size} ---- {size} ---- {size}] ;** ਜਿਵੇਂ ਪਹਿਲਾਂ ਹਰ ਇਕ ਖਾਸ ਸਾਈਜ਼ ਬਰੈਕਟਸ ਦੇ ਸੈਟ ਵਿਚ ਪੇਸ਼ ਕੀਤਾ ਗਿਆ ਹੈ। Storage – class ਵਿੱਕਲਪੀ (optional) ਹੈ।

char words[30][15]; or char tens[10][15];

15 ਵਾਂ character string ਉਪਰੋਕਤ ਮਿਸਾਲਾਂ ਵਿਚ string terminator '/0' ਹੈ।

8.3.1 ਮਲਟੀ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇਸ ਦੇ ਐਲੀਮੈਂਟਸ ਤੱਕ ਪਹੁੰਚ (Accessing elements of Multidimensional Arrays)

ਮਲਟੀ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇਸ (Multidimensional Arrays) ਦੇ ਐਲੀਮੈਂਟਸ (elements) ਦੇ ਮੁੱਲ ਵੇਰੀਏਬਲ (variable) ਦੇ ਨਾਮ, ਰੋ (row) ਅਤੇ ਕਾਲਮ (colum) ਦੇ ਐਲੀਮੈਂਟਸ (elements) ਨੂੰ ਦਰਸਾ ਕੇ ਹੋ ਸਕਦੀ ਹੈ।

ਪਰ char ਦੇ array ਦੇ case ਵਿਚ ਸਟਰਿੰਗ (string) ਅਤੇ ਸਟਰਿੰਗ (string) ਦੇ ਐਲੀਮੈਂਟ (element) ਤਕ ਪਹੁੰਚ ਕਰਨ ਵਿਚ ਫਰਕ ਹੁੰਦਾ ਹੈ ਜਦ ਕਿ ਇਹ ਦੋਵੇਂ ਹੀ ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ ਐਰੇ ਦੇ ਹਿੱਸੇ ਹਨ। ਇਸ ਨੁਕਤੇ ਨੂੰ ਨਿਤਾਰਨ ਲਈ ਨਿਮਨਲਿਖਤ ਉਦਾਹਰਣ ਵੇਖੋ:

char input [2][3];

Input [0] ਰੋ (row) 0 ਵਿਚ, ਸਟਰਿੰਗ (string) ਦੇ ਸਾਰੇ characters ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।
Input [1] ਰੋ (row) 0 ਵਿਚ, ਸਟਰਿੰਗ (string) ਦੇ ਸਾਰੇ characters ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।
Input [0][0] ਪਹਿਲੇ ਸਟਰਿੰਗ (string) ਵਿਚ ਪਹਿਲੇ char ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।
Input [0][1] ਪਹਿਲੇ ਸਟਰਿੰਗ (string) ਵਿਚ ਦੂਜੇ char ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।
Input [0] [2] ਪਹਿਲੇ ਸਟਰਿੰਗ (string) ਦੇ ਸਟਰਿੰਗ (string) ਖਤਮ ਕਰਨ ਵਾਲੇ ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।
Input [1] [0] ਦੂਜੇ ਸਟਰਿੰਗ (string) ਵਿਚ ਪਹਿਲੇ char ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।
Input [1] [1] ਦੂਜੇ ਸਟਰਿੰਗ (string) ਵਿਚ ਦੂਜੇ char ਤੱਕ ਪਹੁੰਚ ਕਰੇਗਾ।

8.3.2 ਕੈਰ ਟਾਈਪ ਮਲਟੀ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ ਦਾ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization of char type multi dimensional array)

ਇਸ ਨੰਬਰ ਦੇ ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ ਦੇ ਇਨਿਸ਼ਿਯਲਾਈਜ਼ੇਸ਼ਨ (initialization) ਦੀ ਤਰ੍ਹਾਂ ਹੀ ਹੁੰਦਾ ਹੈ। ਜਿਵੇਂ **char words [5][21] = {"programming", "computer", "debug", "system", "about"};** ਪਹਿਲਾ ਸਬਸਕ੍ਰਿਪਟ (subscript) ਐਰੇ ਵਿਚ ਸਟਰਿੰਗ (string) "programming", "computer", ਆਦਿ ਦੱਸਦਾ ਹੈ ਅਤੇ ਦੂਜਾ ਸਬਸਕ੍ਰਿਪਟ (subscript) ਸਟਰਿੰਗ ਵਿਚ character ਦੱਸਦਾ ਹੈ। ਵੇਰਿਏਬਲ (variable) ਸ਼ਬਦ [2][0] ਤੀਜਾ ਸਟਰਿੰਗ (string) ਅਤੇ ਪਹਿਲਾ character ਦੱਸਦਾ ਹੈ ਤੀਜੀ ਸਟਰਿੰਗ (string) "debug"

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਹੈ ਅਤੇ ਪਹਿਲਾ character ਇਸ ਸਟਰਿੰਗ (string) ਵਿਚ "d" ਹੈ। ਐਰੇ words ਵਿਚ ਹਰ ਇਕ ਸਟਰਿੰਗ (string) ਵਿਚ 21 characters ਸਟਰਿੰਗ (string) ਖਤਮ ਕਰਨ ਵਾਲੇ ਸਾਮਲ ਹੁੰਦੇ ਹਨ। Characters ਦੀ ਗਿਣਤੀ ਐਰੇਜ਼ (arrays) ਦੀ ਮਿਥੀ ਗਿਣਤੀ ਨਾਲੋਂ ਘੱਟ ਹੋ ਸਕਦੀ ਹੈ। ਪਰ ਗਿਣਤੀ ਦੂਜੇ ਸਬਸਕ੍ਰਿਪਟ (subscrict) ਦੇ ਵੱਧ ਮੁੱਲ ਨਾਲੋਂ ਵੱਧ ਨਹੀ ਹੋ ਸਕਦੀ। ਐਰੇ (array) ਸ਼ਬਦਾਂ ਵਿਚ ਸਟਰਿੰਗ (string) ਦੀ ਲੰਬਾਈ ਸਟਰਿੰਗ (string) ਖਤਮ ਕਰਨ ਵਾਲੇ ਦੇ ਤੌਰ ਤੇ 20 chracters ਤੋਂ ਵੱਧ ਨਹੀ ਹੋ ਸਕਦੀ। ਪਹਿਲਾ ਸਬਸਕ੍ਰਿਪਟ (subscript) 0-4 ਅਤੇ ਦੂਜਾ ਸਬਸਕ੍ਰਿਪਟ (subscript) 0=19 ਵਿਚ ਬਦਲਦਾ ਹੈ।

ਜੇਕਰ ਪੜ੍ਹੇ ਗਏ ਅੱਖਰ, ਦਰਸਾਏ ਗਏ ਅੱਖਰ ਨਾਲੋ ਜੋ ਕਿ ਇਸ ਐਰੇ ਵਿਚ 20 ਤੋਂ ਘੱਟ ਹਨ। ਖਾਲੀ ਰੱਖੇ ਗਏ ਸਥਾਨ ਨੂੰ '\0' ਦੁਆਰਾ ਭਰਿਆ ਜਾਵੇਗਾ, ਜੋ ਕਿ ਸਟਰਿੰਗ (string) ਸਮਾਪਿਤ '\0' hexadecimal ਬਰਾਬਰ ਹੈ, ਜਿਵੇਂ ਕੀ ਹੇਠਾਂ ਦਰਸਾਇਆ ਗਿਆ ਹੈ।

p	r	o	g	r	a	m	m	i	n	g	\x0	\x0	\x0	\x0	\x0	\x0	\x0	\x0	\x0	

ਪ੍ਰਿੰਟ ਫੰਕਸ਼ਨ printf ("%s", wards [1]), ਪ੍ਰੋਗਰਾਮਿੰਗ ਸਟਰਿੰਗ (string) ਨੂੰ ਪ੍ਰਿੰਟ ਕਰੇਗਾ, ਜਿਸ ਵਿਚ ਸਿਰਫ 11 ਅੱਖਰ ਹਨ ਅਤੇ 12 ਵੇਂ ਅੱਖਰ ਦੇ ਸਥਾਨ ਤੇ ਸਟਰਿੰਗ terminator ਹੈ ਜੋ ਕਿ printf ਫੰਕਸ਼ਨ ਨੂੰ ਅੱਗੇ ਵਧੱਣ ਤੋਂ ਰੋਕੇਗਾ। ਪਰ ਹੇਠਾਂ ਲਿਖਿਆ ਫੰਕਸ਼ਨ (function), ਸਾਰੇ ਅੱਖਰਾਂ ਨੂੰ ਸਟਰਿੰਗ (string) ਸਮਾਪਤੀ ਅੱਖਰ ਨਾਲ ਪ੍ਰਿੰਟ ਕਰੇਗਾ।

8.3.3 ਕੈਰ ਵੱਰਡ ਪ੍ਰੋਸੈਸਿੰਗ ਦੇ ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ ਐਰੇ (Two dimensional array of char word processing)

Text ਨੂੰ ਸ਼ਬਦ ਦਰ ਸ਼ਬਦ ਵਰਤਣ ਲਈ ਇਕ ਵੱਖਰੀ ਅੱਖਰਾਂ ਨੂੰ ਇਨਪੁਟ ਕਰਨ ਦੀ ਤਕਨੀਕ ਜਰੂਰੀ ਹੈ। ਇਕ ਸ਼ਬਦ ਨੂੰ ਦਰਸਾਉਣ ਲਈ, ਲੜੀਬੱਧ ਅੱਖਰ ਜੋ ਕਿ ਬਗੈਰ ਖਾਲੀ ਸਥਾਨ ਤੋਂ ਹੋਣ ਅਤੇ ਅੰਤ ਵਿਚ space ਜਾਂ punctuation ਨਿਸ਼ਾਨ ਨਾਲ, ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਅਸੀ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਅਸਾਨ ਬਣਾਉਣ ਲਈ ਉਹਨ੍ਹਾਂ ਸ਼ਬਦਾਂ ਦੀ ਹੀ ਵਰਤੋਂ ਕਰਾਂਗੇ ਜਿੰਨ੍ਹਾਂ ਦੇ ਅੰਤ ਵਿਚ space ਹੋਵੇ।

```
// To read five words and print them.
// inputting and outputting text word by
word
        #include <stdio.h>

        void main()
        {
          char words[5][41];
          unsigned m ;
             puts("enter 5 words");
          for (m=0;m<5;++m)
             scanf("%s" ,words[m]);
          for (m=0;m<5;++m)
             printf("%s\n",words[m]);
        }
```

Here: puts function is used to put a whole sentence on the screen;

ਵਿਸਥਾਰ :

scanf ਫੰਕਸ਼ਨ, ਜਿਵੇਂ ਹੀ space ਪੜ੍ਹਦਾ ਹੈ ਆਰਗੂਮੈਂਟਸ (arguments) ਪੜਨ ਨੂੰ ਰੋਕ ਦਿੰਦਾ ਹੈ। ਭਾਵੇਂ ਕਿ for ਸਟੇਟਮੈਂਟ ਉਹੀ scanf ਫੰਕਸ਼ਨ ਨੂੰ 5 ਵਾਰ ਵਰਤਦੀ ਹੈ। 10 ਸ਼ਬਦ, ਹਰ ਇਕ ਸ਼ਬਦ ਨੂੰ ਸਥਾਨ ਨਾਲ ਅਲੱਗ ਕੀਤਾ ਗਿਆ, ਅਤੇ ਐਰੇ ਸ਼ਬਦਾਂ ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਗਿਆ ਹੈ।

Output

```
Enter 5 words
Array two dimensional of clanguage

Array
Two
Dimensional
Of
Clanguage
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ :- This program reads and prints the elements of a matrix

```
#include <stdio.h>
void main( )
{
 int n ,a ,b, row,  col;
 int mat[10][10] ;
 printf("Enter the order of the matrix\n");
 scanf("%d %d", &row , &col);
 printf("Enter the elements of the
matrix\n");
  for(a= 0; a<row; a++)
   {
        for(b =0; b<col; b++)
         {
           scanf("%d" ,&mat[a][b]) ;
         }
   }
 printf("Matrix is \n") ;
  for(a= 0 ; a<row; a++)
   {
          for(b =0 ; b<col ; b++)
           {
             printf("%3d", mat[a][b] ) ;
           }
        printf("\n") ;
   }
}
```

Output

```
Enter the order of the matrix
3   3
Enter the elements of the matrix
1
2
3
4
5
6
7
8
9

Matrix is

    1       2       3
    4       5       6
    7       8       9
```

ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ :- To calculate the average of elements of row in 2 dimensional array

```
#include<stdio.h>
int main ()
{
 int age[2][5]= { {12,13,14,15,15}, {
12,16,17,13,12}};
 int i,j;
 int sum=0,avg=0;
 for(i=0;i<2;i++)
   {
    for( j =0;j< 5;j++)
      {
       sum=sum+age[i][j];
      }
  avg=sum/5;
 printf("Average of the elements of the row ;
%d is %d \n" , i+1 ,avg );
    sum=0;
   }
return(0);
}
```

Explanation:

ਪ੍ਰੋਗਰਾਮ ਦਾ ਪਰਿਣਾਮ ਹੈ :-

```
Average of the elements of the row 1 is 13
Average of the elements of the row 2 is 14
Press any key to continue
```

8.4.0 ਇਕ-ਕੈਰ ਟਾਈਪ ਇਨਪੁੱਟ ਆਊਟਪੁਟ (Single character input output)

ਇਨਪੁਟ ਆਊਟਪੁਟ ਦਾ ਆਮ ਕੰਮ, ਅਖਰਾਂ ਨੂੰ ਸਟੈਂਡਰਡ ਇਨਪੁੱਟ ਯੰਤਰ ਜਿਵੇਂ ਕਿ ਕੀ-ਬੋਰਡ (key board) ਤੋਂ ਪੜ੍ਹਨਾ ਅਤੇ ਅੱਖਰਾਂ ਨੂੰ ਇਨਪੁੱਟ ਯੰਤਰ ਤੋ ਪੜ੍ਹਨ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। scanf ਫੰਕਸ਼ਨ ਨੂੰ ਵੀ ਇਸ ਮੰਤਵ ਲਈ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ getchar ਫੰਕਸ਼ਨ ਦੀ ਰੂਪਰੇਖਾ ਹੇਠਾਂ ਲਿਖੇ ਪ੍ਰਕਾਰ ਹੈ :-

variable name = getchar

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਵੇਰੀਏਬਲ ਨੇਮ, ਇਕ ਸਹੀ 'C' ਵੇਰੀਏਬਲ ਹੈ, ਜੋ ਕਿ ਪਹਿਲਾ ਹੀ ਘੋਸ਼ਿਤ (declared) ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਜਿਸਦੀ ਕਿਸਮ ਕੈਰ (char) ਹੈ।

ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ:

```
# include < stdio.h >                       // assigns stdio-h header file to your program
void main ( )                               // Indicates the starting point of the program.
{
char C;                                     // variable declaration
printf ("Type one character:") ;            // message to user
C = getchar () ;                            // get a character from key board and stores it in variable C.
printf(" The character you typed is = %c", C) ;   // output
}                                           // Statement which displays value of C on
                                            // Standard screen.
```

putchar ਫੰਕਸ਼ਨ, ਜੋ ਕਿ getchar ਦੀ ਤਰ੍ਹਾਂ ਹੀ ਹੈ, ਦੀ ਵਰਤੋਂ ਅੱਖਰਾਂ ਨੂੰ ਇਕ-ਇਕ ਕਰ ਕੇ ਸਕਰੀਨ ਤੇ ਦਰਸਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਦੀ ਕਿਸਮ ਇਸ ਪ੍ਰਕਾਰ ਹੈ :-

putchar (variable name);

(ਜਿਥੇ ਵੇਰੀਏਬਲ (variable) ਇਕ ਵੇਲਿਡ (valid) C ਟਾਈਪ ਹੈ ਜੋ ਕਿ ਪਹਿਲਾਂ ਹੀ ਦਰਸਾਈ ਜਾ ਚੁਕੀ ਹੈ)

putchar(); C ਵੇਰੀਏਬਲ (variable) ਵਿਚ ਸਟੋਰ ਮੁਲੋਂ ਨੂੰ ਸਕਰੀਨ ਤੇ ਦਰਸਾਉਂਦਾ ਹੈ
ਇਹ ਪ੍ਰੋਗਰਾਮ getchar() ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ।

```
#include < stdio.h >                        // Inserts stdio.h header file into the Pgm
void main ( )                               // Beginning of main function.
{
char in;                                    // character declaration of variable in.
printf (" please enter one character");     // message to user
in = getchar ( ) ;                          // assign the keyboard input value to in.
putchar (in);                               // out put 'in' value to standard screen.
}
```

ਹੇਠ ਦਿਤਾ ਪ੍ਰੋਗਰਾਮ, keyboard ਤੇ ਅੱਖਰਾਂ ਨੂੰ ਉਦੋਂ ਤੱਕ ਪੜ੍ਹਦਾ ਹੈ ਜਦੋਂ ਤੱਕ ਅੱਖਰ '.' ਪ੍ਰਾਪਤ ਨਹੀ ਹੋ ਜਾਂਦਾ।

```
#include<stdio.h>
#include<conio.h>
main()
{
char ch;
while(ch!='.')
{
ch=getchar();
putchar(ch);
}
}
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਇਸ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਹਰ ਇਕ key ਜਿਸ ਨੂੰ ਦਬਾਇਆ ਜਾਂਦਾ ਹੈ ਸਕਰੀਨ ਉਪਰ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਨਾਲ ਹੀ Buffer ਵਿਚ ਸਟੋਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਸ Buffer ਨੂੰ "ch" variable ਨੂੰ ਸਮਰਪਿਤ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਜਦੋਂ ਉਪਭੋਗਤਾ Enter key ਨੂੰ ਦਬਾਉਂਦਾ ਹੈ।
ਇਸ ਤਰ੍ਹਾਂ input ਅੱਖਰਾਂ ਨੂੰ buffer ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਦੋਂ ਤੱਕ ਅਸੀਂ Entery key ਨੂੰ ਨਹੀ ਦਬਾਉਂਦੇ ਅਤੇ ਇਸ ਤੋਂ ਬਾਅਦ ਪ੍ਰੋਗਰਾਮ getchar() ਫੰਕਸ਼ਨ ਤੋਂ ਬਾਅਦ ਆਉਣ ਵਾਲਿਆ statemtns ਨੂੰ ਚਲਾਉਂਦਾ ਹੈ। ਜੇਕਰ buffered ਅੱਖਰਾਂ ਵਿਚ ਅੱਖਰ '.' ਹੋਵੇ, loop statement ਲਗਾਤਾਰ ਅੱਖਰਾਂ ਨੂੰ ਸਕਰੀਨ ਉਪਰ putchar() ਫੰਕਸ਼ਨ ਦੀ ਮਦਦ ਨਾਲ ਭੇਜਦਾ ਹੈ, ਜਦੋਂ ਅੱਖਰ '.' ਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ loop ਦੀ execution ਨੂੰ ਖਤਮ ਕਰ ਦਿੰਦਾ ਹੈ ਅਤੇ while loop ਤੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ।

ਜੇਕਰ ਅੱਖਰ '.' ਨਾ ਆਵੇ ਤਾ ਇਹ loop ਦੀ ਸ਼ੁਰੂਆਤ ਤੇ ਆ ਜਾਂਦਾ ਹੈ, ਜਿਥੇ getchar() ਦੀ ਸ਼ੁਰੂਆਤ ਹੁੰਦੀ ਹੈ ਅਤੇ ਉਪਭੋਗਤਾ ਦੀ input ਦੀ ਉਡੀਕ ਕਰਦਾ ਹੈ।

8.5.0 # define directive:

ਸੀ 'C' ਦੀ ਪੂਰਵ ਕ੍ਰਿਆਤਮਕ ਸ਼੍ਰੇਣੀ ਨਾਲ ਸਬੰਧ ਰਖਦੀ ਹੈ। ਜਿਵੇਂ ਕਿ ਇਸ ਦੇ ਨਾਮ ਤੋਂ ਹੀ ਪਤਾ ਲਗਦਾ ਹੈ (ਪੂਰਵ ਕ੍ਰਿਆਤਮਕ) ਕਿ ਇਹ ਹਮੇਸ਼ਾਂ ਕੰਪਾਈਲੇਸ਼ਨ (Compilation) ਤੋਂ ਪਹਿਲਾ ਰੱਖੀ ਜਾਂਦੀ ਹੈ। ਇਥੇ main() ਫੰਕਸ਼ਨ ਤੋਂ ਪਹਿਲਾਂ ਰੱਖੀ ਜਾਂਦੀ ਹੈ। ਜਿਵੇਂ ਕਿ ਅਸੀ ਜਾਂਣਦੇ ਹਾਂ ਕਿ 'C' ਭਾਸ਼ਾ ਵਿਚ ਜਦੋਂ ਵੀ ਪ੍ਰੋਗਰਾਮ ਚਲਾਇਆ ਜਾਂਦਾ ਹੈ, ਇਕ ਵੇਰੀਏਬਲ (variable) ਦੀ ਕੀਮਤ ਬਦਲਦੀ ਰਹਿੰਦੀ ਹੈ। ਜਦ ਕਿ ਇਕ ਕਾਂਸਟੈਂਟ ਦੀ ਕੀਮਤ ਇਸ ਦੀ ਭਾਸ਼ਾ ਦੇ ਆਧਾਰ ਤੇ ਸਥਿਰ ਰਹਿੰਦੀ ਹੈ। ਇਕ ਕਾਂਸਟੈਂਟ ਨੂੰ ਪ੍ਰਭਾਵਿਤ ਕਰਨ ਦੀ ਆਮ ਵਿਧੀ ਹੈ

define name value

ਕੰਪਾਈਲਰ (compiler) ਆਇਡੈਂਟਿਫਾਈਰ (identifier) ਨਾਮ ਦੀ ਹਰ ਇਕ ਹੋਂਦ ਨੂੰ ਇਸ ਦੀ ਕੀਮਤ ਨਾਲ ਬਦਲ ਦਿੰਦਾ ਹੈ। ਉਦਾਹਰਣ ਲਈ, ਇਕ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਨ੍ਹਾਂ ਕਾਂਸਟੈਂਟ (constant) ਪਰਿਭਾਸ਼ਾਵਾਂ ਨੂੰ ਪ੍ਰਯੋਗ ਕਰਦੇ ਹੋਏ ਅਸੀਂ ਚਾਹੁੰਦੇ ਹਾਂ ਕਿ ਹਰ ਇਕ ਹੋਂਦ numb, ਕੀਮਤ 7 ਨਾਲ ਬਦਲ ਜਾਵੇ, ਤਦ ਅਸੀਂ ਲਿਖਾਂਗੇ **# define numb 7** ਦੂਜੇ ਸ਼ਬਦਾਂ ਵਿਚ **# define directive** (ਨਿਰਦੇਸ਼ਿਤ) ਨੂੰ macro substitution ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ । (ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਕ ਆਈਡੈਂਟੀਫਾਇਰ (identifier) ਨੂੰ ਕਾਂਸਟੈਂਟ ਜਾਂ ਤਿੰਨ ਕਾਂਸਟੈਂਟ ਨਾਲ ਬਦਲਣ ਦੀ ਵਿਧੀ) ਆਓ ਦੇਖੀਏ ਇਹ ਕਿਵੇ ਪ੍ਰਯੋਗ ਹੁੰਦੀ ਹੈ।

ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ 1:-

```
#include <stdio.h>
#define PI  3.142
Void main ( )
{
  float rad , area ;
    printf("Enter the radius \n") ;
    scanf("%f" , &rad) ;
    area = PI * rad * rad ;
     printf("area of a circle  = %f\n" , area) ;
}
```

ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ 2:-

```
#include <stdio.h>
# define CONDITION if (a>b)
# define PRINT printf("a is greater than b\n");
main ( )
{
  int a, b ;
   printf("Enter the value of a and b \n") ;
   scanf("%d %d " , &a, &b);
CONDITION
PRINT
}
```

Note: ਕੋਈ ਵੀ Macro ਵਿਕਲਪ ਸਟਰਿੰਗ (string) ਦੇ ਵਿਚ ਬਦਲਿਆ ਨਹੀ ਜਾਵੇਗਾ।

ਮਿਸਾਲ :

```
printf("PI" = %f \n", PI);
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਇਸ ਸਟੇਟਮੇਂਟ ਵਿਚ P1 ਦੀ ਹੋਂਦ 3.142 ਨਾਲ ਬਦਲੀ ਨਹੀ ਜਾ ਸਕਦੀ ਜਦ ਕਿ ਦੂਜੀ 3.142 ਨਾਲ ਬਦਲੀ ਜਾ ਸਕਦੀ ਹੈ।

ਆਉ ਹੁਣ **# define directive** (ਨਿਰਦੇਸ਼ਿਤ) ਨੂੰ ਇਕ ਗੁਣਾਂ ਦੇ ਟੇਬਲ ਪ੍ਰਿੰਟ ਕਰਨ ਲਈ ਵਰਤੀਏ।

ਮਿਸਾਲ ਪ੍ਰੋਗਰਾਮ :-

```
//* Program to display a Multiplication table
#include <stdio.h>
#define  ROWS 5
#define COLUMNS 5
void main()
{
int row, column , product [ROWS][COLUMNS] ;
 int i,j;
 printf( " MULTIPLICATION TABEL \n\n") ;
 printf("  ") ;
 for ( j = 1;j <= COLUMNS ;j++)
          printf("%4d",j);
          printf("\n") ;
          printf("_____________________\n") ;
 for(i = 0 ; i<ROWS ; i++)
        {
          row = i+1;
          printf("%2d" ,row) ;
   for(j =1 ;j<=COLUMNS ; j++)
      {
     column = j;
     product[i][j] = row * column ;
printf("%4d",product[i][j]);
        }
          printf("\n") ;
      }
}
```

ਵਿਆਖਿਆ :

1. step 1 ਅਤੇ 2 ਦੇ ਵੇਰੀਏਬਲਸ (variables) ਰੋ (row) ਅਤੇ ਕਾਲਮ (column) ਲਈ ਬਣਾਏ ਗਏ ਹਨ।
2. step 4 ਵਿਚ ਅਸੀ 2 ਮਾਪ ਵਾਲਾ ਐਰੇ ਬਣਾਉਂਦੇ ਹਾਂ ਜਿਸ ਦਾ ਆਕਾਰ 5X5 ਹੈ।
3. step 5 ਗੁਣਾ ਦੇ ਟੇਬਲ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨ ਵਾਸਤੇ ਹੈ। ਇਥੇ ਅਸੀ ਐਰੇ (array) ਨੂੰ ਅਸੈਸ (Access) ਕਰਨ ਲਈ ਦੋ for loops ਦੀ ਵਰਤੋਂ ਕਰਾਂਗੇ। ਇਹ for(i=0;i<rows;i++) loop, ਐਲੀਮੈਂਟਸ ਨੂੰ ਰੋ (row) ਵਿਚ ਪ੍ਰਿੰਟ ਕਰੇਗਾ ਅਤੇ i ਦੀ ਹਰ ਇਕ ਕੀਮਤ ਲਈ ਸਾਰੇ ਕਾਲਮ ਪ੍ਰਿੰਟ ਕੀਤੇ ਜਾਣਗੇ ।
4. ਅਗਲੇ step ਵਿਚ ਰੋ (row) ਅਤੇ ਕਾਲਮ (column) ਦੀ ਗੁਣਾਂ ਰਖੀ ਜਾਵੇਗੀ ।
5. for loop ਜਿੰਨ੍ਹੀ ਵਾਰ ਚਲੇਗਾ ਉਹੀ ਐਰੇ (array) ਬਣੇਗਾ।

Output :
Multiplication Table

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

int age[2][5]= { {12,13,14,15,15}, { 12,16,17,13,12}};

ਉਪਰ ਦਿੱਤੀ ਸਟੇਟਮੇਂਟ ਦੋ ਡਾਇਮੈਂਸ਼ਨਲ (dimensional) ਐਰੇ ਦੇ ਪ੍ਰੋਗਰਾਮ ਨੂੰ ਘੋਸ਼ਿਤ ਅਤੇ ਆਰੰਭ ਕਰਦੀ ਹੈ।

ਇਥੇ ਪਹਿਲੀ ਰੋ (row) {12,13,14,15,15} ਅਤੇ ਦੂਜੀ ਰੋ (row) { 12,16,17,13,12} ਦੇ ਐਲੀਮੈਂਟਸ (elements) ਦਾ ਆਰੰਭ ਕਿਸ ਤਰ੍ਹਾਂ ਹੁੰਦਾ ਹੈ ਦਰਸਾਇਆ ਗਿਆ ਹੈ ।

sum=sum+age[i][j];

age[i][j] ਐਰੇ ਨੂੰ ਰੋ (row) i ਅਤੇ ਕਾਲਮ j ਦੇ ਨਾਲ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਗਿਆ ਹੈ ਅਤੇ ਇਹ ਸਟੇਟਮੇਂਟ ਰੋ (row) ਦੇ ਐਲੀਮੈਂਟਸ (elements) ਦੇ ਜਦੋ ਰੋ (row) ਦੇ ਐਲੀਮੈਂਟਸ ਦਾ ਜੋੜ ਨਿਰਧਾਰਿਤ ਹੋ ਜਾਂਦਾ ਹੈ ਤਾਂ ਸਟੇਟਮੇਂਟ ਦੁਆਰਾ ਔਸਤ (average) ਨਿਰਧਾਰਿਤ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

avg=sum/5;

ਜੋੜ ਮੁੱਲ ਜ਼ੀਰੋ (0) ਤੇ ਆਰੰਭ ਹੁੰਦਾ ਹੈ ਅਤੇ ਦੁਬਾਰਾ ਉਹੀ ਪ੍ਰਕ੍ਰਿਆ ਦੂਜੀ ਰੋ (row) ਲਈ ਦੋਹਰਾਈ ਜਾਂਦੀ ਹੈ।

o

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸੰਖੇਪ ਰਚਨਾ (Summary)

ਐਰੇਸ (Arrays)

ਐਰੇ ਦੀਆਂ ਦੋ ਕਿਸਮਾਂ ਹੁੰਦੀਆਂ ਹਨ - ਇਕ ਵਿਸਥਾਰੀ ਅਤੇ ਦੂਸਰੀ ਮਲਟੀ ਵਿਸਥਾਰਮਈ।

ਇਕ ਜੋੜ ਦੀ ਚਕੋਰ ਬਰੈਕਟ ਨੂੰ ਹਰੇਕ ਵਿਸਥਾਰ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਮਿਸਾਲ ਲਈ ਇਕ ਵਿਸਥਾਰਮਈ ਐਰੇ ਦਾ ਕੇਵਲ ਇਕ ਸਬ-ਸਕਰਿਪਟ (subscript) ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਹ ਮੈਟਰਿਕਸ ਜਾਂ ਕਾਲਮ ਮੈਟਰਿਕਸ ਵਜੋਂ ਦੇਖਿਆ ਜਾਂਦਾ ਹੈ।

ਬਹੁ-ਵਿਸਥਾਰੀ ਐਰੇਸ ਦੇ ਇਕ ਤੋਂ ਵਧ ਸਬ-ਸਕਰਿਪਟ (subscript) ਹੁੰਦੇ ਹਨ। C ਵਲੋਂ ਸਟਰਿੰਗ ਡਾਟਾ ਦੀ ਸਹਾਇਤਾ ਨਹੀਂ ਕੀਤੀ ਜਾਂਦੀ। ਪਰੰਤੂ ਸਟਰਿੰਗਸ ਚਿੰਨ੍ਹਾਂ ਦੀ ਐਰੇ ਵਜੋਂ ਇਸਤੇਮਾਲ ਕੀਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ।

ਸਟਰਿੰਗਸ ਦੀ ਜਰੂਰਤਾਂ ਲਈ ਫੰਕਸ਼ਨ ਦੀ ਭਰਪੂਰ ਲਾਇਬ੍ਰੇਰੀ ਹੁੰਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਮੱਸਿਆਵਾਂ

ਪ੍ਰਸ਼ਨ 1. ਹੇਠਾਂ ਦਿਤੇ ਪ੍ਰਸ਼ਨਾਂ ਦੇ ਜਵਾਬ ਦਿਉ ?

1. ਮਲਟੀ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ (multi dimensional array) ਨੂੰ ਪ੍ਰਭਾਸ਼ਿਤ ਕਰੋ ?

2. ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ ਐਰੇ (two dimensional array) ਨੂੰ ਪ੍ਰਭਾਸ਼ਿਤ ਕਰਨ ਦੇ ਵੱਖ-ਵੱਖ ਤਰੀਕੇ ਕੀ ਹਨ ?

3. ਉਸ ਸੂਰਤ ਦੀ ਵਿਆਖਿਆ ਕਰੋ ਜਦੋਂ ਇਕ ਰੋ (row) ਦੇ ਸਾਰੇ ਐਲੀਮੈਂਟ (element) ਆਪਣੇ ਆਪ ਜ਼ੀਰੋ ਹੋ ਜਾਂਦੇ ਹਨ ।

4. # define row 10 ਦੀ ਕੀ ਮੱਹਤੱਤਾ ਹੈ?

5. ਇਕ ਐਰੇ (array) sales ਜਿਹੜਾ 3 ਡਾਇਮੈਂਨਸ਼ਨਲ (dimensional) ਹੋ ਬਣਾਉ, ਜਿਸ ਵਿਚ 5 ਦੁਕਾਨਾਂ ਦਾ 12 ਮਹਿਨੇ ਅਤੇ 3 ਸਾਲ ਦਾ ਡਾਟਾ ਰੱਖਿਆ ਜਾ ਸਕੇ, Double sales [5][12][3] ਦੀ ਵਿਆਖਿਆ ਕਰੋ ਕਿ ਇਥੇ double ਡਾਟਾ ਟਾਈਪ ਕਿਉਂ ਪ੍ਰਯੋਗ ਕੀਤਾ ਗਿਆ ਹੇ ?

ਪ੍ਰਸ਼ਨ 2 ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ (Fill in the blanks)

1. ਪ੍ਰੋਗਰਾਮ ਦੇ ਇਕ ਆਈਡੈਂਟੀਫਾਈਰ (identifier) ਨੂੰ ਇਕ ਕਾਂਸਟੈਂਟ (constant) ਨਾਲ ਬਦਲਣ ਦੀ ਪ੍ਰਕ੍ਰਿਆ ਨੂੰ _______ ਕਹਿੰਦੇ ਹਨ?

2. _______ ਫੰਕਸ਼ਨ (function) ਇਕ ਸਟੈਂਡਰਡ (standard) ਇਨਪੁਟ ਡਿਵਾਈਸ (device) ਤੋਂ ਸ਼ਬਦ (character) ਪੜ੍ਹਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ?

3. _______ ਫੰਕਸ਼ਨ (unction) ਇਕ ਸਮੇਂ ਤੇ ਇਕ ਹੀ ਸ਼ਬਦ ਆਉਟਪੁਟ-ਯੰਤਰ (output device) ਤੇ ਲਿਖਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ?

4. ਜੇ ਕਰ ਪੜ੍ਹੇ ਗਏ character, ਨਿਸ਼ਚਿਤ ਨੰਬਰਾਂ ਨਾਲੋਂ ਘੱਟ ਹਨ ਤਾਂ ਅਣਭਰੀਆਂ ਥਾਵਾਂ _______ ਨਾਲ ਭਰੀਆਂ ਜਾਣਗੀਆਂ ?

ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ (two dimensional) ਐਰੇ (array) ਨੂੰ ਆਮ ਤੋਰ ਤੇ _______ ਕਿਹਾ ਜਾਂਦਾ ਹੈ ?

ਪ੍ਰਸ਼ਨ 3 ਗਲਤ ਜਾਂ ਸਹੀ ਦੀ ਵਿਆਖਿਆ ਕਰੋ

1. ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ (two dimensional) ਐਰੇ ਦੇ ਐਲੀਮੈਂਟ (element) ਲਾਈਨ ਵਿਚ ਰਖੇ ਜਾਂਦੇ ਹਨ ?

2. ਐਰੇ ਦੇ ਐਲੀਮੈਂਟ (element) ਉਸ ਦੀ subscript ਦੇ ਅਨੁਸਾਰ Access ਕਿਤੇ ਜਾ ਸਕਦੇ ਹਨ ?

3. ਸ਼ਬਦਾਂ ਦੀ ਕਤਾਰ ਨੂੰ ਦੋ ਡਾਇਮੈਂਨਸ਼ਨਲ (two dimensional) ਐਰੇ (array) ਦੀ ਸੂਚੀ ਵਿਚ ਰਖਿਆ ਜਾਂਦਾ ਹੈ ?

4. 5 ਕਤਾਰ ਤੇ 3 ਕਾਲਮ (column) ਵਾਲੇ ਐਰੇ (array) ਵਿਚ 8 ਐਲੀਮੈਂਟ (element) ਹੋਣਗੇ ?

5. ਜੇ ਕਰ ਐਰੇ (array) ਦਾ ਸਾਇਜ਼ ਇਸ ਨੂੰ ਸ਼ੁਰੂਆਤੀ ਦਿੱਤੇ ਗਏ ਸਾਈਜ਼ ਤੋਂ ਵੱਧ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ ਆਪਣੇ ਆਪ ਨੂੰ ਵਧਾ ਲੈਂਦਾ ਹੈ ?

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰਸ਼ਨ 4: ਬਹੁ ਪਸੰਦੀ ਪ੍ਰਸ਼ਨ

1. ਐਰੇ (array) ਪਰਿਭਾਸ਼ਾ int marks [10] [5] ਕਿਸ ਐਰੇ (array) ਦੀ ਪਰਿਭਾਸ਼ਾ ਦਿੰਦੀ ਹੈ।

a) 5 column, 10 rows
b) 10 columns 5 rows
c) both a and b are correct
d) none of these

2. ਸਟੇਟਮੈਂਟ (statement) int num[2][3] = {{3, 8 ,6}} , {9, 4, 7}};
a) assigns a value 4 to num[1][2]
b) assigns a value 7 to num[1][2]
c) assigns a value 8 to num[1][2]
d) assigns a value 9 to num[1][2]

3. ਸਟੇਟਮੈਂਟ (statement) int num[2][3] = {{1, 2}, {3, 4}, {5, 6}};

a) Assigns a value 2 to num[1][2]
b) Assigns a value 4 to num[1][2]
c) Assigns a value 3 to num[1][2]
d) gives an error message.

Answer to objective questions

ਪ੍ਰਸ਼ਨ	.1	.2	.3	.4	.5
2	Marco Subsitution	getch()	putch( )	'\x0'	matix
3	T	F	F	F	F
4	b	a	d		

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਅਧਿਆਇ

9

ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ
DESKTOP PUBLISHING

ਇਸ ਪਾਠ ਦੇ ਅੰਤ ਤੱਕ ਅਸੀਂ ਹੇਠ ਲਿਖੇ ਵਿਸ਼ਿਆਂ ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਲ ਕਰਾਂਗੇ।

1. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਕੀ ਹੈ।
2. ਪ੍ਰਿੰਟਿੰਗ ਦੇ ਤਰੀਕੇ
3. ਫੌਂਟਸ (Fonts)
4. ਸਕੇਲਿੰਗ, ਟਰੈਕਿੰਗ ਅਤੇ ਲੀਡਿੰਗ (Scaling tracking and leading)
5. ਫਰੇਮਜ਼, ਪੇਜ ਲੇਆਊਟ (Frames, Page Layout)
6. WYSIWYG
7. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਦੇ ਵਰਡ ਪ੍ਰੋਸੈਸਿੰਗ ਉੱਪਰ ਲਾਭ
8. ਡਾਕੂਮੈਂਟ ਬਾਰੇ ਵਿਚਾਰ (Planning)

9.0 ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ (ਡੀ.ਟੀ.ਪੀ.)

ਡੀ.ਟੀ.ਪੀ. ਬਾਰੇ ਜਾਣਕਾਰੀ : ਡੀ.ਟੀ.ਪੀ. ਦਾ ਸਬੰਧ ਕੰਪਿਊਟਰ ਦੀ ਉਹ ਅਪਲੀਕੇਸ਼ਨ ਨਾਲ ਹੈ ਜਿਹੜਾ ਛੋਟੀਆਂ ਕੰਪਨੀਆਂ ਜਾਂ ਵਿਅਕਤੀਆਂ ਲਈ ਕੰਪਿਊਟਰ ਰਾਹੀਂ, ਰਿਪੋਰਟਾਂ (Reports), ਵਿਜ਼ਟਿੰਗ ਕਾਰਡ (Visiting Card), ਕੈਲੰਡਰ ਇਸ਼ਤਿਹਾਰ, ਮੈਗਜ਼ੀਨ ਆਦਿ ਨੂੰ ਚੰਗੀ ਕੁਆਲਟੀ ਨਾਲ ਛਾਪਦਾ ਹੈ। ਨਵੇਂ ਯੁਗ ਦੇ ਕੰਪਿਊਟਰ ਨਾਲ ਡੀ.ਟੀ.ਪੀ. ਸਾਫਟਵੇਅਰ ਦੁਆਰਾ ਬਹੁਤ ਤਰ੍ਹਾਂ ਦੇ ਕੰਮ (ਜਿਵੇਂ ਕਿ ਇਸ਼ਤਿਹਾਰ, ਕਿਤਾਬਾਂ ਤਿਆਰ ਕਰਨਾ ਆਦਿ) ਬੜੀ ਅਸਾਨੀ ਨਾਲ ਨੇਪਰੇ ਚਾੜ੍ਹੇ ਜਾ ਸਕਦੇ ਹਨ। ਅਜਿਹੇ ਡੀ.ਟੀ.ਪੀ. ਸਾਫਟਵੇਅਰ ਦੁਆਰਾ ਕਿਸੇ ਵੀ ਭਾਸ਼ਾ ਦੇ ਅੱਖਰਾਂ ਨੂੰ ਬੜੇ ਵਧੀਆ ਢੰਗ ਨਾਲ ਅਤੇ ਬਹੁਤ ਸਾਰੇ ਤਰੀਕਿਆਂ ਨਾਲ ਲਿਖਿਆ ਜਾਂ ਛਾਪਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਸਾਧਾਰਨ ਤੌਰ ਤੇ ਇਹ ਕੰਪਿਊਟਰ ਦੀ ਯੋਗਤਾ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ਕਿ ਇਸ ਦੀ ਪ੍ਰੀਟਿੰਗ ਹੋਣ ਤੋਂ ਪਹਿਲਾਂ ਡਾਕੂਮੈਂਟ ਦਾ ਪ੍ਰਵਿਊ (Preview) ਕੰਪਿਊਟਰ ਦੀ ਸਕਰੀਨ ਤੇ ਦੇਖਿਆ ਜਾਵੇ। ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਇਕ ਪੇਜ਼ ਦੇ ਲੇਆਊਟ (Layout) ਦੀ ਯੋਗਤਾ ਬਾਰੇ ਦੱਸਦੀ ਹੈ। ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ 1985 ਵਿਚ "Introduction to Mac Publisher" ਸਾਫਟਵੇਅਰ ਨਾਲ ਸ਼ੁਰੂ ਹੋਈ ਸੀ। ਇਹ WYSIWYG layout ਦਾ ਪਹਿਲਾ ਪ੍ਰੋਗਰਾਮ ਸੀ। ਇਹ ਗੱਲ ਵੀ ਦੱਸਣਯੋਗ ਹੈ ਕਿ ਡੈਸਕ ਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਵਾਲੇ ਸਾਫਟਵੇਅਰਜ਼ ਬਣਨ ਤੋਂ ਪਹਿਲਾਂ ਇਹ ਕੰਮ ਬਹੁਤ ਸਾਰੇ ਆਮ ਲੋਕਾਂ ਦੁਆਰਾ ਕੀਤੇ ਜਾਂਦੇ ਸਨ। ਗ੍ਰਾਫਿਕ, ਟੈਕਸਟ ਜਾਂ ਹੋਰ ਡਿਜ਼ਾਇਨ (Design) ਲਈ ਇਸ ਕੰਮ ਵਿੱਚ ਖ਼ਾਸ ਹੁਨਰਮੰਦ ਲੋਕ ਸ਼ਾਮਿਲ ਕੀਤੇ ਜਾਂਦੇ ਸਨ, ਜਿਸ ਕਰਕੇ ਬਹੁਤ ਸਮਾਂ ਲੱਗਦਾ ਸੀ।

ਹੁਣ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਇਸ ਤਰ੍ਹਾਂ ਦੇ ਕੰਮ ਨੂੰ ਬੜੇ ਸੁਚੱਜੇ ਢੰਗ ਨਾਲ ਕਰਦੇ ਹਨ ਅਤੇ ਇਹ ਸਾਫਟਵੇਅਰ ਟੈਕਸਟ ਜਾਂ ਗ੍ਰਾਫਿਕ ਤੇ ਪੂਰਾ ਕੰਟਰੋਲ ਕਰ ਸਕਦੇ ਹਨ। ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਆਮ ਤੌਰ ਤੇ ਤਿੰਨ ਤਰ੍ਹਾਂ ਦੀ ਹੈ :-

1. ਪੇਜ਼ ਲੇਆਊਟ (Page Layout), ਅਡੋਬ ਪੇਜਮੇਕਰ (Adobe Page Maker) ਆਦਿ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

2. ਏਡਿਟਿੰਗ (editing) ਜਿਵੇਂ ਕਿ ਅਡੋਬ ਫੋਟੋਸ਼ੋਪ (Adobe Photoshop), ਕੋਰਲ ਫੋਟ ਪੇਂਟ Corel Photo Paint ਆਦਿ।

3. ਇਲੱਸਟ੍ਰੇਸ਼ਨ (Illustration) : ਕੋਰਲ ਡਰਾਅ (Corel Draw), ਐਮ ਐੱਸ. ਪਬਲਿਸ਼ਰ (M.S. Publisher) ਆਦਿ।

ਕੋਈ ਵੀ ਡਾਕੂਮੈਂਟ ਤਿਆਰ ਕਰਨ ਲਈ ਸਭ ਤੋਂ ਪਹਿਲਾਂ ਕਦਮ ਇਹ ਹੈ ਕਿ ਇਹ ਨਿਸ਼ਚਿਤ ਕਰ ਲਿਆ ਜਾਵੇ ਕਿ ਤੁਸੀ ਸ਼ਾਦੀ ਕਾਰਡ ਤਿਆਰ ਕਰਨਾ ਹੈ ਜਾਂ ਕਿਤਾਬ ਦਾ ਸਭ ਤੋਂ ਪਹਿਲਾਂ ਕਵਰ ਜਾਂ ਕੋਈ ਬਰਾਊਚਰ (Brochure) ਜਾਂ ਕੋਈ ਬਿਜਨਸ ਕਾਰਡ। ਜੇਕਰ ਤੁਸੀਂ ਕਿਤਾਬ ਦਾ ਟਾਈਟਲ ਕਵਰ (Title Cover) ਬਣਾਉਣਾ ਚਾਹੁੰਦੇ ਹੋ ਤਾਂ ਤੁਹਾਨੂੰ ਉਸਦਾ ਸਾਈਜ ਪਤਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਜੇ ਤੁਸੀਂ ਪਿੱਛੇ ਦੱਸਿਆ ਗੱਲਾਂ ਬਾਰੇ ਜਾਣਦੇ ਹੋ ਤਾਂ ਤੁਸੀਂ ਕਿਸੇ ਵੀ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਰਾਹੀਂ ਕੁਝ ਵੀ ਬਣਾ ਸਕਦੇ ਹੋ।

9.1 ਡਾਕੂਮੈਂਟਸ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਨਾ

ਇਹ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਦਾ ਬਹੁਤ ਜਰੂਰੀ ਹਿੱਸਾ ਹੈ। ਅਸੀਂ ਬਹੁਤ ਸਾਰੇ ਤਰੀਕਿਆ ਨਾਲ ਡਾਕੂਮੈਂਟ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਦੇ ਹਾਂ। ਪ੍ਰਿੰਟਿੰਗ ਇਕ ਪ੍ਰੋਸੈਸ ਹੈ। ਜਿਸ ਵਿਚ ਟੈਕਸਟ, ਗਰਾਫਿਕਸ ਅਤੇ ਤਸਵੀਰਾਂ ਆਦਿ ਨੂੰ ਬਾਰ-ਬਾਰ ਪ੍ਰਿੰਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਪ੍ਰਿੰਟਿੰਗ ਇਕ ਵੱਡੀ ਫੈਕਟਰੀ ਦਾ ਕੰਮ ਵੀ ਹੋ ਸਕਦੀ ਹੈ ਅਤੇ ਇਕ ਆਮ ਇੱਕਲੇ ਆਦਮੀ ਲਈ ਵੀ।

ਪ੍ਰਿੰਟਿੰਗ ਦੇ ਤਰੀਕੇ : (ਲਿਥੋ ਗ੍ਰਾਫੀ)

1. **ਆਫਸੈਟ ਪ੍ਰਿੰਟਿੰਗ** : ਇਸ ਵਿਚ ਸ਼ਿਆਹੀ ਪੇਪਰ ਤੇ ਬੈਠ ਜਾਂਦੀ ਹੈ ਹੁਣ ਆਮ ਤੌਰ ਤੇ ਹਰ ਪ੍ਰਿੰਟਿੰਗ ਆਫਸੈਟ ਹੁੰਦੀ ਹੈ। ਜੇਕਰ ਬਹੁਤ ਥੋੜ੍ਹੇ ਕਾਗਜਾਂ ਤੇ ਪ੍ਰਿੰਟਿੰਗ ਕਰਨੀ ਹੋਵੇ ਤਾਂ ਉਹ ਕੰਪਿਊਟਰ ਪ੍ਰਿੰਟਰ/ਕਾਪੀਅਰ ਤੇ ਹੋਵੇਗੀ। ਇਸ ਕਰਕੇ ਜੇਕਰ ਸਾਨੂੰ ਕਿਸੇ ਡਾਕੂਮੈਂਟ ਦੀਆਂ ਬਹੁਤ ਸਾਰੀਆਂ ਕਾਪੀਆਂ (ਸੈਂਕੜੇ, ਹਜ਼ਾਰਾਂ, ਲੱਖਾਂ ਜਾਂ ਕਰੋੜਾਂ) ਕਰਨੀਆਂ ਹੋਣ ਤਾਂ ਆਫਸੈਟ ਪ੍ਰਿੰਟਿੰਗ ਹੀ ਸਸਤੀ ਅਤੇ ਬਹੁਤ ਵਧੀਆ ਹੋਵੇਗੀ।

2. **ਲੇਜਰ ਪ੍ਰਿੰਟਿੰਗ** : ਇਹ ਆਮ ਪ੍ਰਿੰਟਿੰਗ ਹੈ ਜਿਹੜੀ ਵਧੀਆ ਟੈਕਸਟ ਅਤੇ ਗ੍ਰਾਫਿਕਸ ਨੂੰ ਪ੍ਰਿੰਟ ਕਰਦੀ ਹੈ। ਲੇਜਰ ਪ੍ਰਿੰਟਰ ਇੱਕ ਨੌਨ-ਇਮਪੈਕਟ ਫੋਟੋ ਕੋਪੀਅਰ ਟੈਕਨਾਲੋਜੀ (Non-impact Photocopier Technology) ਇਸਤੇਮਾਲ ਕਰਦਾ ਹੈ। ਇਸ ਪ੍ਰਿੰਟਿੰਗ ਵਿਚ ਇਲੈਕਟ੍ਰੀਕਲ ਚਾਰਜਜ਼ (Electrical Charges) ਰਾਹੀਂ ਸਲੀਨੀਅਮ ਕੋਟਡ ਡਰਮ (Selenium coated drum) ਉੱਤੇ ਲੇਜ਼ਰ ਬੀਮ ਪਾਈ ਜਾਂਦੀ ਹੈ। ਜਦੋਂ ਡਰਮ ਚਾਰਜ ਹੁੰਦਾ ਹੈ ਇਹ ਘੁੰਮਣ ਲੱਗ ਜਾਂਦਾ ਹੈ।

ਕਲਰ ਪ੍ਰਿੰਟਿੰਗ ਰਾਹੀਂ ਕੀਤਾ ਹੋਇਆ ਕੰਮ ਬਲੈਕ ਐਂਡ ਵਾਈਟ ਲੇਜਰ ਪ੍ਰਿੰਟਰ ਨਾਲੋਂ 10 ਗੁਣਾਂ ਮਹਿੰਗਾ ਪੈਂਦਾ ਹੈ। ਇਕ ਬਲੈਕ ਐਂਡ ਵਾਈਟ ਲੇਜ਼ਰ ਪ੍ਰਿੰਟਰ ਮਾਡਲ ਇਕ ਮਿੰਟ ਵਿਚ 200 ਪੇਜ਼ ਪ੍ਰਿੰਟ ਕਰ ਸਕਦਾ ਹੈ। ਪ੍ਰਿੰਟਿੰਗ ਸਪੀਡ ਗ੍ਰਾਫਿਕਸ ਤੇ ਵੀ ਨਿਰਭਰ ਕਰਦੀ ਹੈ। ਸਭ ਤੋਂ ਰੰਗਦਾਰ ਪ੍ਰਿੰਟ ਕਰਨ ਵਾਲਾ ਇੱਕ ਮਿੰਟ ਵਿੱਚ 100 ਪੇਜ ਪ੍ਰਿੰਟ ਕਰ ਸਕਦਾ ਹੈ। ਡੀ.ਟੀ.ਪੀ. ਲਈ ਲੇਜ਼ਰ ਪ੍ਰਿੰਟਰ ਦਾ ਰੈਜੋਲੂਸ਼ਨ 600 ਡਾਟਸ ਪਰ ਇੰਚ (resolution 600 Dots per inch (dpi)] ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।

ਫੌਂਟਸ (Fonts) : ਫੌਂਟ ਟਾਈਪ ਫੇਸ (Type Face) ਤੇ ਹੋਰ ਖੂਬੀਆਂ ਦਾ ਮਿਸ਼ਰਣ ਹੁੰਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਫੌਂਟ ਦਾ ਆਕਾਰ, ਨੰਬਰ (Point) ਅਤੇ ਚੌੜਾਈ ਆਦਿ। ਉਦਾਹਰਣ ਦੇ ਤੌਰ ਤੇ ਏਰੀਅਲ (Arial) ਇੱਕ ਟਾਈਪ ਫੇਸ ਹੈ, ਜਿਹੜਾ ਕਿ ਹਰੇਕ ਅੱਖਰ ਦੇ ਆਕਾਰ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਹੋਰ ਸ਼ਬਦਾ ਵਿਚ ਫੌਂਟ ਸਾਰੇ ਅੱਖਰਾਂ ਦਾ ਮਿਸ਼ਰਣ ਹੁੰਦਾ ਹੈ। ਇਸ ਵਿੱਚ ਸਾਰੇ ਖਾਸ ਅੱਖਰ (Special Character) ਵੀ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ ਜਿਵੇਂ ਕਿ comma (,), Semicolon (;), hyphen (-) @ ਆਦਿ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਫੌਂਟ ਫੈਮਿਲੀ ਕਈ ਤਰ੍ਹਾਂ ਦੇ ਰੂਪਾਂਤਰਾਂ (Variation) ਦਾ ਸਮੂਹ ਹੁੰਦੀ ਹੈ। ਉਦਾਹਰਣ ਦੇ ਤੌਰ ਤੇ Arial, Arial Bold, Arial Narrow ਆਦਿ ਸਾਰੀਆਂ ਏਰੀਅਲ ਫੌਂਟ ਫੈਮਿਲੀ ਦਾ ਹਿੱਸਾ ਹਨ। ਫੌਂਟ ਦੀ ਉਚਾਈ ਪੁਆਇੰਟਸ ਵਿੱਚ ਨਾਪੀ ਜਾਂਦੀ ਹੈ। ਫੌਂਟ ਦੇ ਘੱਟ ਪੁਆਇੰਟ ਦਾ ਮਤਲਬ ਹੈ ਕਿ ਛੋਟੇ ਅੱਖਰ ਅਤੇ ਜ਼ਿਆਦਾ ਪੁਆਇੰਟ ਦਾ ਮਤਲਬ ਹੈ ਵੱਡੇ ਅੱਖਰ ਹਨ। ਉਦਾਹਰਣ ਦੇ ਤੌਰ ਤੇ

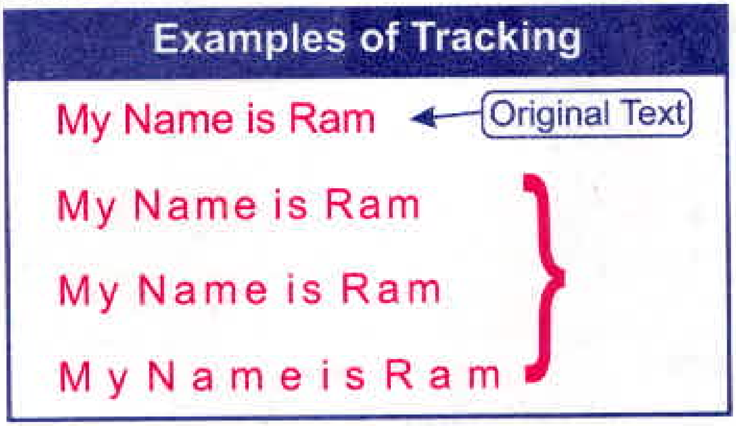
View of Content	Font Size
School	8
School	10
School	12
School	14
School	16

ਸਕੇਲਿੰਗ, ਟਰੈਕਿੰਗ ਅਤੇ ਲੀਡਿੰਗ (Scaling, Tracking and Leading) :

ਡਾਕੂਮੈਂਟ ਵਿਚ ਫੌਂਟ ਨੂੰ ਠੀਕ (Adjust) ਕਰਨਾ : ਡਾਕੂਮੈਂਟ ਵਿਚ ਬਹੁਤ ਸਾਰੇ ਤਰੀਕਿਆਂ ਦੁਆਰਾ ਫੌਂਟ ਨੂੰ ਫੈਲਾਇਆ ਅਤੇ ਸੁੰਗੋੜਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਦਾ ਮਤਲਬ ਇਹ ਹੈ ਕਿ ਫੌਂਟ ਦੇ ਪੁਆਇੰਟ ਨੂੰ ਬਿਨਾਂ ਵਧਾਇਆ ਜਾਂ ਘਟਾਇਆ, ਅਸੀਂ ਅੱਖਰਾਂ ਦੀ ਚੌੜਾਈ ਨੂੰ ਵਧਾ (increase) ਜਾਂ ਘਟਾ (decreases) ਸਕਦੇ ਹਾਂ। ਇਸ ਨੂੰ ਸਕੇਲਿੰਗ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਉਦਾਹਰਨ :

ਟਰੈਕਿੰਗ (Tracing) : ਟਰੈਕਿੰਗ ਦਾ ਮਤਲਬ ਹੈ ਕਿ ਕਿਸੇ ਵੀ ਵਰਡ ਜਾਂ ਲਾਈਨ ਦੇ ਅੱਖਰਾਂ ਵਿਚ ਦੂਰੀ (Space) ਲਿਆਉਣਾ। ਇਸਨੂੰ ਪੁਆਂਇਟਾਂ ਵਿਚ ਵੀ ਮਾਪਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਾਂ ਕਿਸੇ ਹੋਰ ਵਿਧੀ ਦੁਆਰਾ। ਉਦਾਹਰਨਾਂ

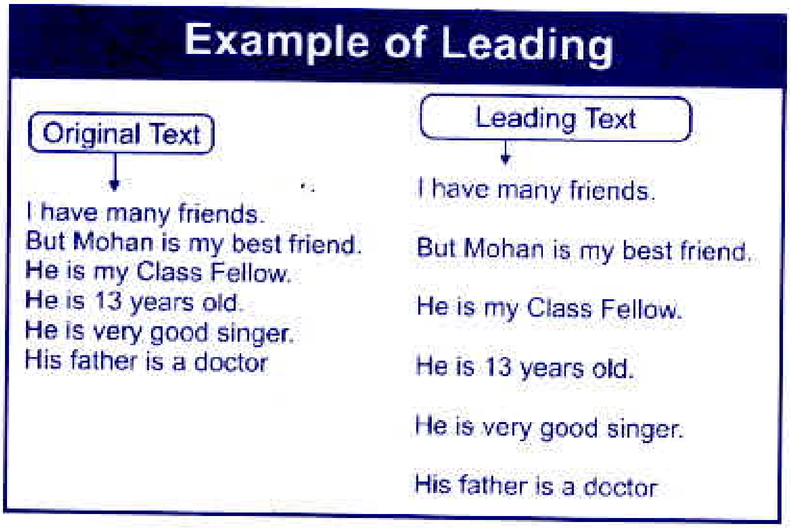


ਚਿੱਤਰ 9.1 ਟਰੈਕਿੰਗ

ਲੀਡਿੰਗ (Leading) : ਇਹ ਦੋ ਜਾਂ ਜ਼ਿਆਦਾ ਲਾਈਨਾਂ ਵਿਚਲੀ ਦੂਰੀ ਨੂੰ ਦਰਸ਼ਾਉਂਦੀ ਹੈ। ਇਸਨੂੰ ਵੀ ਪੁਆਇਂਟਾ ਵਿਚ ਮਾਪਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਾ ਇਹ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਤੇ ਨਿਰਭਰ ਕਰਦਾ ਹੈ। ਉਦਾਹਰਨ :

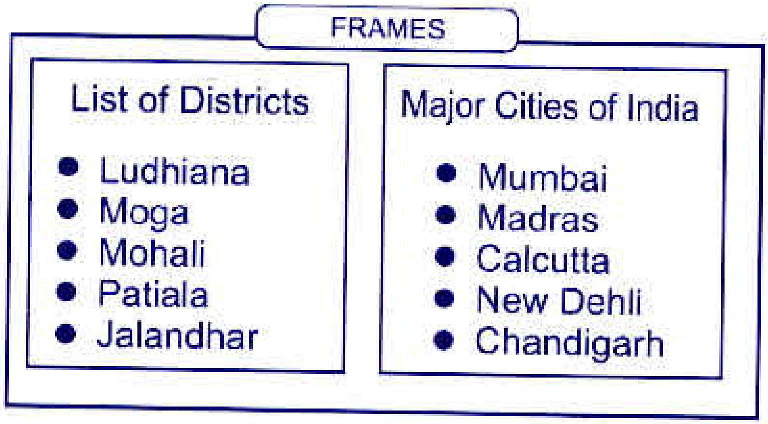
Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com



ਚਿੱਤਰ 9.2 ਲੀਡਿੰਗ

ਫਰੇਮਜ਼ (Frames) : ਫਰੇਮ ਸਬੰਧਤ ਸੂਚਨਾ ਅਤੇ ਗ੍ਰਾਫਿਕਸ ਨੂੰ ਇੱਕਠਾ ਕਰਦੇ ਹਨ। ਹਰ ਫਰੇਮ ਵਿਚ ਹਰੇਕ ਓਬਜ਼ੇਕਟ ਦੀ ਵੱਖਰੀ ਐਨਟਿਟੀ (Entity) ਹੈ। ਹਰੇਕ ਫਰੇਮ, ਬਾਕਸ, ਬਾਰਡਰ ਦਾ ਖਾਸ ਉਦੇਸ਼ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਫਰੇਮ ਅੱਖਰਾਂ ਦੀ ਸੀਮਾ ਬਣਾਉਂਦੇ ਹਨ। ਖਾਸ ਅੱਖਰਾਂ ਤੇ ਧਿਆਨ ਦਵਾਉਣ ਲਈ ਫਰੇਮ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਫਰੇਮ ਵਿਚਲੀ ਸੂਚਨਾ ਜ਼ਿਆਦਾ ਮੱਹਤਵ ਵਾਲੀ ਹੁੰਦੀ ਹੈ।



ਚਿੱਤਰ 9.3 ਫਰੇਮਜ਼

ਪੇਜ ਲੇਆਊਟ (Page Layout) : ਪੇਜ ਲੇਆਊਟ ਦਾ ਕੰਮ ਪੇਜ ਵਿਚ, ਖਾਸ ਮਕਸਦ ਲਈ ਸਾਰੇ ਆਬਜੇਕਟਾਂ (ਟੈਕਸਟ, ਗ੍ਰਾਫਿਕ ਆਦਿ) ਨੂੰ ਖਾਸ ਥਾਂ ਤੇ ਰੱਖਣਾ ਅਤੇ ਲੜੀਬੱਧ ਕਰਨਾ ਹੁੰਦਾ ਹੈ। ਪੇਜ ਲੇਆਊਟ ਦਾ ਡਿਜਾਇਨ ਕਿਸੇ ਕਾਗਜ ਤੇ ਪੇਜ ਪੈਨ ਜਾਂ ਪੈਨਸਲ ਸਕੈਚ ਨਾਲ ਬਣਾਇਆ ਜਾ ਸਕਦਾ ਹੈ ਇੱਕ ਕੰਪਿਊਟਰ ਵਿਚਲਾ ਇਲੈਕਟ੍ਰਾਨਿਕ ਪੇਜ ਜ਼ਿਆਦਾ ਗ੍ਰਾਫੀਕਲ ਯੂਜਰ ਇੰਟਰਫੇਸ (GUI) ਨਾਲ ਜਾਣਿਆ ਜਾਂਦਾ ਹੈ। ਪੇਜ ਲੇਆਊਟ ਦਾ ਸਬੰਧ ਪੇਜ ਤੇ ਉਸਦੀ ਕੰਪੋਜੀਸ਼ਨ (composition) ਨਾਲ ਹੈ। ਪੇਜ ਲੇਆਊਟ ਵਿਚ ਉਹ ਸਾਰੇ ਪਲੇਸਹੋਲਡਰ (ਗਰਿੱਡ, ਲਾਈਨ, ਬੋਕਸ) ਵੀ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ ਜਿਹੜੇ ਪ੍ਰਿੰਟ ਨਹੀਂ ਹੁੰਦੇ।

WYSIWYG (What – You – See – Is What – You – Get):- ਇਹ ਸ਼ਬਦ ਕੰਪਊਟਿੰਗ ਵਿਚ ਇਹ ਦੱਸਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਕਿ ਕੰਪਿਊਟਰ ਵਿਚਲਾ ਇਲੈਕਟ੍ਰਾਨਿਕ ਪੇਜ ਜੋ ਦਿਖਾ ਰਿਹਾ ਹੈ ਉਹ ਸਾਡੀ ਆਊਟਪੁਟ ਦੀ ਤਰ੍ਹਾਂ ਹੈ। ਇਹ ਪੇਜ ਇਕ ਪ੍ਰਿੰਟਿੰਗ ਡਾਕੂਮੈਂਟ ਹੋ ਸਕਦਾ ਹੈ, ਵੈਬ-ਪੇਜ ਜਾਂ ਸਲਾਈਡ ਵੀ ਹੋ ਸਕਦਾ ਹੈ। ਹੁਣ ਵਾਲੇ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਖਾਸ ਆਊਟਪੁਟ ਲਈ ਸਕਰੀਨ ਨੂੰ ਉਸੇ ਅਨੁਸਾਰ ਢਾਲ ਲੈਂਦੇ ਹਨ। ਉਦਾਹਰਨ ਦੇ ਤੌਰ ਤੇ ਐਮ.ਐੱਸ ਵਰਡ ਸੋਫਟਵੇਅਰ (MS-Word Software) ਆਮ ਆਦਰਸ਼ ਰੰਗਦਾਰ ਲੇਜ਼ਰ ਪ੍ਰਿੰਟਰ (Typical Color Laser Printer) ਕਾਗਜ (Hard Copy) ਤੇ ਆਊਟਪੁਟ ਲੈਣ ਯੋਗ ਹੈ। ਡੀ.ਟੀ.ਪੀ. (DTP) ਵਿੱਚ ਤਿਆਰ ਕੀਤਾ ਹੋਇਆ ਕੋਈ ਵੀ ਡਾਕੂਮੈਂਟ ਪ੍ਰਿੰਟ ਕਰਾਂਗੇ ਤਾਂ ਉਹ WYSIWYG ਦੇ ਬਿਲਕੁਲ ਨੇੜੇ ਹੋਵੇਗਾ।

ਕੁਝ ਹਾਲਤਾਂ ਵਿਚ WYSIWYG ਦਾ ਕੋਈ ਮਹੱਤਵ ਨਹੀਂ ਰਹਿ ਜਾਂਦਾ ਕਿਉਂਕਿ ਡਾਕੂਮੈਂਟ ਵਿਚ ਪੇਜ ਦੀ ਰਚਨਾ ਸਮੇਂ :-

- ਵਾਧੂ ਜਾਣਕਾਰੀ ਲਾਭਕਾਰੀ ਹੁੰਦੀ ਹੈ। ਜਿਵੇਂ ਕਿ ਕਈ ਵਾਰ ਸਾਨੂੰ ਨਾਨ-ਪ੍ਰਿੰਟਿੰਗ ਕਰੈਕਟਰ ਆਦਿ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ।
- ਪੇਜ ਲੇਆਊਟ ਮੋਡ ਵਿੱਚ ਅਸੀਂ ਅਕਸਰ ਹੀ ਗਰਿੱਡ, ਗਾਈਡਸ ਲਾਈਨਜ (Guides Lines) ਦਾ ਉਪਯੋਗ ਓਬਜੈਕਟਸ ਨੂੰ ਅਲਾਇਨ ਕਰਨ ਲਈ ਜਾਂ ਪੇਜ ਦੀਆਂ ਮਾਰਜਨ ਲਾਈਨਾਂ ਲਈ ਕਰਦੇ ਹਾਂ। ਉਦਾਹਰਨ :

ਡੈਸਕਟਾਪ ਪਬਲੀਸ਼ਿੰਗ ਅਤੇ ਵਰਡਪ੍ਰੋਸੈਸਰ ਵਿਚ ਅੰਤਰ : ਜੇ ਇਕ ਪਾਸੇ ਤੋਂ ਦੇਖਿਆ ਜਾਵੇ ਤਾਂ ਵਰਡ-ਪ੍ਰੋਸੈਸਰ (MS-Word) ਇਕ ਚੰਗੇ ਇਲੈਕਟ੍ਰਾਨਿਕ ਟਾਈਪਰਾਈਟਰ ਨਾਲੋਂ ਜ਼ਿਆਦਾ ਵਧੀਆ ਹੈ। ਇੱਕ ਸਧਾਰਨ "ਟੈਕਸਟ ਐਡੀਟਰ" ਵਿੱਚ ਤਿੰਨ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਹੁੰਦੀਆਂ ਹਨ : ਸਪੈਲਿੰਗ ਚੈਕ ਕਰਨਾ, ਹੈਡਿੰਗ, ਪੈਰਾਗ੍ਰਾਫ ਬਣਾਉਣਾ ਆਦਿ। ਵਰਡ ਪ੍ਰੋਸੈਸਰ ਦੇ ਮੁੱਖ ਕੰਮ ਹਨ : ਟੈਕਸਟ ਟਾਈਪ ਕਰਨਾ, ਸਪੈਲਿੰਗ ਚੈਕ ਕਰਨਾ, ਜੇ ਜ਼ਰੂਰੀ ਹੋਵੇ ਤਾਂ ਟੈਕਸਟ ਐਡਿਟ ਕਰਨਾ (ਬਦਲਣਾ) ਅਤੇ ਕੁਝ ਪੇਜ ਲੇਆਊਟ ਬਣਾਉਣਾ। ਜੇਕਰ ਤੁਹਾਨੂੰ ਪੇਜ ਲੇਆਊਟ ਕੰਟਰੋਲ ਜ਼ਿਆਦਾ ਲੋੜੀਂਦਾ ਹੋਵੇ (ਜਿਵੇਂ ਕਿ ਬਰਾਊਚਰ, ਅਖਬਾਰ, ਮੈਗਜੀਨ ਜਾ ਬਿਜਨਸ ਕਾਰਡ) ਤਾਂ ਤੁਹਾਨੂੰ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਦੀ ਲੋੜ ਹੋਵੇਗੀ। ਸਧਾਰਨ ਸ਼ਬਦਾਂ ਵਿਚ ਵਰਡ-ਪ੍ਰੋਸੈਸਰ ਇਕ ਟੈਕਸਟ ਬੇਸਡ ਐਡੀਟਰ (Text based editor) ਹੈ ਅਤੇ ਡੀ.ਟੀ.ਪੀ. ਆਬਜੈਕਟ ਬੇਸਿਡ (DTP Object Based) ਸਾਫਟਵੇਅਰ ਹੈ। ਡੀ.ਟੀ.ਪੀ. ਵਿਚ ਪੇਜ ਲੇਆਊਟ ਕਰਨਾ ਆਸਾਨ ਹੁੰਦਾ ਹੈ। ਇਸ ਵਿਚ ਟੈਕਸਟ ਇੱਕ ਤੋਂ ਜ਼ਿਆਦਾ ਸਥਾਨ ਤੇ ਹੁੰਦਾ ਹੈ ਅਤੇ ਹਰ ਓਬਜੈਕਟ ਲਚਕੀਲਾ (Flexible) ਹੁੰਦਾ ਹੈ।

ਡਾਕੂਮੈਂਟ ਪਲੈਨਿੰਗ

ਹੁਣ ਅਸੀਂ ਕੁਝ ਖਾਸ ਗੱਲਾਂ ਬਾਰੇ ਵਿਚਾਰ ਕਰਾਂਗੇ, ਜਿਥੇ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ਕੁਝ ਸਾਧਾਰਨ ਡਾਕੂਮੈਂਟ ਨੂੰ ਡਿਜ਼ਾਇਨ ਕਰਨ ਬਾਰੇ ਸਲਾਹ ਕਰਾਂਗੇ।

ਪੇਜ਼ ਲੇ-ਆਊਟ : ਪੇਜ਼ ਲੇ-ਆਊਟ ਤਿਆਰ ਕਰਨ ਦਾ ਮਤਲਬ ਕਿਸੇ ਡਾਕੂਮੈਟ ਵਿੱਚ ਖਾਕਾ ਤਿਆਰ ਕਰਨਾ ਹੈ।

ਜਦੋਂ ਤੁਸੀਂ ਟੈਕਸਟ, ਗ੍ਰਾਫ਼ਿਕਸ ਜਾਂ ਖਾਸ ਤਰ੍ਹਾਂ ਦੇ ਨਿਸ਼ਾਨ (Symbols), ਜਿਵੇਂ ਕਿ - ਬੌਕਸ (ਡੱਬਾ) ਫਰੇਮ, ਬਾਰਡਰ ਆਦਿ ਇਕੱਠਾ ਕਰ ਲਿਆ ਤਾਂ ਤੁਸੀਂ ਪੇਜ਼ ਲੇ-ਆਊਟ ਤਿਆਰ ਕਰ ਸਕਦੇ ਹੋ। ਇਸ ਵਿੱਚ ਇੱਕ ਤੋਂ ਵੱਧ ਪੇਜਾਂ ਦੀ ਲੋੜ ਪੈ ਸਕਦੀ ਹੈ।

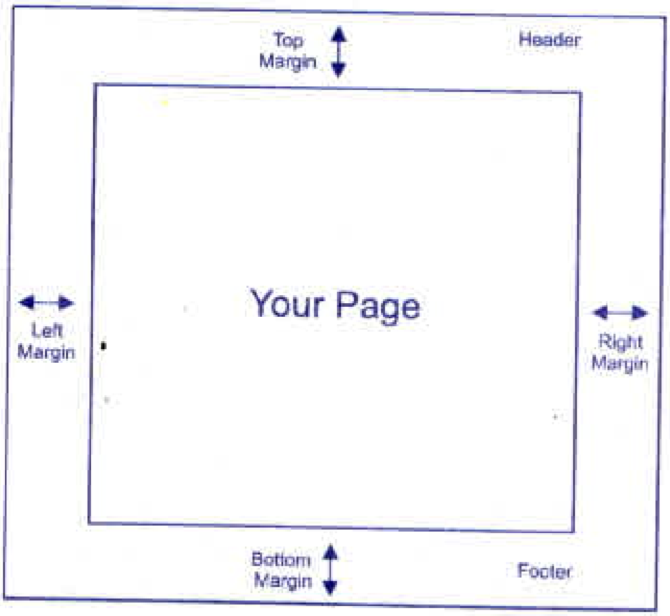
ਸਟਾਇਲ : ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਵਿੱਚ ਤੁਸੀਂ ਕਈ ਸਾਰੇ ਫੌਂਟਸ ਚੁਣ ਸਕਦੇ ਹੋ। ਲੇਆਊਟ ਐਲੀਮੈਂਟ ਗ੍ਰਾਫਿਕਸ ਵਿਚ ਉਹਨਾਂ ਵਿਚ, ਰੰਗ ਤੇ ਉਨ੍ਹਾਂ ਵਿੱਚ ਫਿਲਟਰਸ ਅਤੇ ਇਫੈਕਟ ਵੀ ਲਗਾਏ ਜਾਂ ਭਰੇ ਜਾ ਸਕਦੇ ਹਨ। ਜਿਹੜਾ ਫੌਂਟ ਤੁਸੀਂ ਚੁਣੋ, ਉਸ ਨੂੰ ਸਾਰੇ ਡਾਕੂਮੈਂਟ ਵਿੱਚ ਵਰਤੋਂ। ਡਾਕੂਮੈਂਟ ਵਿਚਲੀ ਜਾਣਕਾਰੀ ਦਾ ਸਟਾਇਲ ਵਧੀਆ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ, ਜਿਹੜਾ ਕਿ ਚੰਗੀ ਤਰ੍ਹਾਂ ਸਮਝ ਆਵੇ ਤੇ ਉਸ ਨੂੰ ਯਾਦ ਰਖਿਆ ਜਾਵੇ। ਸਟਾਈਲ ਦਾ ਮਤਲਬ ਆਪਣੇ ਡਾਕੂਮੈਂਟ ਵਿਚ ਆਪਣੇ ਆਪ ਫੋਰਮੈਟਿੰਗ (formatting) ਕਰਨ ਨਾਲ ਹੈ। ਸਟਾਈਲ ਤੋਂ ਬਿਨਾਂ ਤੁਹਾਨੂੰ ਹਰੇਕ ਪੈਰਾਗ੍ਰਾਫ, ਲਾਈਨ ਨੂੰ ਵੱਖਰੇ ਤੌਰ ਤੇ ਸੈਟਿੰਗ ਕਰਨੀ ਪਵੇਗੀ ਇਹ ਕੰਮ ਬਹੁਤ ਔਖਾ ਅਤੇ ਬੋਰਿੰਗ ਹੋ ਜਾਂਦਾ ਹੈ। ਇੱਕ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸਟਾਇਲ ਵਿੱਚ ਹੈਡਿੰਗ, ਹੈਡਰ ਅਤੇ ਫੁਟਰ, ਬੁਲਿਟ (Heading, Header & footer, Bullets) ਆਦਿ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ।

ਮਾਰਜਨ (Margin): ਮਾਰਜਨ ਟੈਕਸਟ ਦੀ ਖੱਬੇ ਅਤੇ ਸੱਜੇ, ਉਪਰ ਅਤੇ ਥੱਲੇ ਤੋਂ ਟੈਕਸਟ ਦੀ ਦੂਰੀ ਨੂੰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।



ਚਿੱਤਰ 9.4 ਮਾਰਜਨ

ਮਾਰਜਨ ਆਮ ਤੌਰ ਤੇ (From Left, Right, Top, Bottom) ਇੱਕ ਇਂਚ ਦੇ ਹੁੰਦੇ ਹਨ। ਪਰ ਇਸ ਨੂੰ ਆਪਣੀ ਇੱਛਾ ਅਨੁਸਾਰ ਵਧਾਇਆ ਜਾ ਘਟਾਇਆ ਜਾ ਸਕਦਾ ਹੈ।

ਹੈਡਰ (Header) : ਹੈਡਰ ਉਹ ਟੈਕਸਟ ਹੈ ਜਿਹੜਾ ਕਿ ਪੇਜ ਦੇ ਉਪਰਲੇ ਹਿੱਸੇ ਤੇ ਪ੍ਰਿੰਟ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਪੇਜ ਨੰਬਰ, ਪਾਠ ਨੰਬਰ ਜਾਂ ਕੋਈ ਖਾਸ ਸੂਚਨਾ ਹੋ ਸਕਦੀ ਹੈ।

ਫੁਟਰ (Footer) : ਫੁਟਰ ਉਹ ਟੈਕਸਟ ਹੈ, ਜਿਹੜਾ ਪੇਜ ਦੇ ਹੇਠਲੇ ਹਿੱਸੇ ਤੇ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਵਿੱਚ ਵੀ ਪੇਜ ਨੰਬਰ, ਪਾਠ ਦਾ ਨਾਮ ਅਤੇ ਹੋਰ ਜਾਣਕਾਰੀ ਹੋ ਸਕਦੀ ਹੈ। ਤਕਰੀਬਨ ਹਰ ਡੀ.ਟੀ.ਪੀ. ਸਾਫਟਵੇਅਰ ਆਪਣੇ ਆਪ ਹਰ ਪੇਜ ਨੰਬਰ ਅਤੇ ਹੋਰ ਸੂਚਨਾ ਲਿਖ ਲੈਂਦਾ ਹੈ।

ਫੋਂਟ : ਇਹ ਡਾਕੂਮੈਂਟ ਡਿਜ਼ਾਈਨ ਦਾ ਬਹੁਤ ਜਰੂਰੀ ਹਿੱਸਾ ਹੈ। ਆਮ ਤੌਰ ਤੇ ਪਾਠਕ ਏਰੀਅਲ (Arial) ਟਾਇਮਜ਼ ਨਿਊ ਰੋਮਨ (Times New Roman) ਸਟਾਈਲ ਵਾਲਾ ਫੋਂਟ ਪਸੰਦ ਕਰਦੇ ਹਨ ਕਿਉਂਕਿ ਇਸ ਨਾਲ ਅੱਖਰਾ ਨੂੰ ਪੜ੍ਹਨਾ ਅਸਾਨ ਹੋ ਜਾਂਦਾ ਹੈ। ਤੁਸੀਂ ਕੁਝ ਫੌਂਟਸ ਨੂੰ Bold, Italic, Under Line ਆਦਿ ਕਰ ਸਕਦੇ ਹੋ। ਏਰੀਅਲ, ਟਾਇਮਜ ਨਿਊ ਰੋਮਨ ਫੋਂਟ ਕਿਤਾਬਾਂ, ਮੈਗਜ਼ੀਨ ਅਤੇ ਅਖ਼ਬਾਰਾਂ ਵਿੱਚ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

ਇਹ ਡਾਕੂਮੈਂਟ ਪਲੈਨਿੰਗ ਬਾਰੇ ਕੁਝ ਜ਼ਰੂਰੀ ਜਾਣਕਾਰੀ ਹੈ ਪਰ ਹੋਰ ਗੱਲਾਂ ਵੀ ਵਿਚਾਰਣ ਵਾਲੀਆਂ ਹਨ ਜਿਵੇਂ ਕਿ ਪੇਜ ਦਾ ਆਕਾਰ, ਪੇਜਾਂ ਦੀ ਸੰਖਿਆ, ਪੇਜ਼ ਵਿਚਲੇ ਕਾਲਮਾਂ ਦੀ ਗਿਣਤੀ, ਟਾਈਟਲ, ਸਬਟਾਇਟਲ, ਕੈਪਸ਼ਨ (Caption) ਆਦਿ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਯਾਦ ਰੱਖਣ ਯੋਗ ਗੱਲਾਂ

1. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਰਿਪੋਰਟਾਂ, ਵਿਜਿਟਿੰਗ ਕਾਰਡ, ਕੈਲੰਡਰ, ਇਸ਼ਤਿਹਾਰ, ਮੈਗਜ਼ੀਨ ਆਦਿ ਬਣਾਉਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।
2. ਕੋਈ ਵੀ ਡਾਕੂਮੈਂਟ ਤਿਆਰ ਕਰਨ ਤੋਂ ਪਹਿਲਾਂ ਸਾਡਾ ਪਹਿਲਾ ਕੰਮ ਇਹ ਹੈ ਕਿ ਇਹ ਪਤਾ ਕਰ ਲਿਆ ਜਾਵੇ ਕਿ ਕਿਸ ਤਰ੍ਹਾਂ ਦਾ ਡਾਕੂਮੈਂਟ ਤਿਆਰ ਹੋਣ ਜਾ ਰਿਹਾ ਹੈ।
3. ਪ੍ਰਿੰਟਿੰਗ ਦਾ ਕੰਮ ਟੈਕਸਟ, ਤਸਵੀਰ ਜਾਂ ਗ੍ਰਾਫਿਕ ਨੂੰ ਬਾਰ-ਬਾਰ ਛਾਪਣਾ ਹੈ।
4. ਲੇਜ਼ਰ ਪ੍ਰਿੰਟਿੰਗ ਅਤੇ ਆਫਸੈਟ ਪ੍ਰਿੰਟਿੰਗ ਡਾਕੂਮੈਂਟ ਪ੍ਰਿੰਟ ਕਰਨ ਦੇ ਆਮ ਤਰੀਕੇ ਹਨ।
5. ਫੌਂਟ ਟਾਇਪਫੇਸ ਅਤੇ ਹੋਰ ਵਿਸ਼ੇਸ਼ਤਾਵਾਂ ਦਾ ਇੱਕਠਾ ਹੈ ਜਿਵੇਂ ਕਿ ਇਸਦਾ ਅਕਾਰ, ਪੁਆਇੰਟ (Points) ਅਤੇ ਚੌੜਾਈ (Width) ਆਦਿ।
6. ਲੀਡਿੰਗ (Leading) ਦਾ ਮਤਲਬ ਹੈ ਦੋ ਜਾਂ ਜ਼ਿਆਦਾ ਲਾਈਨਾਂ ਵਿਚਕਾਰ ਦੂਰੀ।
7. ਪੇਜ ਲੇਆਊਟ (Page Layout) ਦਾ ਕੰਮ ਹੈ, ਓਬਜੈਕਟਸ ਨੂੰ ਲੜੀਬੱਧ ਤਰੀਕੇ ਨਾਲ ਰੱਖਣਾ।
8. WYSIWYG ਦਾ ਮਤਲਬ ਹੈ What You see Is What You Get.
9. ਜੇ ਤੁਹਾਨੂੰ ਡਾਕੂਮੈਂਟ ਵਿਚ ਜ਼ਿਆਦਾ ਲੇਆਊਟ ਕੰਟਰੋਲ ਦੀ ਲੋੜ ਹੈ ਤਾਂ ਤੁਹਾਨੂੰ ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਦੀ ਲੋੜ ਪਵੇਗੀ। ਉਦਾਹਰਨ ਦੇ ਤੌਰ ਤੇ ਜੇ ਤੁਸੀ ਬਰਾਊਚਰ, ਅਖਬਾਰ, ਮੈਗਜ਼ੀਨ ਜਾਂ ਕੋਈ ਬਿਜਨਸ ਕਾਰਡ ਆਦਿ ਤਿਆਰ ਕਰਨੇ ਹੋਣ।
10. ਮਾਰਜਨ ਟੈਕਸਟ ਦੀ ਸੱਜੇ ਅਤੇ ਖੱਬੇ, ਉੱਪਰ ਤੇ ਹੇਠਾਂ ਤੋਂ ਦੂਰੀ ਨੂੰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

ਅਭਿਆਸ

ਪ੍ਰ. 1 ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ :

1. ------------ਦਾ ਮਤਲਬ ਹੈ ਦੋ ਜਾਂ ਜ਼ਿਆਦਾ ਲਾਈਨਾਂ ਵਿਚਲੀ ਦੂਰੀ।
2. ਫਰੇਮ ਸਬੰਧਤ ਸੂਚਨ ਅਤੇ ਗ੍ਰਾਫਿਕਸ ਨੂੰ -------------- ਕਰਦੇ ਹਨ।
3. WYSIWYG ਦਾ ਮਤਲਬ ਹੈ-----------------।
4. ਅਸੀਂ ਕਈ ਤਰੀਕਿਆਂ ਦੁਆਰਾ ਡਾਕੂਮੈਂਟ ਨੂੰ ---------------- ਕਰ ਸਕਦੇ ਹਾਂ।
5. ਸਭ ਤੋਂ ਤੇਜ ਰੰਗਦਾਰ ਲੇਜਰ ਪ੍ਰਿੰਟਰ-------------------ਪੇਜ ਪ੍ਰਿੰਟ ਕਰ ਸਕਦਾ ਹੈ।

ਪ੍ਰ. 2 ਸਹੀ ਅਤੇ ਗਲਤ :

1. ਹੈਡਰ ਪੇਜ ਪੇਜ ਦੇ ਥੱਲੇ ਲਿਖੇ ਜਾਂਦੇ ਹਨ।
2. ਐਮ.ਐੱਸ ਵਰਡ, ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਸਾਫਟਵੇਅਰ ਹੈ।
3. ਪੇਜ ਤਿਆਰ ਕਰਨ ਲਗੇ ਸਾਨੂੰ ਪਲੈਨਿੰਗ ਦੀ ਲੋੜ ਨਹੀਂ ਹੁੰਦੀ।
4. ਸਟਾਈਲ ਵਿਚ ਬੁਲੈਟਸ ਆਦਿ ਸ਼ਾਮਿਲ ਹੁੰਦੇ ਹਨ।
5. ਸਕੇਲਿੰਗ ਦੋ ਜਾਂ ਜ਼ਿਆਦਾ ਲਾਈਨਾਂ ਵਿਚਲੀ ਦੂਰੀ ਨੂੰ ਕਿਹਾ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਪ੍ਰ. 3 ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ

1. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਤੋਂ ਕੀ ਭਾਵ ਹੈ?
2. ਪ੍ਰਿੰਟਿੰਗ ਦੇ ਕਿਹੜੇ - ਕਿਹੜੇ ਤਰੀਕੇ ਹਨ?
3. ਸਕੇਲਿੰਗ, ਟਰੈਕਿੰਗ ਅਤੇ ਲੀਡਿੰਗ ਬਾਰੇ ਜਾਣਕਾਰੀ ਦਿਉ?
4. WYSIWYG ਤੋਂ ਕੀ ਭਾਵ ਹੈ?
5. ਮਾਰਜਨ ਕੀ ਹੁੰਦੇ ਹਨ?

ਪ੍ਰ. 4 ਵੱਡੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ :

1. ਪੇਜ ਲੇਆਊਟ ਤੋਂ ਕੀ ਭਾਵ ਹੈ?
2. ਫੋਂਟਸ ਕੀ ਹੁੰਦੇ ਹਨ?
3. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਅਤੇ ਵਰਡ-ਪ੍ਰੋਸੈਸਰ ਵਿਚ ਅੰਤਰ ਲਿਖੋ?
4. ਡਾਕੂਮੈਂਟ ਪਲੈਨਿੰਗ ਤੋਂ ਕੀ ਭਾਵ ਹੈ?
5. ਡੈਸਕਟਾਪ ਪਬਲਿਸ਼ਿੰਗ ਕੀ ਹੈ। ਇਸ ਦੀ ਕਿਉਂ ਲੋੜ ਹੁੰਦੀ ਹੈ। ਕਿਹੜੇ-ਕਿਹੜੇ ਸਾਫਟਵੇਅਰ ਦੀ ਵਰਤੋਂ ਡੈਸਕਟਾਪ ਪਬਲੀਸ਼ਿੰਗ (DTP) ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

******

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਆਮ ਪ੍ਰੋਗਰਾਮਿੰਗ ਗਲਤੀਆਂ (Common Programmings Errors)

ਆਓ ਕੁਝ ਉਨ੍ਹਾਂ ਆਮ ਗਲਤੀਆਂ ਤੇ ਵਿਚਾਰ ਕਰੀਏ ਜੋ ਕਿ ਇਕ ਪ੍ਰੋਗਰਾਮਿੰਗ/ਬਿਗਨਰ ਪ੍ਰੋਗਰਾਮਿੰਗ ਕਰਦੇ ਸਮੇਂ ਕਰ ਸਕਦਾ ਹੈ।

ਕੇਸ 1: ਗੈਰ-ਸਮਾਪਤ ਕੀਤੀ ਗਈ ਟਿੱਪਣੀ (Non terminated comments)

ਦਿਤੀਆਂ ਗਈਆਂ ਟਿਪਣੀਆਂ ਦੇ ਨਾਲ ਨਿਮਨਲਿਖਤ ਸਟੇਟਮੈਂਟਸ ਤੇ ਵਿਚਾਰ ਕਰੋ

```
a = b;      /* this is a bug
c = d;      /* c = d will never happen */
```

ਜਿਵੇਂ ਕਿ ਅੰਤਿਮ ਟਿੱਪਣੀ, /* ਪਹਿਲੇ ਸਟੇਟਮੈਂਟ ਵਿਚੋਂ ਗਾਇਬ ਹੈ, ਸਮੂਹ ਸਟੇਟਮੈਂਟ ਜਿਹੜੇ ਪੈਰਵੀ ਕਰਦੇ ਹਨ, ਜਦ ਤਕ ਕਿ ਅੰਤਿਮ ਟਿਪਣੀ */ ਕੰਪਾਈਲਰ ਵਲੋਂ ਅਖੋ-ਪਰੋਖੇ (ignored) ਨਹੀ ਕੀਤੀ ਜਾਂਦੀ ਅਰਥਾਤ ਕੰਪਾਈਲਰ ਅੰਤਿਮ ਟਿੱਪਣੀ (/*) ਦੀ ਖੋਜ ਕਰੇਗਾ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਹੋਰ ਹੇਠਾਂ ਜਾਵੇਗਾ ਅਤੇ ਸਮੂਹ ਲਾਈਨਾਂ ਨੂੰ ਟਿਪੱਣੀ ਦੀ ਤਰ੍ਹਾਂ ਮੰਨੇਗਾ ਪਰੰਤੂ ਜੇਕਰ ਇਹ ਨਹੀ ਮਿਲਦੇ ਤਾਂ ਅਸੀਂ ਗਲਤ ਸੰਦੇਸ਼ ਪ੍ਰਾਪਤ ਕਰਾਂਗੇ।

ਕੇਸ 2: (ੲ) ਅਰਧ ਵਿਰਾਮ ਚਿੰਨ੍ਹ ਦੀ ਕੁਵਰਤੋਂ (Misuse of semi colon)

ਨਿਮਨਲਿਖਤ ਸਟੇਟਮੈਂਟ ਤੇ ਵਿਚਾਰ ਕਰੋ ਜਿਸ ਵਿਚ ਅਸੀ 1 ਤੋਂ 5 ਤਕ ਦੇ ਪੂਰਣ ਮੁੱਲ ਦੇ ਜੋੜ ਕਰਨ ਦੀ ਕੋਸ਼ਿਸ਼ ਕੀਤੀ ਹੈ, ਜੇਕਰ ਅਸੀ ਸਟੇਟਮੈਂਟ ਲਿਖਦੇ ਹਾਂ

```
for(a=1;a<=5;a++);
sum=sum+a;
```

ਕੇਵਲ **a=5** ਦੀ exit value ਹੀ sum ਵਿਚ ਸ਼ਾਮਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

(ਬੀ) ਗਾਇਬ ਅਰਧ ਵਿਰਾਮ ਚਿੰਨ੍ਹ (Missing semi colon)

ਹਰੇਕ C ਸਟੇਟਮੈਂਟ ਦਾ ਅੰਤ ਇਕ ਅਰਧ ਵਿਰਾਮ ਚਿੰਨ੍ਹ ਦੇ ਨਾਲ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਇਕ **Missing semi colon** ਕੰਪਾਈਲਰ ਵਿਚ ਕਾਫ਼ੀ ਗੜਬੜ ਕਰ ਦਿੰਦਾ ਹੈ ਅਤੇ ਇਸਦੇ ਸਿੱਟੇ ਵਜੋ ਇਕ ਗਲਤ ਸਦੇਸ਼ ਆ ਜਾਂਦਾ ਹੈ ਜਿਹੜਾ ਗੁੰਮਰਾਹ ਕਰਦਾ ਹੈ।ਮਿਸਾਲ ਲਈ, ਜੇਕਰ ਅਸੀ ਨਿਮਨਲਿਖਤ ਦੋ ਸਟੇਟਮੈਂਟਸ ਤੇ ਵਿਚਾਰ ਕਰਦੇ ਹਾਂ।

```
c = a + b
d = a / b;
```

ਇਥੇ ਕੰਪਾਈਲਰ d=a/b; ਨੂੰ. ਸਟੇਟਮੈਂਟ c = a+b ਦਾ ਭਾਗ ਸਮਝੇਗਾ ਅਤੇ bd ਨੂੰ ਇਕ ਵੇਰੀਏਬਲ ਦਾ ਨਾਮ ਸਮਝੇਗਾ। ਇਸ "undefined" ਨਾਮ ਦੇ ਕਾਰਣ "error message" ਦੂਜੀ ਲਾਈਨ ਵਿਚ ਚਲਾ ਜਾਵੇਗਾ। ਸਾਫ ਹੈ ਕਿ ਗਲਤੀ ਪਹਿਲੀ ਲਾਈਨ ਵਿਚ ਸੀ ਅਤੇ ਸੰਦੇਸ਼ ਦੂਜੀ ਲਾਈਨ ਵਿਚ ਪ੍ਰਦਰਸ਼ਿਤ ਹੋਇਆ ਜਿਥੇ ਕੋਈ ਗਲਤੀ ਨਹੀ ਸੀ, ਇਸ ਲਈ ਅਜਿਹੀਆਂ ਸਥਿਤੀਆਂ ਵਿਚ ਸਾਨੂੰ "semi colon" ਚਿੰਨ੍ਹ ਦੇ ਗਾਇਬ ਹੋਣ ਲਈ ਅਗਲੀ ਲਾਈਨ ਦੀ ਜਾਂਚ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ।

ਕੇਸ 3: ਗਾਇਬ ਬਰੇਸ਼ਾਂ/ਉਚਿਤ ਸਥਾਨ ਤੇ ਬਰੇਸਿਸ ਰਖਣੀਆਂ (Missing braces/braces at proper places)

ਕਈ ਵਾਰ ਅਸੀ ਬਰੇਸਿਸ ਪਾਉਣੀਆਂ ਭੁਲ ਜਾਂਦੇ ਹਾਂ ਵਿਸ਼ੇਸ਼ ਕਰ ਕੇ ਅੰਤਿਮ ਬਰੇਸਿਸ ਜਦੋਂ ਪ੍ਰੋਗਰਾਮ ਖਤਮ ਕਰਦੇ ਹਾਂ ਇਸਨੂੰ ਆਮਤੌਰ ਤੇ ਕੰਪਾਈਲਰ ਦੁਆਰਾ ਫੜਿਆ ਜਾਂਦਾ ਹੈ ਕਿਉਂਕਿ ਮੁਢਲੀਆਂ ਬਰੇਸਿਸ ਦੀ ਗਿਣਤੀ ਅੰਤਿਮ ਬਰੇਸਿਸ਼ ਦੇ ਬਰਾਬਰ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ। ਭਾਵੇਂ ਕਿ ਜੇਕਰ ਅਸੀ ਮੈਚਿੰਗ ਬਰੇਸਿਸ ਨੂੰ ਗਲਤ ਸਥਾਨ ਤੇ ਰਖਦੇ ਹਾਂ ਤਾਂ ਇਹ ਕੰਪਾਈਲਰ ਦੁਆਰਾ ਨਹੀ ਫੜੀਆਂ ਜਾਣਗੀਆਂ ਪਰੰਤੂ ਪ੍ਰੋਗਰਾਮ ਆਸ਼ਾਪ੍ਰਤਿਕੂਲ ਨਤੀਜੇ ਪੇਸ਼ ਕਰਦੇ ਹਨ। ਇਸ ਨੂੰ ਦੇਖਣ ਲਈ ਆਓ ਨਿਮਨਲਿਖਤ ਮਿਸਾਲ ਤੇ ਗੌਰ ਕਰੀਏ।

Statements without proper braces

```
for(a = 1;a <=5;a++)
sum1 = sum1 + a;
avg = sum1/5 ;
printf( "%d %d\n",sum1,avg);
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਇਹ ਪ੍ਰੋਗਰਾਮ sum1 ਦੀ ਗਣਨਾ ਕਰਦਾ ਹੈ ਅਤੇ 1 ਤੋਂ 5 ਤਕ ਜਾ ਕੇ (in step of one) ਔਸਤ ਕਰਦਾ ਹੈ ਅਤੇ ਫਿਰ ਮੁੱਲ ਪ੍ਰਿੰਟ ਕਰਦਾ ਹੈ। ਪਰੰਤੂ ਅਸਲ ਵਿਚ for loop ਕੇਵਲ ਪਹਿਲੇ ਸਟੇਟਮੈਂਟ ਦੇ ਨਾਲ sum1=sum1+a: ਕਰਦਾ ਹੈ ਅਤੇ ਇਸ ਲਈ ਸਟੇਟਮੈਂਟ ਔਸਤਨ avg=sum1/5; ਦਾ ਮੁਲਾਂਕਣ ਕੇਵਲ ਇਕ ਵਾਰ ਹੀ ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਹ ਲੂਪ ਤੋਂ ਬਾਹਰ ਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਕਰਕੇ ਬਰੇਸਿਸ ਸਹੀ ਜਗ੍ਹਾ ਤੇ ਲਗਾਉਣੀਆਂ ਬਹੁਤ ਜ਼ਰੂਰੀ ਹੁੰਦੀਆਂ ਹਨ ਜਿਸ ਤਰ੍ਹਾਂ ਇਹ ਹੇਠਾਂ ਦਰਸਾਇਆ ਗਿਆ ਹੈ

Statements with proper braces:

```
for(a = 1;a <=5;a++)
{
sum1 = sum1 + a:
avg = sum1/5 ;
}
printf("%d d\n",sum1,avg);
```

ਕੇਸ 4: ਮਿਸਿੰਗ ਕੋਟਸ (Missing quotes)

ਹਰ ਸਟਰਿੰਗ ਇਕ ਦੋਹਰੇ ਕੋਟਾਂ (" ") ਨਾਲ ਨੱਥੀ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ਜਦਕਿ ਇਕਲਾ ਚਿੰਨ੍ਹ (character) ਇਕਲੇ ਕੋਟ (' ') ਵਿਚ ਕੰਾਂਸਟੈਂਟ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਇਹਨਾ ਦੇ ਬਗੈਰ ਸਟਰਿੰਗ ਵੇਰੀਏਬਲ (string variable) ਨਾਮ ਵਜੋਂ ਸਮਝੇ ਜਾਣਗੇ

ਮਿਸਾਲ ਲਈ

```
if ( married = ="No")        /* No is a string */
status = "N"                 /* N is a character constant *\
```

ਇਥੇ No ਅਤੇ N ਨੂੰ ਵੇਰੀਏਬਲਸ ਵਜੋਂ ਵਿਵਹਾਰ ਵਿਚ ਲਿਆਂਦਾ ਜਾ ਰਿਹਾ ਹੈ ਅਤੇ ਇਸ ਲਈ, ਸੰਦੇਸ਼ 'undefined name' ਵਾਪਰ ਸਕਦਾ ਹੈ।

ਇਸ ਵਿਚ **"N"** ਦੀ ਜਗ੍ਹਾ ਤੇ **'N'** ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।

ਕੇਸ 5: ਅਸਾਇੰਨਮੈਂਟ ਦਾ ਸਹੀ ਵਰਤੋ (Use of = instead of = =)

ਮਿਸਾਲ ਲਈ

```
If ( a = 1)
count ++;
```

ਇਥੇ ਵੇਰੀਏਬਲ a ਨੂੰ 1 ਅਸਾਈਨਡ ਹੈ ਅਤੇ, ਕਿਉਂਕਿ a=1 ਸਹੀ (true) ਹੈ, ਪਰ count ਵਧ ਜਾਂਦਾ ਹੈ ਅਤੇ ਸਟੇਟਮੈਂਟ **a** ਉੱਤੇ, ਕੋਈ ਪਰਖ ਨਹੀ ਕਰਦੀ, ਪਰ count++; ਹਮੇਸ਼ਾ ਲਾਗੂ ਹੁੰਦਾ ਹੈ। For ਅਤੇ while ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟਸ ਵਿਚ ਵੀ ਇਹੋ ਜਿਹੀਆਂ ਗਲਤੀਆਂ ਹੋ ਸਕਦੀਆਂ ਹਨ। ਇਸ ਲਈ ਅਸਾਇੰਨਮੈਂਟ ਆਪਰੇਟਰ (assignment operator) ਦਾ ਸਹੀ ਇਸਤੇਮਾਲ **a==1 ਹੋਵੇ ਤਾਂ if ਸਟੇਟਮੈਂਟ** ਦੀ ਪਰਖ ਹੋਵੇਗੀ।

ਕੇਸ 6 : ਅਣਡਿਕਲੇਅਰ ਵੇਰੀਏਬਲਸ (undeclared variables)

ਸਾਨੂੰ ਵੇਰੀਏਬਲਸ ਅਤੇ ਉਸ ਦੀਆਂ ਕਿਸਮਾਂ ਨੂੰ ਵਰਤਣ ਤੋਂ ਪਹਿਲਾਂ ਡਿਕਲੇਅਰ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ, ਵਰਨਾ ਇਕ ਸੰਦੇਸ਼ undefine ਵੇਰੀਏਬਲਸ ਸਕਰੀਨ ਤੇ ਪ੍ਰਦਰਸ਼ਤ ਹੋਵੇਗਾ।

ਕੇਸ 7: ਔਪਰੇਟਰਸ ਦੀ ਪ੍ਰਮੁੱਖਤਾ ਨੂੰ ਭੁਲਣਾ (for getting the precedence of operators)

ਐਕਸਪ੍ਰੇਸ਼ਨ (expression) ਦਾ ਓਪਰੇਟਰਸ ਦੀ ਪ੍ਰਮੁੱਖਤਾ ਦੇ ਅਨੁਸਾਰ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ। ਸਟੇਟਮੈਂਟ ਤੇ ਵਿਚਾਰ ਕਰੋ।

```
If (value = product ( ) >= 5)
Fee = 0.05 * value;
```

ਇਹ ਗਲਤੀ ਅਸਾਇੰਨਡ ਔਪਰੇਟਰ ਦੀ ਤੁਲਨਾ ਵਿਚ ਰਿਲੇਸ਼ਨਲ ਔਪਰੇਟਰ (relational operator) ਦੀ ਉਤਮਤਾ ਦੇ ਕਾਰਨ ਹੁੰਦਾ ਹੈ। ਇਸ ਲਈ ਸੋਧ ਹੈ।

```
If ((value = product ( )) >= 5)
Fee = 0.05 * value;
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਕੇਸ 8: ਫੰਕਸ਼ਨ ਡਿਕਲੇਅਰ ਕਰਨ ਵਿਚ ਭੁਲਣਾ (forgetting to declare function)

ਹਰ ਇਕ ਫੰਕਸ਼ਨ ਜਿਸ ਨੂੰ ਬੁਲਾਇਆ ਜਾਦਾ ਹੈ calling ਫੰਕਸ਼ਨ ਵਿਚ, ਉਸ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਨਾ ਜਰੂਰੀ ਹੁੰਦਾ ਹੈ ਤਾਂ ਕਿ ਵਾਪਸ ਭੇਜੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲ ਦੀ ਟਾਈਪ ਅਤੇ ਉਹ ਫੰਕਸ਼ਨ ਇਕੋ ਜਿਹੇ ਹੋਣ।

ਮਿਸਾਲ ਲਈ, ਹੇਠਾਂ ਦਿਤਾ ਗਿਆ ਫੰਕਸ਼ਨ **double type** ਤੇ ਮੁੱਲ ਨੂੰ ਵਾਪਸ ਮੋੜ੍ਹਦਾ ਹੈ ਪਰੰਤੂ ਇਸ ਤੱਥ ਦਾ ਕਾਲਿੰਗ ਫੰਕਸ਼ਨ ਨੂੰ ਗਿਆਨ ਨਹੀ ਇਸ ਲਈ ਇਹ ***int*** type ਮੁੱਲ ਨੂੰ ਪ੍ਰਾਪਤ ਕਰਨਾ ਸਵੀਕਾਰ ਕਰਦਾ ਹੈ। ਇਸ ਲਈ ਜਦੋ ਵੀ ਇਸਦੀ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਇਕ ਗਲਤਾ ਵਾਲਾ ਸੰਦੇਸ਼ ਜਿਵੇਂ ਕਿ "redefinition" ਜਾਂ ਅਰਥਹੀਣ ਪਰਿਣਾਮ ਪ੍ਰਾਪਤ ਹੁੰਦਾ ਹੈ।

`main( )`
`{ float a = 12.05 ;`
`float b =7.35 ;`
`printf("%f\n " ,division(a,b));`
`}`
`double division(x,y)`
`float x, y;`
`{`
`return (x,y);`
`}`
 	
ਫੰਕਸ਼ਨ **division** ਹੋਰਾਂ ਵੇਰੀਏਬਲਸ ਦੀ ਹੀ ਤਰ੍ਹਾਂ **main( )** ਹੈ ਅਤੇ ਇਸ ਲਈ ਇਸਨੂੰ **main( )** ਵਿਚ **double** ਵਜੋਂ ਡਿਕਲੇਅਰ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ।
`division (x,y)`
`float x, y;`
`{ return(x/y) ;`
`}`
	ਇਸ ਮਿਸਾਲ ਵਿਚ x ਅਤੇ y ਫਲੋਟ ਟਾਈਪ (float type) ਦੇ ਹਨ ਅਤੇ x/y ਦਾ ਪਰਿਣਾਮ ਵੀ ਫਲੋਟ ਹੀ ਹੁੰਦਾ ਹੈ, ਫੰਕਸ਼ਨ ਕੇਵਲ ਪੂਰਣ-ਅੰਕ ਮੁਲ (integer value) ਮੋੜ੍ਹਦਾ ਹੈ ਕਿਉਂ ਜੋ ਕੋਈ type specifier, ਫੰਕਸ਼ਨ ਪਰਿਭਾਸ਼ਾ ਵਿਚ ਨਹੀ ਦਿਤਾ ਜਾਂਦਾ। ਪਰ ਇਹ ਵੀ ਗਲਤ ਹੁੰਦਾ ਹੈ।ਇਸ ਲਈ ਫੰਕਸ਼ਨ ਡਿਕਲੇਅਰ ਵਿਚ type specifier ਦਾ ਨਿਸ਼ਚਿਤ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਤਾਂ ਜੋ ਵਾਪਸ ਭੇਜੇ ਜਾਣ ਵਾਲੇ ਮੁੱਲ ਦੀ ਟਾਈਪ ਇਕੋ ਜਿਹੀ ਹੋਵੇ।

ਕੇਸ 9: scanf ਪੈਰਾਮੀਟਰ ਵਿਚ ਔਪਰੇਟਰ & ਦਾ ਨਾ ਹੋਣਾ

ਜੇਕਰ ਵੇਰੀਏਬਲਜ਼ ਕੋਡ ਨੂੰ ਪੂਰਣ-ਅੰਕ (integer) ਵਜੋਂ ਡਿਕਲੇਅਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਤਾਂ ਸਟੇਟਮੈਂਟ **scanf("%d", a);** ਗਲਤ ਹੁੰਦਾ ਹੈ। ਸਟੇਟਮੈਂਟ **scanf("%d", &a);** ਸਹੀ ਹੁੰਦਾ ਹੈ।

ਕੇਸ 10: ਐਰੇ ਦੇ ਬੰਧਨਾਂ ਨੂੰ ਪਾਰ ਕਰਨਾ

ਪਹਿਲੀ subscript or index ਜਾਂ ਐਰੇ ਵਿਚ ਸੂਚੀ 0 ਹੁੰਦੀ ਹੈ। ਇਕ ਆਮ ਗਲਤੀ 1 ਤੋਂ subscript or index ਨੂੰ ਸ਼ੁਰੂ ਕਰਨਾ ਹੁੰਦਾ ਹੈ।

ਮਿਸਾਲ ਲਈ :

```
int x[5],sum,a;
sum = 0;
for(a = 1;a <=5;a++)
sum = sum +x[a];
```

ਭਾਗ ਇਹ ਐਰੇ **x** ਦੇ ਅੰਸ਼ ਦਾ ਸਹੀ ਜੋੜ ਨਹੀ ਲਭੇਗਾ। **for loop** ਐਕਸਪ੍ਰੇਸ਼ਨ ਸਹੀ ਹੋਵੇਗਾ।

```
for(a = 0;a <5;a++)
```


Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਸ਼ਬਦਾਵਲੀ (Glossary)

(ਏ)	
Analog Data (ਅਨਾਲੌਗ ਡਾਟਾ)	ਡਾਟਾ ਜਿਸ ਦੀ ਇਕ ਰੇਂਜ ਵਿਚ ਕੋਈ ਵੀ ਮੁੱਲ ਹੁੰਦਾ ਹੈ ਅਤੇ ਜਿਹੜੀ ਲਗਾਤਾਰ ਬਦਲ ਸਕਦੀ ਹੈ ਦਿਨ ਦਾ ਸਮਾਂ ਕਲਾਕ ਹੈਂਡ ਦੁਆਰਾ ਪ੍ਰਤਿਨਿਧ ਹੁੰਦਾ ਹੈ ਜਾਂ ਤਾਪਮਾਨ ਇਕ ਤਰਲ ਥਰਮਾਮੀਟਰ ਦੁਆਰਾ ਪ੍ਰਤਿਨਿਧ ਹੁੰਦਾ ਹੈ ਜੋ ਕਿ ਅਨਾਲੌਗ ਡਾਟਾ ਦੀਆਂ ਮਿਸਾਲਾਂ ਹਨ।
Analog Signal (ਅਨਾਲੌਗ ਸਿਗਨਲ)	ਸਿਗਨਲ ਦੀ ਇਕ ਕਿਸਮ ਜੋ ਉਸ ਡਾਟਾ ਨੂੰ ਇਨਕੋਡ ਕਰਦੀ ਹੈ ਜੋ ਤਾਰ ਜਾਂ ਹਵਾ ਰਾਹੀਂ ਸੰਚਾਰ ਹੁੰਦਾ ਹੈ ਅਤੇ ਸਾਂਝੇ ਤੌਰ ਤੇ ਆਸੀਲੇਟਿੰਗ (oscilating) ਤਰੰਗਾ ਨਾਲ ਪ੍ਰਤਿਨਿਧ ਹੁੰਦਾ ਹੈ। ਅਨਾਲੌਗ ਸੰਕੇਤ ਇਕ ਰੇਂਜ ਵਿਚ ਕੋਈ ਵੀ ਮੁੱਲ ਲੈ ਸਕਦੇ ਹਨ ਅਤੇ ਮੁੱਲਾਂ ਦੇ ਦਰਮਿਆਨ ਇਕ ਸਾਰ ਬਦਲਦੇ ਹਨ। ਇਕ ਅਨਾਲੌਗ ਸਿਗਨਲ ਅਨਾਲੌਗ ਜਾਂ ਹਿੰਦਸੇ ਡਾਟਾ ਦਾ ਸੰਚਾਰ ਕਰ ਸਕਦੇ ਹਨ। ਮਿਸਾਲ ਲਈ, ਇਕ ਰੇਡਿਓ ਸਟੇਸ਼ਨ ਅਨਾਲੌਗ ਸੰਕੇਤਾਂ ਦੀ ਵਰਤੋਂ ਨਾਲ ਅਨਾਲੌਗ ਸੰਗੀਤ ਭੇਜਦਾ ਹੈ ਜਦਕਿ ਇਸ ਮਾਡਮ ਡਿਜੀਟਲ ਡਾਟਾ ਨੂੰ ਅਨਾਲੌਗ ਸੰਕੇਤਾਂ ਦੀ ਵਰਤੋਂ ਨਾਲ ਹਿੰਦਸਾ ਡਾਟਾ ਸੰਚਾਰ ਕਰਦਾ ਹੈ।
Argument (ਦਲੀਲ)	ਜਦੋਂ ਇਕ ਫੰਕਸ਼ਨ ਨੂੰ ਬੁਲਾਉਣਾ ਹੋਵੇ ਤਾਂ ਫੰਕਸ਼ਨ ਨੂੰ ਭੇਜੇ ਗਏ ਅਸਲ ਮੁੱਲਾਂ ਨਾਲ ਸੰਬੰਧਤ ਹੈ।
Argument matching (ਦਲੀਲ ਮੇਚਿੰਗ)	ਇਸ ਗੱਲ ਦਾ ਫੈਸਲਾ ਕਰਨ ਦੀ ਪ੍ਰਕ੍ਰਿਆ ਕਿ ਫੰਕਸ਼ਨਸ ਦੇ ਕਿਹੜੇ ਸੈੱਟ ਨੂੰ ਵਿਸ਼ੇਸ਼ ਨਾਮ ਦਾ ਫੰਕਸ਼ਨ ਤੈਅ ਕਰਨਾ ਹੈ, ਫੰਕਸ਼ਨ ਮੰਗ ਵਿਚ ਦਲੀਲ ਦਿੰਦੇ ਹਏ। ਮੇਲ ਇਕ ਫੰਕਸ਼ਨ ਮੰਗ ਵਿਚ ਦਲੀਲਾਂ ਦਿੰਦੇ ਹਨ।
ASCII	ਅਰਥਮੈਟਿਕ ਕੋਡ ਚਿੰਨ੍ਹਾਂ ਦਾ ਮਿਆਰ ਹੁੰਦੇ ਹਨ। ਸ਼ਾਬਦਿਕ ਤੌਰ ਤੇ ASCII (American standard code) ਸੂਚਨਾ ਅੰਦਰ ਤਬਾਦਲੇ ਲਈ ਮਿਆਰੀ ਕੋਡ ਲਈ ਹੁੰਦਾ ਹੈ। ਇਹ ਪੂਰਣ ਅੰਕ ਅਨੁੰਵਾਦ ਕੋਡ ਲਈ ਬਹੁਤ ਆਮ ਚਿੰਨ੍ਹ ਹੁੰਦਾ ਹੈ।
Assignment (ਸਮਰਪਣ)	ਪੂਰਵ-ਮੌਜੂਦਾ ਆਬਜੈਕਟ ਲਈ ਇਕ ਮੁੱਲ ਪ੍ਰਦਾਨ ਕਰਨ ਦੀ ਪ੍ਰਕ੍ਰਿਆ।
Assignment operator (ਐਸਾਈਨ ਸੰਚਾਲਕ)	ਐਸਾਈਨਮੈਂਟ ਕਰਨ ਦਾ ਔਪਰੇਟਰ
ਬੀ	
Backbone network (ਬੈਕਬੋਨ ਨੈੱਟਵਰਕ)	ਇਕ ਨੈੱਟਵਰਕ ਜਿਹੜਾ ਅਨੇਕਾਂ LANs ਨੂੰ ਜੋੜਣ ਲਈ ਬੈਕਬੋਨ ਦਾ ਕੰਮ ਕਰਦਾ ਹੈ।
Bandwidth (ਬੈਂਡਵਿਡਥ)	ਇਕ ਸਿਗਨਲ ਦਾ ਪ੍ਰਸਾਰ ਕਰਨ ਵਾਲੇ ਮਾਧਿਅਮ ਦੀ ਸਮਰਥਾ।
Boolean (ਬੂਲੀਅਨ)	C ਕੀ-ਵਰਡ ਬੂਲੀਅਨ ਡਾਟਾ ਟਾਈਪ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ
Bridge (ਬ੍ਰਿਜ)	ਡਿਵਾਈਸ ਜਿਹੜਾ ਦੋ ਨੈੱਟਵਰਕਾਂ ਨੂੰ ਇਕਠਿਆਂ ਜੋੜਦਾ ਹੈ।
ਸੀ	
Cable (ਕੇਬਲ)	ਤਾਂਬੇ ਦੀ ਤਾਰ ਜਾਂ ਸੁਰੱਖਿਅਤ ਕਵਰ ਵਿਚ ਲਪੇਟੀ ਆਪਟੀਕਲ ਫਾਈਬਰ ਦਾ ਸੰਚਾਰ ਮਾਧਿਅਮ।
Call (ਕਾਲ)ਵਿਧੀ	ਪ੍ਰੋਗਰਾਮ, ਰੂਟੀਨ ਜਾਂ ਓਪ-ਰੂਟੀਨ ਦੇ ਕੰਟਰੋਲ ਨੂੰ ਬਦਲਣਾ।
Call by reference ਹਵਾਲੇ ਨਾਲ ਬੁਲਾਉਣਾ	ਇਕ ਫੰਕਸ਼ਨ ਵੱਲ ਦਲੀਲ, ਅਤੇ ਪੁਆਂਇੰਟਰ ਗੁਜ਼ਾਰਨਾ। ਫਿਰ ਦਲੀਲ ਮੁੱਲ ਬਦਲ ਸਕਦਾ ਹੈ।
Call by value ਮੁੱਲ ਦੁਆਰਾ ਬੁਲਾਉਣਾ	ਇਕ ਫੰਕਸ਼ਨ ਨੂੰ ਦਲੀਲ ਦੀ ਕਾਪੀ ਭੇਜਣੀ। ਫੰਕਸ਼ਨ for ਦਲੀਲ ਮੁੱਲ ਨਹੀਂ ਬਦਲ ਸਕਦਾ। C ਦੀ ਵਰਤੋਂ ਮੁੱਲ ਦਲੀਲ ਰਾਹੀਂ ਹੁੰਦੀ ਹੈ। ਪਰੰਤੂ ਹਵਾਲੇ ਦੇ ਰਾਹੀਂ ਵੀ ਹੁੰਦੀ ਹੈ
Carriage return character (ਕੈਰਿਜ ਰਿਟਰਨ ਕਰੈਕਟਰ)	ਕੈਰਿਜ ਰਿਟਰਨ ਕਰੈਕਟਰ, ਇਕ ਚਿੰਨ੍ਹ ਜਿਹੜਾ ਪਰਿਣਾਮ ਧਾਰਾ ਵਿਚ ਸੰਕੇਤ ਦਿੰਦਾ ਹੈ ਕਿ ਪ੍ਰਿੰਟਿੰਗ ਉਸ ਭੌਤਿਕ ਰੇਖਾ ਤੋਂ ਸ਼ੁਰੂ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ਜਿਸ ਵਿਚ ਕੈਰਿਜ ਵਾਪਸੀ ਦੇ ਚਿੰਨ੍ਹ ਵਾਪਰਦੇ ਹਨ। ਕੈਰਿਜ ਵਾਪਸੀ ਇਕ ਚਿੰਨ੍ਹ ਹੈ ਜੋ '\r' ਦੁਆਰਾ C ਭਾਸ਼ਾ ਵਿਚ ਦਿਤਾ ਗਿਆ ਹੈ।
Case clause (ਕੇਸ ਖੰਡ)	ਇਕ ਸਵਿਚ ਸਟੇਟਮੈਂਟ ਵਿਚ, ਕੇਸ ਪੱਧਰ ਦਾ ਅਨੁਸਰਣ ਕਿਸੇ ਵੀ ਸੰਖਿਆ ਸਟੇਟਮੈਂਟਸ ਦੁਆਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Case label (ਕੇਸ ਲੇਬਲ)	ਸ਼ਬਦ **ਕੇਸ** ਦਾ ਅਨੁਸਰਣ ਇਕ ਕੰਨਸਟੈਂਟ ਐਕਸਪ੍ਰੇਸ਼ਨ ਅਤੇ ਵਿਸ਼ਰਾਮ ਚਿੰਨ੍ਹ (:) ਦੁਆਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਸਿਲੈਕਟਰ ਵਲੋਂ ਕੰਨਸਟੈਂਟ ਐਕਸਪ੍ਰੇਸ਼ਨ ਦੇ ਮੁੱਲ ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਲੇਬਲ ਦੀ ਅਨੁਸਰਣ ਕਰਨ ਵਾਲੇ ਸਟੇਟਮੈਂਟਸ ਦੀ ਪ੍ਰਕ੍ਰਿਆ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
Character (ਚਿੰਨ੍ਹ)	(1) ਅੱਖਰ, ਹਿੰਦਸਾ ਜਾਂ ਹੋਰ ਚਿਨ੍ਹ ਜਿਹੜੇ ਡਾਟਾ ਦੇ ਪ੍ਰਗਟਾਵਾ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। (2) ਇਕ ਜਾਂ ਵਧੇਰੇ ਬਾਈਟਾਂ ਦਾ ਸਾਰ ਜੋ ਇਕਹਰੇ ਗ੍ਰਾਫਿਕ ਚਿਨ੍ਹ ਜਾਂ ਕੰਟਰੋਲ ਕੋਡ ਦਾ ਪ੍ਰਗਟਾਵਾ ਕਰਦਾ ਹੈ।
Character constant	ਕਾਨਸਟੈਂਟ ਲੋਪ ਚਿੰਨ੍ਹ ਵਿਚ ਨਥੀ ਇਕ ਚਿੰਨ੍ਹ ਜਾਂ ਬਚਾਉ ਸਾਰ
Character set ਚਿੰਨ੍ਹ ਸੈੱਟ	1)ਚਿੰਨ੍ਹ ਸੈੱਟ ਭਿੰਨ-ਭਿੰਨ ਚਿੰਨ੍ਹਾਂ ਦਾ ਅਨੰਤ ਸੈੱਟ (finite set) ਜਿਹੜਾ ਦਿਤੇ ਗਏ ਮੰਤਵ ਲਈ ਮੁਕੰਮਲ ਹੁੰਦਾ ਹੈ। ਮਿਸਾਲ ਲਈ, ਚਿੰਨ੍ਹ ਸੈੱਟ ISO ਸਟੈਂਡਰਡ 646 ਵਿਚ, 7 ਬਿਟ ਕੋਡ ਵਾਲੇ ਚਿੰਨ੍ਹ ਸੈੱਟ ਚਿੰਨ੍ਹ ਸੂਚਨਾ ਪ੍ਰਕ੍ਰਿਆ ਅੰਤਰ ਤਬਦੀਲੀ (2) ਇਕ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਲਈ ਸਮੂਹ ਵੱਧ ਚਿੰਨ੍ਹ ਜਾਂ ਇਕ ਕੰਪਿਊਟਰ ਸਿਸਟਮ ਲਈ (3) ਵਿਸ਼ੇਸ਼ ਕਾਰਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਚਿੰਨ੍ਹਾਂ ਦਾ ਇਕ ਸਮੂਹ, ਮਿਸਾਲ ਲਈ, ਚਿੰਨ੍ਹਾਂ ਦਾ ਇਕ ਸੈੱਟ ਜੋ ਪ੍ਰਿੰਟਰ ਛਾਪ ਸਕਦਾ ਹੈ।
Character string ਚਿੰਨ੍ਹ ਸਟਰਿੰਗ	ਖਤਮ ਕੀਤੇ ਗਏ ਚਿੰਨ੍ਹਾਂ ਦਾ ਨਿਰੰਤਰ ਸਾਰ ਜਿਸ ਵਿਚ ਪਹਿਲਾਂ ਵਿਅਰਥ ਬਾਈਟ ਸ਼ਾਮਲ ਹੈ
Character variable ਚਿੰਨ੍ਹ ਪਰਿਵਰਤਤ	ਚਿੰਨ੍ਹ ਡਾਟਾ ਮੱਦ ਦਾ ਨਾਮ ਜਿਸ ਦਾ ਮੁੱਲ ਅਸਾਈਨ ਹੋਵੇ ਜਾਂ ਪ੍ਰੋਗਰਾਮ ਲਾਗੂਕਰਨ ਸਮੇਂ ਬਦਲਿਆ ਗਿਆ ਹੋਵੇ।
Client/server (ਗ੍ਰਾਹਕ/ਸਰਵਰ)	ਇਕ ਨੈੱਟਵਰਕ ਸਿਸਟਮ ਜਿਸ ਵਿਚ ਇਕ ਜਾਂ ਵਧੇਰੇ ਫਾਈਲ ਸਰਵਰ ਉਪਲਬਧ ਹੋਣ ਜੋ ਸੇਵਾਵਾਂ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਣ। ਜਿਵੇਂ ਕਿ ਨੈੱਟਵਰਕ ਪ੍ਰਬੰਧ, ਪ੍ਰਯੋਗ ਅਤੇ ਕੇਂਦਰੀ ਡਾਟਾ ਭੰਡਾਰ ਵਰਕਸਟੇਸ਼ਨ (ਗ੍ਰਾਹਕਾਂ ਲਈ)
Coded Character Set ਕੋਡਿਡ ਚਿੰਨ੍ਹ ਸੈੱਟ	(1) ਗ੍ਰਾਫਿਕਸ ਚਿੰਨ੍ਹਾਂ ਦਾ ਇਕ ਸੈੱਟ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਕੋਡ ਪੁਆਇੰਟ ਸਮਰਪਣ। ਸੈੱਟ ਵਿਚ ਕੁਝ ਇਕ ਚਿੰਨ੍ਹ ਸ਼ਾਮਲ ਹੋ ਸਕਦੇ ਹਨ ਨਾ ਕਿ ਸੰਭਵ ਚਿੰਨ੍ਹਾਂ ਦੀ ਕੁਲ ਗਿਣਤੀ। ਕੁਝ ਕੋਡ ਪੁਆਇੰਟ ਗੈਰ-ਐਸਾਈਨ ਵੀ ਹੋ ਸਕਦੇ ਹਨ।(2) ਕੋਡ ਕੀਤੇ ਸੈੱਟ ਜਿਨ੍ਹਾਂ ਦੇ ਅੰਸ਼ ਇਕਹਿਰੇ ਚਿੰਨ੍ਹ ਹੁੰਦੇ ਹਨ। ਮਿਸਾਲ ਲਈ, ਇਕ ਵਰਣਮਾਲਾ ਦੇ ਸਮੂਹ ਚਿੰਨ੍ਹ।
Comma expression (ਵਿਸ਼ਰਾਮ ਚਿੰਨ੍ਹ ਐਕਸਪ੍ਰੇਸ਼ਨ)	ਇਕ ਐਕਸਪ੍ਰੇਸ਼ਨ ਜਿਸ ਵਿਚ ਇਕ ਕੌਮੇ ਜਾਂ ਵਿਸ਼ਰਾਮ ਚਿੰਨ੍ਹ ਦੁਆਰਾ ਵੱਖ ਕੀਤੇ ਗਏ ਦੋ ਉਪਰੈਂਡਸ, ਹੁੰਦੇ ਹੋਣ। ਭਾਵੇਂ ਕੰਪਾਈਲਰ ਐਕਸਪ੍ਰੇਸ਼ਨ ਦੁਆਰਾ ਦੋਹਾਂ ਔਪਰੈਂਡਾਂ ਦਾ ਮੁਲਾਂਕਣ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਐਕਸਪ੍ਰੇਸ਼ਨ ਦਾ ਮੁੱਲ ਸਹੀ ਔਪਰੈਂਡ ਦਾ ਮੁੱਲ ਹੁੰਦਾ ਹੈ। ਜੇਕਰ ਖੱਬਾ ਔਪਰੈਂਡ ਮੁੱਲ ਉਤਪਾਦਨ ਕਰਦਾ ਹੈ ਤਾਂ ਕੰਪਾਈਲਰ ਇਸ ਮੁੱਲ ਨੂੰ ਨਕਾਰਦਾ ਹੈ। ਵਿਸ਼ੇਸ਼ ਕਰਕੇ ਖੱਬਾ ਔਪਰੈਂਡ ਇਕ ਕੌਮੇ ਦਾ ਐਕਸਪ੍ਰੇਸ਼ਨ ਬਿਆਨ ਕਰਨ ਲਈ ਹੋਰ ਗੈਰ-ਪ੍ਰਭਾਵ ਪੈਦਾ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
Command (ਕਮਾਂਡ)	ਇਕ ਪ੍ਰੋਗਰਾਮ ਚਲਾਉਣ ਲਈ ਜਾਂ ਫੰਕਸ਼ਨ ਕਰਨ ਲਈ ਦਿਤੀ ਗਈ ਹਿਦਾਇਤ। ਜਦੋਂ ਪੈਰਾਮੀਟਰ, ਦਲੀਲਾਂ, ਫਲੈਗ ਜਾਂ ਹੋਰ ਉਪਰੈਂਡਾਂ ਦਾ ਕੰਮਾਂਡ ਨਾਲ ਸੰਬੰਧ ਬਣਦਾ ਹੈ ਤਾਂ ਪਰਿਣਾਮ ਸਵਰੂਪੀ ਚਿੰਨ੍ਹ ਤਾਰ ਇਕ ਇਕਹਰੀ ਕਮਾਂਡ ਹੁੰਦੀ ਹੈ।
CD ROM (ਸੀ-ਡੀ ਰੋਮ)	ਉੱਚ ਸਮਰੱਥਾ, ਆਪਟੀਕਲੀ ਰੀਡ ਕੰਪੈਕਟ ਡਿਸਕ ਦੇ ਰੂਪ ਵਿਚ ਹੀ ਕੇਵਲ ਮੈਮੋਰੀ ਪੜੀ ਜਾਂਦੀ ਹੈ।
Compilation unit (ਕੰਪਾਈਲੇਸ਼ਨ ਯੂਨਿਟ)	ਕੰਪਿਊਟਰ ਪ੍ਰੋਗਰਾਮ ਦਾ ਇਕ ਭਾਗ ਜੋ ਸਹੀ ਤੌਰ ਤੇ ਕੰਪਾਈਲ ਕਰਨ ਲਈ ਚੋਖਾ ਸੰਪੂਰਨ ਹੁੰਦਾ ਹੈ। (2) ਇਕ ਇਕਹਰੀ ਸੰਗ੍ਰਹਿ ਵਾਲੀ ਫਾਈਲ ਅਤੇ ਇਸਦੇ ਸਮੂਹ ਸੰਬੰਧੀ ਜਿਸ ਵਿਚ ਫਾਈਲਾਂ ਸ਼ਾਮਲ ਹੁੰਦੀਆਂ ਹਨ। (3) ਹਾਈ ਲੈਵਲ ਭਾਸ਼ਾ ਸਟੇਟਮੈਂਟਸ ਦੀ ਸੁਤੰਤਰ ਪਾਲਣਾ ਵਾਲੇ ਸਾਰ। ਹਰੇਕ ਹਾਈ ਲੈਵਲ ਉਪਜ ਦੇ ਆਪਣੇ ਹੀ ਨਿਯਮ ਹੁੰਦੇ ਹਨ ਜੋ ਕੰਪਾਈਲੇਸ਼ਨ ਨੂੰ ਯੂਨਿਟ ਬਣਾਉਂਦੇ ਹਨ।
Compile (ਕੰਪਾਈਲ ਸੰਗ੍ਰਹਿ)	ਸੋਰਸ ਕੋਡ ਤੋ ਆਬਜੈਕਟ ਰੂਪ ਵਿਚ ਰੂਪਾਂਤਰ ਕਰਨਾ।
Compiler options (ਸੰਗ੍ਰਹਿ ਵਿਕਲਪ)	ਕੀ-ਵਰਡ ਜਿਹੜੇ ਕੰਪਾਈਲੇਸ਼ਨ ਦੇ ਕੁਝ ਵਿਸ਼ੇਸ਼ ਪਹਿਲੂਆਂ ਤੇ ਕੰਟਰੋਲ ਕਰਨ ਲਈ ਨਿਸ਼ਚਿਤ ਕੀਤੇ ਜਾ ਸਕਦੇ ਹਨ।
Composite ਮਿਸ਼ਰਣ (ਕੰਪੋਜ਼ਿਟ)	ਇਕਹਿਰੇ ਫਰੇਮ ਜਾਂ ਪ੍ਰਦਰਸ਼ਨ ਵਿਚ, ਦੋ ਜਾਂ ਵਧੇਰੇ ਫਿਲਮ, ਵੀਡੀਓ ਜਾਂ ਇਲੈਕਟ੍ਰਾਨਿਕ ਪ੍ਰਤਿਬਿੰਬਾਂ ਦਾ ਸੁਮੇਲ।
Concentrator (ਕੰਨਸਨਟਰੇਟਰ)	ਅਜਿਹਾ ਡਿਵਾਈਸ ਜੋ ਵਰਕਸਟੇਸ਼ਨਾਂ ਸਰਵਰ ਤੇ ਸੀਮਾਵਾਂ ਤੇ ਕੇਬਲਾਂ ਲਈ ਕੇਂਦਰੀ ਕੁਨੈਕਸ਼ਨ ਪੁਆਇੰਟ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਬਹੁਤ ਸਾਰੇ ਕੰਨਸਨਰੇਟਰਾਂ ਵਿਚ ਇਲੈਕਟਰਿਕ ਸਿਗਨਲ ਜਿਹੜੇ ਉਹ ਪ੍ਰਾਪਤ ਕਰਦੇ ਹਨ ਨੂੰ ਐਂਪਲੀਫਾਈ *(amplify)* ਕਰਨ ਦੀ ਯੋਗਤਾ ਹੁੰਦੀ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

Condition (ਕੰਡੀਸ਼ਨ)	(1) ਇਕ ਸੰਪਰਕੀ ਐਕਸਪ੍ਰੇਸ਼ਨ ਜਿਹੜਾ ਸਹੀ ਜਾਂ ਗਲਤ ਦੇ ਮੁੱਲ ਦਾ ਮੁਲਾਂਕਣ ਕਰਦਾ ਹੈ। (2) ਇਕ ਅਪਵਾਦ ਜਿਹੜਾ ਭਾਸ਼ਾ ਵਾਤਾਵਰਣ ਦੁਆਰਾ ਪਛਾਣਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਸ ਪ੍ਰਕਾਰ ਯੂਜ਼ਰ ਨੂੰ ਸਰਗਰਮ ਕਰਦਾ ਹੈ ਅਤੇ ਭਾਸ਼ਾ ਕੰਡੀਸ਼ਨਸ ਦੇ ਹੈਂਡਲਰ ਨੂੰ ਯੋਗ ਬਣਾਉਂਦਾ ਹੈ। ਆਮ ਪ੍ਰੋਗਰਾਮ ਲਈ ਕੋਈ ਵੀ ਬਦਲਾਵ ਇਕ ਪ੍ਰਯੋਗ ਹੁੰਦਾ ਹੈ। ਸ਼ਰਤਾਂ ਹਾਰਡਵੇਅਰ ਜਾਂ ਸੰਚਾਲਤ ਸਿਸਟਮ ਦੁਆਰਾ ਫੜ੍ਹੀਆਂ ਜਾਂਦੀਆਂ ਹਨ ਅਤੇ ਪਰਿਣਾਮ ਹੁੰਦਾ ਹੈ ਰੁਕਾਵਟ। ਉਹ ਭਾਸ਼ਾ-ਵਿਸ਼ੇਸ਼ ਪੈਦਾ ਹੋਏ ਕੋਡ ਜਾਂ ਭਾਸ਼ਾ ਲਾਇਬ੍ਰੇਰੀ ਕੋਡ ਦੁਆਰਾ ਵੀ ਫੜ੍ਹੇ ਜਾ ਸਕਦੇ ਹਨ।
Conditional compilation directive (ਕੰਡੀਸ਼ਨਲ ਕੰਪਾਈਲੇਸ਼ਨ ਡਾਇਰੈਕਟਿਵ)	ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਨਿਰਦੇਸ਼ ਦਿੰਦਾ ਹੈ ਕਿ ਇਕ ਨਿਸ਼ਚਤ ਸਥਿਤੀ ਕੰਪਾਈਲੇਸ਼ਨ ਡਾਇਰੈਕਟਿਵ ਦੇ ਮੁਲਾਂਕਣ ਤੇ ਨਿਰਭਰ ਕਰਦੇ ਹੋਏ ਫਾਈਲ ਵਿਚ (ਕੰਡੀਸ਼ਨਲ ਕੰਪਾਈਲੇਸ਼ਨ ਡਾਇਰੈਕਟਿਵ) ਨਿਸ਼ਚਿਤ ਸੋਰਸ ਕੋਡ ਦੀ ਪ੍ਰਕ੍ਰਿਆ ਨੂੰ ਹੀ ਪ੍ਰੀ-ਪ੍ਰੋਸੈਸਰ ਕੀਤਾ ਜਾਵੇ।
Conditional expression (ਕੰਡੀਸ਼ਨਲ ਐਕਸਪ੍ਰੇਸ਼ਨ)	ਇਕ ਸੰਯੁਕਤ ਐਕਸਪ੍ਰੇਸ਼ਨ ਜਿਸ ਵਿਚ ਇਕ ਕੰਡੀਸ਼ਨ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ (ਪਹਿਲਾ ਐਕਸਪ੍ਰੇਸ਼ਨ) ਐਕਸਪ੍ਰੇਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਤਦ ਹੁੰਦਾ ਹੈ ਜੇਕਰ ਕੰਡੀਸ਼ਨ ਦਾ ਗੈਰ ਜ਼ੀਰੋ ਮੁੱਲ ਹੋਵੇ (ਦੂਜਾ ਐਕਸਪ੍ਰੇਸ਼ਨ) ਅਤੇ ਐਕਸਪ੍ਰੇਸ਼ਨ ਦਾ ਮੁਲਾਂਕਣ ਤਦ ਹੁੰਦਾ ਹੈ ਜੇਕਰ ਕੰਡੀਸ਼ਨ ਦਾ ਮੁੱਲ ਜ਼ੀਰੋ (ਤੀਜਾ ਐਕਸਪ੍ਰੇਸ਼ਨ) ਹੋਵੇ।
const (ਕਾਂਨਸਟ)	(1) ਇਕ ਡਾਟਾ ਆਬਜੈਕਟ ਦਾ ਗੁਣ ਆਰੋਪਣ ਜਿਹੜਾ ਡਿਕਲੇਅਰ ਕਰਦਾ ਹੈ ਕਿ ਆਬਜੈਕਟ ਕਦੇ ਨਹੀਂ ਬਦਲਿਆ ਜਾ ਸਕਦਾ। (2) ਇਕ ਫੰਕਸ਼ਨ ਦਾ ਗੁਣ ਆਰੋਪਣ ਜਿਹੜਾ ਡਿਕਲੇਅਰ ਕਰਦਾ ਹੈ ਕਿ ਫੰਕਸ਼ਨ ਵਲੋਂ ਇਸਦੀ ਸ਼੍ਰੇਣੀ ਦੇ ਡਾਟਾ ਮੈਂਬਰਾਂ ਵਿਚ ਸੋਧ ਨਹੀਂ ਲਿਆਂਦੀ ਜਾਵੇਗੀ।
constant (ਕਾਂਨਸਟੈਂਟ)	(1) ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾ ਵਿਚ, ਇਕ ਭਾਸ਼ਾ ਆਬਜੈਕਟ ਜਿਹੜਾ ਕੇਵਲ ਇਕ ਵਿਸ਼ੇਸ਼ ਮੁੱਲ ਹੀ ਲੈਂਦਾ ਹੈ। (2) ਇਕ ਮੁੱਲ ਦੇ ਨਾਲ ਡਾਟਾ ਮੱਦ ਜੋ ਕਿ ਪ੍ਰੋਗਰਾਮ ਚਲਣ ਦੌਰਾਨ ਬਦਲਿਆ ਨਹੀਂ ਜਾ ਸਕਦਾ।
constant expression (ਕੌਂਸਟੈਂਟ ਐਕਸਪ੍ਰੇਸ਼ਨ)	ਇਕ ਐਕਸਪ੍ਰੇਸ਼ਨ ਜਿਸਦਾ ਉਹ ਮੁੱਲ ਹੁੰਦਾ ਹੈ ਜਿਸਦਾ ਫੈਸਲਾ ਕੰਪਾਈਲੇਸ਼ਨ ਦੇ ਦੌਰਾਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਜਿਹੜਾ ਪ੍ਰੋਗਰਾਮ ਦੇ ਚਲਣ ਦੌਰਾਨ ਬਦਲਦਾ ਨਹੀਂ।
Control character (ਕੰਟਰੋਲ ਚਿੰਨ੍ਹ)	(1) ਇਕ ਚਿੰਨ੍ਹ ਜਿਸਦਾ ਵਾਪਰਣਾ ਇਕ ਵਿਸ਼ੇਸ਼ ਸੰਦਰਭ ਦੇ ਦੌਰਾਨ ਚਿੰਨ੍ਹ ਤੋਂ ਇਲਾਵਾ ਹੋਰ ਚਿੰਨ੍ਹ ਜਿਹੜਾ ਟੈਕਸਟ ਦੀ ਰਿਕਾਰਡਿੰਗ, ਪ੍ਰੋਸੈਸਿੰਗ, ਸੰਚਾਰ ਜਾਂ ਦੁਭਾਸ਼ਾ ਨੂੰ ਪ੍ਰਭਾਵਿਤ ਕਰਦਾ ਹੈ।
Control statement (ਕੰਟਰੋਲ ਸਟੇਟਮੈਂਟ)	ਇਕ ਭਾਸ਼ਾ ਸਟੇਟਮੈਂਟ ਜਿਹੜਾ ਲਾਗੂ ਕਰਨ ਦੇ ਆਮ ਮਾਰਗ ਨੂੰ ਬਦਲਦਾ ਹੈ।
Cursor (ਕਰਸਰ)	ਡਾਟਾ ਸਟਰਕਚਰ ਵਿਚ ਇਕ ਵਿਸ਼ੇਸ਼ ਸਥਿਤੀ ਤੇ ਅੰਸ਼ ਦਾ ਹਵਾਲਾ ਕਰਸਰ
Cursor iteration ਕਰਸਰ ਇੰਟਰੇਸ਼ਨ	ਇਕ ਸੰਗ੍ਰਹਿ ਵਿਚ ਕਰਸਰ ਨੂੰ ਅਗਲੇ ਅੰਸ਼ ਤਕ ਉਦੋਂ ਤਕ ਦੁਹਰਾਏ ਜਾਣ ਦੀ ਪ੍ਰਕ੍ਰਿਆ ਹੁੰਦੀ ਹੈ ਜਦੋਂ ਤਕ ਕਿਸੇ ਸ਼ਰਤ ਦੀ ਸੰਤੁਸ਼ਟੀ ਨਹੀਂ ਹੋ ਜਾਂਦੀ।
ਡੀ	
Declaration (ਡਿਕਲੇਰੇਸ਼ਨ)	C ਵਜੂਦ ਜਿਹੜੀ ਇਕ ਪ੍ਰੋਗਰਾਮ ਵਿਚ ਇਕ ਜਾਂ ਵਧੇਰੇ ਨਾਮਾਂ ਨੂੰ ਪ੍ਰਵੇਸ਼ ਕਰਦਾ ਹੈ।
Declaration statement (ਡਿਕਲੇਅਰ ਸਟੇਟਮੈਂਟ)	ਇਕ ਸਟੇਟਮੈਂਟ ਦੇ ਰੂਪ ਵਿਚ ਡਿਕਲੇਅਰ ਜਿਹੜੀ C ਵਿਚ ਵਰਤੀ ਜਾ ਸਕਦੀ ਜਿਥੇ ਆਮ ਤੌਰ ਤੇ ਸਟੇਟਮੈਂਟ ਵਰਤੇ ਜਾਣਗੇ
Default argument (ਡਿਫਾਲਟ ਦਲੀਲ)	ਇਕ ਫੰਕਸ਼ਨ ਲਈ ਵਿਕਲਪੀ ਦਲੀਲ। ਇਸ ਫੰਕਸ਼ਨ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿਚ ਨਿਸ਼ਚਿਤ ਮੁੱਲ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਬਸ਼ਰਤੇ ਦਲੀਲ ਨਾ ਦਿਤੀ ਗਈ ਹੋਵੇ।
Digital data (ਡਿਜੀਟਲ ਡਾਟਾ)	ਡਾਟਾ ਜਿਸਦੇ ਕੇਵਲ ਸੀਮਤ ਵਖਰੇ ਮੁੱਲ ਦੇ ਅੰਕ ਹੋਣ। ਦਿਨ ਦਾ ਸਮਾਂ ਡਿਜੀਟਲ ਕਲਾਕ ਜਾਂ ਤਾਪਮਾਨ ਜੋ ਕਿ ਡਿਜੀਟਲ ਥਰਮਾਮੀਟਰ ਰਾਹੀ ਦਰਸਾਇਆ ਜਾਂਦਾ ਹੈ ਡਿਜੀਟਲ ਡਾਟਾ ਦੀਆਂ ਮਿਸਾਲਾਂ ਹਨ। ਡਿਜੀਟਲ ਮੁੱਲ ਲਗਾਤਾਰ ਨਹੀ ਬਦਲਦਾ। ਪਰੰਤੂ ਇਕ ਮੁੱਲ ਤੇ ਹੀ ਰਹਿੰਦਾ ਹੈ ਤੇ ਫਿਰ ਦੂਜੇ ਵਿਚ ਬਦਲ ਜਾਂਦਾ ਹੈ।
Dump terminal (ਡੰਪ ਟਰਮਿਨਲ)	ਉਨ੍ਹਾਂ ਯੰਤਰਾਂ ਦਾ ਹਵਾਲਾ ਦਿੰਦਾ ਹੈ ਜਿਹੜੇ ਇਕ ਮੇਜ਼ਬਾਨ (ਮੁੱਖ ਫਰੇਮ) ਕੰਪਿਊਟਰ ਦੇ ਨਾਲ ਵਿਸ਼ੇਸ਼ ਤੌਰ ਤੇ ਸੰਚਾਰ ਕਰਨ ਲਈ ਤਿਆਰ ਕੀਤੇ ਗਏ ਹੁੰਦੇ ਹਨ। ਇਹ ਮੇਜ਼ਬਾਨ ਕੰਪਿਊਟਰ ਤੋਂ ਸਮੂਹ ਸਕਰੀਨ ਖਾਕੇ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ ਅਤੇ ਮੇਜ਼ਬਾਨ ਨੂੰ ਸਮੂਹ ਕੀ-ਬੋਰਡ ਪ੍ਰਵੇਸ਼ ਭੇਜਦਾ ਹੈ। ਇਕ ਮੇਜ਼ਬਾਨ ਕੰਪਿਊਟਰ ਤੋਂ ਬਿਨਾ ਫੰਕਸ਼ਨ ਹੀ ਨਹੀਂ ਕਰ ਸਕਦਾ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਈ	
Else (ਏਲੱਸ)	C ਕੀ-ਵਰਡ, if ਸਟੇਟਮੈਂਟ ਦਾ ਭਾਗ
enum (ਏਨਮ)	C ਕੀ-ਵਰਡ ਦੀ ਵਰਤੋ ਗਿਣਤੀ ਦੀ ਡਿਕਲੇਅਰ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
Expansion slot (ਵਿਸਥਾਰਮਈ ਸਲਾਟ)	ਕੰਪਿਊਟਰ ਵਿਚ ਇਕ ਖੇਤਰ ਜੋ ਕੰਪਿਊਟਰ ਦੀ ਯੋਗਤਾ ਵਿਸਥਾਰਮਈ ਸਲਾਟ ਵਿਚ ਵਾਧਾ ਕਰਨ ਲਈ ਵਧੀਕ ਇਨਪੁਟ/ਪਰਿਣਾਮ ਬੋਰਡਾਂ ਦੀ ਸਵੀਕ੍ਰਿਤੀ ਕਰਦਾ ਹੋਵੇ।
expression	ਐਕਸਪ੍ਰੇਸ਼ਨ ਕਿਸੇ ਪ੍ਰਕਾਰ ਦੇ ਮੁੱਲ ਪੈਦਾ ਕਰਨ ਲਈ ਕੌਂਸਟੈਂਟ ਵੇਰਿਏਬਲ ਅਤੇ ਸੰਚਾਲਕਾਂ ਦੇ ਸੁਮੇਲ।
Expression statement (ਐਕਸਪ੍ਰੇਸ਼ਨ ਸਟੇਟਮੈਂਟ)	ਇਕ ਸਟੇਟਮੈਂਟ ਜੋ ਐਕਸਪ੍ਰੇਸ਼ਨ ਹੁੰਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਫੰਕਸ਼ਨ ਮੰਗ ਜਾਂ ਐਸਾਈਨ
extern (ਐਕਸਟਰਨ)	ਇਕ C ਕੀ-ਵਰਡ ਜੋ ਬਾਹਰੀ ਨਾਮ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
ਐਫ	
False (ਗਲਤ)	ਬੂਲੀਅਨ ਟਾਈਪ ਲਈ ਮੁੱਲ ਨਿਸ਼ਚਤ ਕਰਨ ਵਾਲਾ C ਕੀ-ਵਰਡ ਜੋ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
Fiber Optic Cable (ਫਾਈਬਰ ਆਪਟਿਕ ਕੇਬਲ)	ਇਕ ਕੇਬਲ ਜੋ ਪਲਾਸਟਿਕ ਦੀਆਂ ਪਰਤਾਂ ਦੁਆਰਾ ਘਿਰੇ ਹੋਏ *centre glass core* ਵਿਚ ਸ਼ਾਮਲ ਹੋਵੇ ਜਿਹੜਾ ਬਿਜਲੀ ਦੀ ਬਜਾਏ ਰੌਸ਼ਨੀ ਦੀ ਵਰਤੋਂ ਨਾਂਲ ਡਾਟਾ ਸੰਚਾਰ ਕਰੇ। ਇਸ ਵਿਚ ਵਧੇਰੇ ਲੰਬੀ ਦੂਰੀ ਤੇ ਵਧੇਰੇ ਸੂਚਨਾ ਲਿਜਾਣ ਦੀ ਯੋਗਤਾ ਹੁੰਦੀ ਹੈ।
File Server (ਫਾਈਲ ਸਰਵਰ)	ਇਕ ਕੰਪਿਊਟਰ ਜਿਹੜਾ ਨੈੱਟਵਰਕ ਦੇ ਨਾਲ ਜੁੜਿਆ ਹੁੰਦਾ ਹੈ ਜਿਸ ਵਿਚ ਪ੍ਰਾਈਮਰੀ ਫਾਈਲਾਂ/ਪ੍ਰਯੋਗ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ ਅਤੇ ਇਹ ਨੈੱਟਵਰਕ ਤੇ ਹੋਰਾਂ ਕੰਪਿਊਟਰਾਂ ਦੇ ਨਾਲ ਬੇਨਤੀ ਕਰਨ ਤੇ ਹੀ ਜੋੜੇ ਜਾਂਦੇ ਹਨ। ਜੇਕਰ ਫਾਈਲ ਸਰਵਰ ਨੂੰ ਕੇਵਲ ਉਸੇ ਮੰਤਵ ਲਈ ਹੀ ਸਮਰਪਿਤ ਕੀਤਾ ਜਾਵੇ ਤਾਂ ਇਹ ਗ੍ਰਾਹਕ/ਸਰਵਰ ਨੈੱਟਵਰਕ ਦੇ ਨਾਲ ਜੁੜ ਜਾਂਦਾ ਹੈ।
Firewall (ਫਾਇਰਵਾਲ) -	ਇਕ ਨੈੱਟਵਰਕ ਤੋਂ / ਵੱਲ ਗੈਰ ਅਧਿਕਾਰਤ ਪਹੁੰਚ ਰੋਕਣ ਦੀ ਪ੍ਰਣਾਲੀ।
Function (ਫੰਕਸ਼ਨ)	ਇਕ C ਵਜੂਦ ਜੋ ਕਿ ਸਟੇਟਮੈਂਟਸ ਦਾ ਸਾਰ ਹੁੰਦਾ ਹੈ। ਇਸਦੀ ਆਪਣੀ ਹੀ ਗੁੰਜਾਇਸ਼ ਹੁੰਦੀ ਹੈ। ਦਲੀਲ ਮੁੱਲਾਂ ਦੇ ਸੈੱਟ ਨੂੰ ਸਵੀਕਾਰ ਕਰਦਾ ਹੈ ਅਤੇ ਮੁਕੰਮਲ ਕਰਨ ਤੇ ਮੁੱਲ ਮੋੜਦਾ ਹੈ।
ਜੀ	
Gigabyte (ਗੀਗਾਬਾਈਟਸ)	ਸੂਚਨਾ ਦਾ ਇਕ ਬਿਲੀਅਨ ਬਾਈਟਸ/ਇਕ ਹਜਾਰ ਮੇਗਾਬਾਈਟਸ।
global variable (ਗਲੋਬਲ ਵੇਰਿਏਬਲਸ)	ਇਕ ਵੇਰੀਏਬਲਸ ਜਿਹੜਾ ਸਮੁੱਚੇ ਪ੍ਰੋਗਰਾਮ ਦੇ ਵਿਚ ਪਹੁੰਚਯੋਗ ਹੁੰਦਾ ਹੈ ਜਿਸਦਾ ਜੀਵਨ ਕਾਲ ਪ੍ਰੋਗਰਾਮ ਜਿੰਨ੍ਹਾ ਹੀ ਹੁੰਦਾ ਹੈ।
goto	C ਕੀ-ਵਰਡ ਜੋ C ਫੰਕਸ਼ਨ ਦੇ ਵਿਚ ਕੰਟਰੋਲ ਬਦਲਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਐਚ	
header file (ਹੈਡਰ ਫਾਈਲ)	ਇਕ ਫਾਈਲ ਜਿਸ ਵਿਚ ਸ੍ਰੇਣੀ ਡਿਕਲੇਅਰਸ, ਪ੍ਰੀਪ੍ਰੋਸੈਸਰ, ਡਾਇਰੈਕਟਿਵ ਆਦਿ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ ਅਤੇ ਰੂਪਾਂਤਰ ਇਕਾਈ ਵਿਚ ਸ਼ਾਮਲ ਹੁੰਦੇ ਹਨ।ਇਨ੍ਹਾਂ ਦਾ ਵਿਸਥਾਰ ਪ੍ਰੀਪ੍ਰੋਸੈਸਰ ਦੁਆਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।
host (ਮੇਜ਼ਬਾਨ)	ਇਕ ਨੈੱਟਵਰਕ ਤੇ ਮਸ਼ੀਨਾਂ ਜੋ ਯੂਜ਼ਰ ਪ੍ਰੋਗਰਾਮ ਚਲਾਉਂਦੀਆਂ ਹਨ।
Hub (ਹੱਬ)	ਇਕ ਹਾਰਡਵੇਅਰ ਡਿਵਾਈਸ ਜਿਸ ਵਿਚ ਬਹੁਪੱਖੀ ਸੁਤੰਤਰ ਪਰੰਤੂ ਇਕ ਨੈੱਟਵਰਕ ਨਾਲ ਜਿਸ ਵਿਚ ਮਾਡਯੂਲ ਅਤੇ ਅੰਦਰੂਨੀ ਫੰਕਸ਼ਨ ਉਪਕਰਣ ਜੁੜੇ ਹੁੰਦੇ ਹਨ। ਹੱਬ ਸਰਗਰਮ ਹੋ ਸਕਦੇ ਹਨ (ਜਿਥੇ ਉਹ ਉਨ੍ਹਾਂ ਰਾਹੀਂ ਭੇਜੇ ਗਏ ਸਿਗਨਲਾਂ ਨੂੰ ਦੁਹਰਾਉਂਦੇ ਹਨ) ਜਾਂ ਗੈਰ-ਸਰਗਰਮ (ਜਿਥੇ ਉਹ ਦੁਹਰਾਉਂਦੇ ਨਹੀਂ ਬਲਕਿ ਆਪਣੇ ਰਾਹੀਂ ਭੇਜੇ ਗਏ ਸਿਗਨਲਾਂ ਨੂੰ ਨਿਖੇੜਦੇ ਹਨ।
ਆਈ	
Infrared (ਇਨਫਰਾਰੈਡ)	ਇਲੈਕਟਰੋਮੈਗਨੇਟਿਕ ਤਰੰਗਾਂ ਜਿਨ੍ਹਾਂ ਦੀ ਰਫ਼ਤਾਰ ਮਾਈਕਰੋਵੇਵ ਤੋਂ ਵੀ ਉਪਰ ਹੁੰਦੀ ਹੈ ਪਰੰਤੂ ਦ੍ਰਿਸਟਗੋਚਰ ਸਪੈਕਟਰਮ ਤੋਂ ਘਟ ਹੁੰਦੀ ਹੈ।
initialization (ਆਰੰਭਕਰਨ)	ਇਕ ਵੇਰੀਏਬਲਜ਼ ਜਾਂ ਕਾਨਸਟੈਂਟ ਨੂੰ ਮੁਢਲਾ ਮੁੱਲ ਦੇਣ ਲਈ।
Initialize (ਆਰੰਭ ਕਰਨ)	ਆਰੰਭ-ਕਰਨ ਦੀ ਪ੍ਰਕ੍ਰਿਆ
Initializer (ਆਰੰਭ ਕਰਤਾ)	ਆਰੰਭ-ਕਰਨ ਸਮੇਂ ਇਕ ਆਬਜੈਕਟ ਨੂੰ ਆਰੰਭ-ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਐਕਸਪ੍ਰੇਸ਼ਨ ਜਾਂ ਮੁੱਲ
int (ਇੰਟ)	C ਕੀ-ਵਰਡ ਅਤੇ ਮੌਲਿਕ ਟਾਈਪ ਜੋ ਪੂਰਣਅੰਕ ਟਾਈਪ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

Integral conversion (ਪੂਰਨ ਅੰਕ ਵੇਰੀਏਬਲਸ)	ਸਿਸਟਮ ਜਿਸ ਰਾਹੀ ਪੂਰਣ ਅੰਕ ਨੂੰ ਹਸਤਾਖਰ ਜਾਂ ਗੈਰ ਹਸਤਾਖਰ ਵਿਚ ਵੇਰੀਏਬਲਜ਼ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।
internet (ਇੰਟਰਨੈੱਟ)	ਨੈੱਟਵਰਕ ਜਿਹੜਾ ਇਕ ਸੰਸਥਾ ਜੋ ਇੰਟਰਨੈੱਟ ਪ੍ਰੋਟੋਕੋਲਾਂ ਦੀ ਵਰਤੋਂ ਕਰਦੀ ਹੋਵੇ ਦਾ ਅੰਦਰੂਨੀ ਹੋਵੇ।
ਕੇ	
keyword (ਕੀ-ਵਰਡ)	C ਵਿਚ ਰਾਖਵਾਂ ਸ਼ਨਾਖਤ ਕਰਤਾ, ਜੋ ਡਾਟਾ-ਟਾਈਪ, ਭਾਸ਼ਾ ਦੇ ਸਟੇਟਮੈਂਟ ਆਦਿ ਨੂੰ ਸੂਚਿਤ ਕਰਦਾ ਹੈ
ਐਲ	
LAN (ਲੈਨ)	ਇਕ ਨੈੱਟਵਰਕ ਜੋ ਇਕ ਛੋਟੇ ਜਿਹੇ ਸਾਪੇਖਿਕ ਖੇਤਰ ਜਿਵੇਂ ਬਿਲਡਿੰਗ ਵਿਚ ਕੰਪਿਊਟਰਾਂ ਦੇ ਨਾਲ ਜੁੜਿਆ ਹੁੰਦਾ ਹੈ।
Library (ਲਾਇਬ੍ਰੇਰੀ)	ਫਾਈਲਾਂ ਦਾ ਸੈਟ ਜੋ ਇਕੱਠਾ ਹੁੰਦਾ ਹੈ। ਇਕ ਲਿੰਕਰ ਵਲੋਂ ਉਨ੍ਹਾਂ ਨੂੰ ਮੁੜ-ਮੁੜ ਖੋਜਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਜਿਹੜੀਆਂ ਲਕਸ਼ ਫਾਈਲਾਂ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ।
literal (ਸ਼ਾਬਦਿਕ)	1234 ਦੀ ਤਰ੍ਹਾਂ ਸਥਾਈ।
Local variable (ਸਥਾਨਕ ਵੇਰੀਏਬਲਸ)	ਇਕ ਵੇਰੀਏਬਲਜ਼ ਜੋ ਇਕ ਫੰਕਸ਼ਨ ਦੁਆਰਾ ਸਥਾਨਕ ਡਿਕਲੇਅਰ ਕੀਤਾ ਜਾਂਦਾ ਹੈ
Long (ਲੰਬਾ)	C ਕੀ-ਬੋਰਡ ਜੋ ਇਕ ਲੰਬੇ ਪੂਰਣ ਅੰਕ ਡਾਟਾ ਕਿਸਮ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ
Long double ਲੰਬਾ ਦੁਗਣਾ	C ਵਿਚ ਅਸਥਾਈ ਨੁਕਤਾ ਕਿਸਮ

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਐਮ	
MAN (ਮੈਨ)	(ਮੈਟਰੋਪੋਲੀਟਨ ਏਰੀਆ ਨੈੱਟਵਰਕ) ਇਕ ਨੈੱਟਵਰਕ ਜੋ ਕੰਪਿਊਟਰਾਂ ਨੂੰ ਵੱਡੇ ਭੂਗੋਲਿਕ ਖੇਤਰ ਜਿਵੇਂ ਇਕ ਸ਼ਹਿਰ ਜਾਂ ਸਕੂਲ ਜ਼ਿਲ੍ਹੇ ਨਾਲ ਜੋੜਦਾ ਹੈ।
modem (ਮਾਡਮ)	ਮਾਡਯੁਲੇਟਰ/ਡਿਮਾਡਯੁਲੇਟਰ/ਡਿਵਾਈਸ ਜਿਹੜੇ ਡਿਜ਼ੀਟਲ ਅਤੇ ਐਨਾਲਾਗ ਸਿਗਨਲਾਂ ਨੂੰ ਬਦਲਦੇ ਹਨ। ਮਾਡਮ ਕੰਪਿਊਟਰ ਡਾਪ (ਡਿਜ਼ੀਟਲ ਨੂੰ ਅਵਾਜ਼ ਗਰੇਡ ਟੈਲੀਫ਼ੋਨ ਫਾਈਲਾਂ ਟੇਨਾਲੋਗ ਵਿਚ ਸੰਚਾਰ ਕਰਨ ਲਈ ਹੁੰਦੇ ਹਨ।
Multiplexer (ਮਲਟੀਪਲੈਕਸਰ)	ਇਕ ਡਿਵਾਈਸ ਜਿਹੜਾ ਇਕ ਇਕਹਰੇ ਭੌਤਿਕ ਚੈਨਲ ਨੂੰ ਨਾਲ-ਨਾਲ ਸੰਚਾਰ ਕਰਨ ਲਈ ਬਹੁ-ਪੱਖੀ ਲੌਜ਼ੀਕਲ ਸਿਗਨਲਾਂ ਲਈ ਆਗਿਆ ਦਿੰਦਾ ਹੈ।
ਐਨ	
Network Interface Card (ਨੈੱਟਵਰਕ ਇੰਟਰਫੇਸ ਕਾਰਡ)	ਇਕ ਬੋਰਡ ਜਿਹੜਾ ਕਿ ਇਕ ਕੰਪਿਊਟਰ ਤੋਂ ਨੈੱਟਵਰਕ ਸੰਚਾਰ ਯੋਗਤਾਵਾਂ ਪਰਵਾਨ ਕਰਦਾ ਹੈ।
Network Modem (ਨੈੱਟਵਰਕ ਮਾਡਮ)	ਇਕ ਮਾਡਮ ਜੋ ਕਿ ਇਕ ਸਥਾਨਕ ਖੇਤਰੀ ਲੋਕਲ ਏਰੀਆ ਨੈੱਟਵਰਕ (ਲੈਨ) ਜੋ ਕਿ ਨੈੱਟਵਰਕ ਤੇ ਕਿਸੇ ਵੀ ਨੈੱਟਵਰਕ ਸਟੇਸ਼ਨ ਤੋਂ ਪਹੁੰਚਣ ਯੋਗ ਹੁੰਦਾ ਹੈ।
Network Operating System (ਨੈੱਟਵਰਕ ਔਪਰੇਟਿੰਗ ਸਿਸਟਮ)	ਨੈੱਟਵਰਕ ਔਪਰੇਟਿੰਗ ਸਿਸਟਮ ਇਕ ਤੋਂ ਵਧੇਰੇ ਕੰਪਿਊਟਰਾਂ ਦੇ ਦਰਮਿਆਨ ਸੰਚਾਰ ਅਤੇ ਸੂਚਨਾ ਭੇਜਣ ਲਈ ਤਿਆਰ ਕੀਤਾ ਗਿਆ ਹੈ। ਮਿਸਾਲਾਂ ਲਈ Windows NT ਸਰਵਰ ਸ਼ਾਮਲ ਹਨ।
Node (ਨੋਡ)	ਇਕ ਨੈੱਟਵਰਕ ਕਾਨਸਟੈਂਟ ਦਾ ਅੰਤਿਮ ਬਿੰਦੂ। ਨੋਡਜ਼ ਵਿਚ ਇਕ ਨੈੱਟਵਰਕ ਦੇ ਨਾਲ ਜੁੜਿਆ ਹੋਇਆ ਕੋਈ ਵੀ ਡਿਵਾਈਸ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਫਾਈਲ ਸਰਵਰ, ਪ੍ਰਿੰਟਰ ਜਾਂ ਵਰਕ-ਸਟੇਸ਼ਨ।
Null (ਟਲ)	ਇਕ ਵਿਸ਼ੇਸ਼ ਕਾਨਸਟੈਂਟ ਮੁੱਲ
ਓ	
Object (ਆਬਜੈਕਟ)	ਇਸ ਦੇ ਅਨੇਕਾਂ ਅਰਥ ਹਨ। C++, ਵਿਚ ਇਕ ਸ਼੍ਰੇਣੀ ਦੀ ਮਿਸਾਲ ਦਾ ਅਕਸਰ ਹਵਾਲਾ ਦਿਤਾ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਹ ਇਕ ਵੇਰੀਏਬਲਸ ਜਾਂ ਹੋਰ ਵਜੂਦ ਜਿਸ ਵਿਚ ਭੰਡਾਰਣ ਸ਼ਾਮਲ ਹੁੰਦਾ ਹੈ
ਪੀ	
parameter (ਪੈਰਾਮੀਟਰ)	ਇਹ ਕਾਲਿੰਗ ਕੋਡ ਨੂੰ ਬੁਲਾਏ ਜਾਣ ਤੇ ਵਿਧੀ ਦੇ ਮੁੱਲ ਦੀ ਆਗਿਆ ਦਿੰਦਾ ਹੈ।
Peer-to-Peer Network (ਪੀਅਰ-ਟੂ-ਪੀਅਰ ਨੈੱਟਵਰਕ)	ਇਕ ਨੈੱਟਵਰਕ ਜਿਸ ਵਿਚ ਇਕ ਕੇਂਦਰੀ ਪ੍ਰਬੰਧ ਵਸੀਲੇ ਤੋਂ ਬਿਨਾਂ ਸਾਧਨ ਅਤੇ ਫਾਈਲਾਂ ਦੀ ਵੰਡ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
Physical Topology (ਭੌਤਿਕ ਟੋਪੋਲੋਜੀ)	ਨੈੱਟਵਰਕ ਦਾ ਭੌਤਿਕ ਖ਼ਾਕਾ ਕਿ ਕੇਬਲਸ ਦਾ ਪ੍ਰਬੰਧ ਕਿਵੇਂ ਕਰਨਾ ਹੈ ਅਤੇ ਕੰਪਿਊਟਰ ਕਿਵੇਂ ਜੋੜੇ ਜਾਂਦੇ ਹਨ।
Point-to-Point (ਪੋਆਂਇੰਟ ਟੂ ਪੁਆਂਇੰਟ)	ਇਕ ਨੈੱਟਵਰਕ ਵਿਚ ਦੋ ਆਬਜੇਕਟਸ ਦੇ ਦਰਮਿਆਨ ਸਿੱਧਾ ਸੰਪਰਕ।
Ports (ਪੋਰਟ)	ਇਕ ਕੇਬਲ ਲਈ ਕੁਨੈਕਸ਼ਨ ਪੁਆਂਇੰਟ
programming environment (ਪ੍ਰੋਗਰਾਮਿੰਗ ਵਾਤਾਵਰਣ)	ਵਿਕਸਿਤ ਹੋ ਰਹੇ ਸਾਫ਼ਟ-ਵੇਅਰ ਵਿਚ ਜਿਸ. ਵਿਚ ਇਕ ਕੰਪਾਇਲਰ, ਲਿੰਕਰ, ਡੀ-ਬਗਰ ਅਤੇ ਬ੍ਰਾਊਜ਼ਰ ਸ਼ਾਮਲ ਹਨ, ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਸੰਗਠਿਤ ਉਪਕਰਣਾਂ ਦਾ ਸੈਟ।
Protocol (ਪ੍ਰੋਟੋਕੋਲ)	ਨਿਯਮਾਂ ਅਤੇ ਪਰੰਪਰਾਵਾਂ ਦੇ ਇਕ ਸੈਟ ਦਾ ਰਸਮੀ ਸਟੇਟਮੇਂਟ ਜਿਹੜਾ ਇਸ ਗੱਲ ਦਾ ਪ੍ਰਬੰਧ ਕਰਦਾ ਹੋਵੇ ਕਿ ਇਕ ਨੈੱਟਵਰਕ ਉਤੇ ਡਿਵਾਂਈਸ ਕਿਵੇਂ ਸੂਚਨਾਵਾਂ ਅਦਾਨ-ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਆਰ	
Reference (ਹਵਾਲਾ)	ਇਕ ਆਬਜੈਕਟ ਲਈ ਹੋਰ ਨਾਮ/ਇਕ ਆਬਜੈਕਟ ਵਲ ਪਹੁੰਚ ਜੋ ਕਿ ਹਵਾਲੇ ਦੀ ਵਰਤੋਂ ਕਰਦੀ ਹੈ। ਹਵਾਲਿਆਂ ਨੂੰ ਵਿਸ਼ੇਸ਼ ਕਰਕੇ ਪੁਆਂਇੰਟਰਾਂ ਵਜੋਂ ਲਾਗੂ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।
Register (ਰਜਿਸਟਰ)	C ਕੀ-ਵਰਡ ਦੀ ਵਰਤੋਂ ਕੰਪਾਇਲਰ ਨੂੰ ਇਸ਼ਾਰੇ ਵਜੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ ਇਕ ਵਿਸ਼ੇਸ਼ ਸਥਾਨਕ ਵੇਰੀਏਬਲ ਇਕ ਮਸ਼ੀਨ ਰਜਿਸਟਰ ਵਿਚ ਰਖਿਆ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ।
Return (ਵਾਪਸੀ)	C ਕੀ-ਵਰਡ ਦੀ ਵਰਤੋਂ ਇਕ ਫੰਕਸ਼ਨ ਤੋਂ ਵਾਪਸ ਮੋੜਣ ਵਾਲੇ ਮੁੱਲ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
Return value (ਵਾਪਸੀ ਮੁੱਲ)	ਇਕ ਫੰਕਸ਼ਨ ਤੋਂ ਵਾਪਸ ਆਇਆ ਮੁੱਲ।
ਐਸ	
Short (ਛੋਟਾ)	ਇਕ C ਬੁਨਿਆਦੀ ਕਿਸਮ ਜਿਹੜੀ ਛੋਟੇ ਪੂਰਨ ਅੰਕਾਂ ਦਾ ਐਲਾਨ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Signed	C ਕੀ-ਵਰਡ ਜੋ ਕਿ ਹਸਤਾਖਰ ਡਾਟਾ-ਟਾਈਮ ਦੇ ਪ੍ਰਤੀਕ ਵਜੋਂ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Sizeof (ਦਾਆਕੌਰ)	C ਕੀ-ਵਰਡ ਜੋ ਕਿ ਇਕ ਔਬਜੈਕਟ ਜਾਂ ਟਾਈਪ ਦਾ ਆਕਾਰ ਲੈਣ ਲਈ ਹੁੰਦਾ ਹੈ।
Speed of Data Transfer (ਡਾਟਾ ਬਦਲੀ ਦੀ ਗਤੀ)	ਉਹ ਦਰ ਜਿਸ ਤੇ ਇਕ ਨੈੱਟਵਰਕ ਰਾਹੀਂ ਸੂਚਨਾ ਯਾਤਰਾ ਕਰਦੀ ਹੈ ਜੋ ਕਿ ਆਮ ਤੌਰ ਤੇ ਪ੍ਰਤੀ ਸੈਕਿੰਡ ਮੈਗਾ-ਬਿੱਟ ਵਿਚ ਮਾਪੀ ਜਾਂਦੀ ਹੈ।
Star-Wired Ring (ਸਟਾਰ-ਵਾਈਰਡ ਰਿੰਗ)	ਨੈੱਟਵਰਕ ਟੋਪਾਲੋਜੀ ਜਿਹੜੀ ਨੈੱਟਵਰਕ ਯੰਤਰਾਂ ਨੂੰ ਜੋੜਦੀ ਹੈ। (ਜਿਵੇਂ ਕਿ ਸੰਪੂਰਨ ਸਰਕਲ ਵਿਚ ਕੰਪਿਊਟਰ ਅਤੇ ਪ੍ਰਿੰਟਰ।)
Statement (ਸਟੇਟਮੈਂਟ)	ਇਕ ਪ੍ਰੋਗਰਾਮ ਦੇ ਭਾਗ ਜਿਹੜੇ ਅਸਲ ਵਿਚ ਕੰਮ ਕਰਦੇ ਹਨ।
Static Member (ਸਟੈਟਿਕ ਮੈਂਬਰ)	ਇਕ ਸ਼੍ਰੇਣੀ ਜਿਹੜੇ ਕਿ ਪਹੁੰਚ ਕੰਟਰੋਲ ਦੇ ਮਕਸਦ ਲਈ ਇਕ ਸ਼੍ਰੇਣੀ ਦਾ ਭਾਗ ਹੁੰਦੇ ਹਨ ਪ੍ਰੰਤੂ ਸ਼੍ਰੇਣੀ ਦੇ ਕਿਸੇ ਵਿਸ਼ੇਸ਼ ਆਬਜੈਕਟ ਮਿਸਾਲਾਂ ਤੇ ਕੰਮ ਨਹੀਂ ਕਰਦੇ।
Static Object (ਸਟੈਟਿਕ ਆਬਜੈਕਟ)	ਇਕ ਆਬਜੈਕਟ ਜਿਹੜਾ ਕਿ ਫੰਕਸ਼ਨ ਲਈ ਸਥਾਨਕ ਹੁੰਦਾ ਹੈ ਜਾਂ ਰੁਪਾਂਤਰ ਇਕਾਈ ਲਈ ਅਤੇ ਜਿਸ ਦਾ ਜੀਵਨਕਾਲ ਇਕ ਪ੍ਰੋਗਰਾਮ ਦੇ ਜੀਵਨ ਬਰਾਬਰ ਹੁੰਦਾ ਹੈ।
Switch (ਸਵਿਚ)	ਕੀ-ਵੱਰਡ ਇਕ ਸਟੇਟਮੈਂਟ ਕਿਸਮ ਦੀ ਟਿੱਪਣੀ ਕਰਦਾ ਹੈ ਜੋ ਕਿ ਮੁੱਲ ਉਤੇ ਆਧਾਰਤ ਸਟੇਟਮੈਂਟਸ ਦੇ ਅਨੇਕਾਂ ਭਾਗਾਂ ਚੋਂ ਇਕ ਨੂੰ ਡਿਸਪੈਚ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
Syntax (ਸਟਰਿੰਗ ਬਣਤਰ)	ਸਟਰਿੰਗ-ਬਣਤਰ ਨਿਯਮ ਜਿਹੜੇ ਕਿ C ਵਿਚ ਪ੍ਰਬੰਧ ਕਰਦੇ ਹਨ ਕਿ ਐਕਸਪ੍ਰੇਸ਼ਨ, ਸਟੇਟਮੈਂਟ, ਡਿਕਲੇਅਰੇਸ਼ਨ ਅਤੇ ਪ੍ਰੋਗਰਾਮ ਕਿਵੇਂ ਬਣਾਉਣੇ ਹਨ।

Downloaded from https:// www.studiestoday.com

Downloaded from https:// www.studiestoday.com

ਟੀ	
Tag (ਟੈਗ)	ਇਕ ਸ਼੍ਰੇਣੀ, ਨਿਰਮਾਣ ਜਾਂ ਏਕਤਾ ਦਿੱਤਾ ਗਿਆ ਨਾਮ।
Terminator (ਟਰਮੀਨੇਟਰ)	ਇਕ ਡਿਵਾਈਸ ਜਿਹੜਾ ਇਕ ਸੰਚਾਰਨ ਲਾਈਨ ਦੇ ਅੰਤ ਤੇ ਬਿਜਲੀ ਪ੍ਰਤੀਰੋਧ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਸ ਦਾ ਕੰਮ ਲਾਈਨ ਤੇ ਸਿਗਨਲਾਂ ਨੂੰ ਗ੍ਰਹਿਣ (absorb) ਕਰਨਾ ਹੁੰਦਾ ਹੈ ਅਤੇ ਇਸ ਪ੍ਰਕਾਰ ਇਹ ਉਨ੍ਹਾਂ ਨੂੰ ਉਛਲਣ ਤੋਂ ਰੋਕਦੇ ਹਨ ਅਤੇ ਮੁੜ ਨੈੱਟਵਰਕ ਦੁਆਰਾ ਪ੍ਰਾਪਤ ਕੀਤੇ ਜਾਂਦੇ ਹਨ।
Token (ਟੋਕਨ)	ਇਕ ਵਿਸ਼ੇਸ਼ ਪੈਕਿਟ ਜਿਸ ਵਿਚ ਡਾਟਾ ਭਰਿਆ ਹੁੰਦਾ ਹੈ ਅਤੇ ਜੋ ਇਕ ਰਿੰਗ-ਟਾਪੋਲੋਜੀ ਤੇ ਹਰੇਕ ਕੰਪਿਊਟਰ ਅਤੇ ਡਿਵਾਈਸ ਦੇ ਦਰਮਿਆਨ ਇਕ ਸੰਦੇਸ਼ਵਾਹਕ ਜਾਂ ਨਿਸ਼ਾਣ ਵਾਲੇ ਵਜੋਂ ਕੇਸ ਕਰਦਾ ਹੈ। ਹਰੇਕ ਕੰਪਿਊਟਰ ਨੂੰ ਨੈੱਟਵਰਕ ਤੇ ਡਾਟਾ ਭੇਜਣ ਤੋਂ ਪਹਿਲਾਂ ਆਪਣੀ ਨੋਡ ਤੇ ਸੰਦੇਸ਼ਵਾਹਕ ਲਈ ਇੰਤਜ਼ਾਰ ਕਰਨੀ ਚਾਹੀਦੀ ਹੈ।
Token ring (ਟੋਕਨ ਰਿੰਗ)	IBM ਦੁਆਰਾ ਵਿਕਸਿਤ ਕੀਤਾ ਗਿਆ ਨੈੱਟਵਰਕ ਪ੍ਰੋਟੋਕੋਲ ਜਿਸ ਵਿਚ ਕੰਪਿਊਟਰ ਟੋਕਨ ਪਾਸਿੰਗ ਰਾਹੀ ਨੈੱਟਵਰਕ ਕੋਲ ਜਾਂਦੇ ਹਨ। ਅਕਸਰ ਸਟਾਰ ਵਾਇਰਡ ਰਿੰਗ ਟਾਪੋਲੋਜੀ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
Topology (ਟਾਪੋਲੋਜੀ)	ਇਸ ਦੀਆਂ ਦੋ ਕਿਸਮਾਂ ਹੁੰਦੀਆਂ ਹਨ, ਭੌਤਿਕ ਅਤੇ ਲੌਜਿਕਲ। ਭੌਤਿਕ ਟੋਪੋਲੋਜੀ ਕੇਬਲਾਂ, ਕੰਪਿਊਟਰਾਂ ਅਤੇ ਹੋਰ ਸੀਮਾਵਾਂ ਦੇ ਆਕਾਰ ਦਾ ਹਵਾਲਾ ਦਿੰਦੀਆ ਹਨ। ਲੌਜਿਕਲ ਟਾਪੋਲੋਜੀ ਅਜਿਹੀ ਵਿਧੀ ਹੈ ਜੋ ਵਰਕ ਸਟੇਸ਼ਨਾਂ ਦੇ ਦਰਮਿਆਨ ਸੂਚਨਾ ਭੇਜਦੀ ਹੈ।
Transceiver (ਟਰਾਂਸਮੀਵਰ)	ਟਰਾਂਸਮੀਟਰ/ਰਿਸੀਵਰ ਇਕ ਡਿਵਾਈਸ ਜਿਹੜਾ ਇਕ ਮਾਧਿਅਮ ਰਾਹੀ ਸਿਗਨਲ ਭੇਜਦਾ ਅਤੇ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ। ਨੈੱਟਵਰਕ ਵਿਚ ਇਸ ਨੂੰ ਆਮ ਤੌਰ ਤੇ ਦੋ ਵੱਖ-ਵੱਖ ਟਾਈਪ ਦੇ ਕੇਬਲ ਕੁਨੈਕਟਰਾਂ ਜਿਵੇਂ ਕਿ AUI ਅਤੇ RJ 45 ਦੇ ਦਰਮਿਆਨ ਕੁਨੈਕਸ਼ਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ।
Twisted Pair (ਟਵਿਸਟਿਡ ਪੇਅਰ)	ਨੈੱਟਵਰਕ ਕੇਬਲ ਜਿਹੜਾ ਕੇਬਲਜ਼ ਦੇ ਚਾਰ ਜੋੜਿਆਂ ਨਾਲ ਬਣਿਆ ਹੁੰਦਾ ਹੈ ਜਿਹੜੇ ਕੁਝ ਵਸ਼ਿਸ਼ਠਤਾਵਾਂ ਦੇ ਨਾਲ ਮੋੜੀਆਂ ਹੋਈਆਂ ਕੇਬਲਜ਼ ਨਾਲ ਬਣਾਏ ਗਏ ਹੁੰਦੇ ਹਨ। ਇਹ ਸ਼ੀਲਡਿਡ ਅਤੇ ਗੈਰ ਸ਼ੀਲਡਿਡ ਰੂਪਾਂਤਰਾਂ ਵਿਚ ਹੁੰਦੇ ਹਨ।
ਯੂ	
Unsigned (ਅਨਸਾਈਨਡ)	C ਕੀ-ਵਰਡ ਵਲੋਂ ਪੂਰਣ ਅੰਕ ਗੈਰ ਅਧਿਕਾਰਤ ਬੁਨਿਆਦੀ ਟਾਈਪ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਦਾ ਹੈ।
USB (ਯੂ ਐਸ ਬੀ) ਯੂਨੀਵਰਸਲ ਸੀਰੀਅਲ ਬੱਸ	ਪੋਰਟ ਇਕ ਹਾਰਡਵੇਅਰ ਜਿਹੜਾ ਘਟ ਗਤੀ ਦੀਆਂ ਸੀਮਾਵਾਂ ਲਈ ਇੰਟਰਫੇਸ ਕਰਦਾ ਹੈ। ਜਿਵੇਂ ਕਿ ਕੀ-ਵਰਡ, ਮਾਊਸ, ਸਕੈਨਰ, ਪ੍ਰਿੰਟਰ ਅਤੇ ਟੈਲੀਫੋਨੀ ਡਿਵਾਈਸ ਆਦਿ।
ਡਬਲਯੂ	
WAN (ਵੈਨ) (ਵਾਈਡ ਏਰਿਆ ਨੈੱਟਵਰਕ)	ਇਕ ਨੈੱਟਵਰਕ ਜਿਹੜਾ ਕੰਪਿਊਟਰਾਂ ਨੂੰ ਬਹੁਤ ਵੱਡੇ ਖੇਤਰਾਂ ਜਿਵੇਂ ਰਾਜਾਂ, ਦੇਸ਼ਾਂ ਅਤੇ ਵਿਸ਼ਵ ਨਾਲ ਜੋੜਦਾ ਹੈ।
While (ਵਾਈਲ)	C ਭਾਸ਼ਾ ਵਿਚ ਇਕ ਲੂਪ ਜੋ ਸਟੇਟਮੈਂਟਸ ਦੀ ਪਰਿਸਥਿਤੀ ਨੂੰ ਲਾਗੂ ਕਰਨ ਲਈ ਬਾਰ-ਬਾਰ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਦ ਤਕ ਨਿਸ਼ਚਤ ਹਾਲਾਤ *(condition)* ਸਹੀ ਹੈ
Workgroup (ਵਰਕਗਰੁੱਪ)	LAN ਤੇ ਵਰਕਸਟੇਸ਼ਨ ਅਤੇ ਸਰਵਰ ਜੋ ਇਕ ਦੂਜੇ ਨਾਲ ਡਾਟਾ ਦਾ ਵਿਚਾਰ ਵਟਾਂਦਰਾ ਅਤੇ ਸੰਚਾਰ ਕਰਨ ਲਈ ਬਣਾਇਆ ਗਿਆ ਹੈ।
Workstation (ਵਰਕਸਟੇਸ਼ਨ)	ਵਰਕਸਟੇਸ਼ਨ ਨੈੱਟਵਰਕ ਦੇ ਨਾਲ ਜੁੜਿਆ ਹੋਇਆਂ ਇਕ ਕੰਪਿਊਟਰ ਜਿਸ ਤੇ ਯੂਜ਼ਰ ਨੈੱਟਵਰਕ ਤੇ ਸਟੋਰ ਕੀਤੇ ਗਏ ਸਾਫਟਵੇਅਰ ਦੇ ਨਾਲ ਪਰਸਪਰ ਪ੍ਰਭਾਵ ਲੈਂਦਾ ਹੈ।

Downloaded from https:// www.studiestoday.com