باب نمبر 2

"C" پروگرامنگ

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- پروگرام کی ترکیب (Syntax) اور مفہوم (Semantic) واضح کر سکیں۔
- کمپیوٹر کی زبانوں کی مختلف اقسام کو تفصیل سے بیان کر سکیں جیسے ہائی لیول لینگوئج اور لو لیول لینگوئج۔
- ہائی لیول لینگوئج کی خصوصیات کو تفصیلاً بیان کر سکیں۔
- مختلف طرح کی ہائی لیول لینگوئج کو جان سکیں۔
- کمپائلر اور انٹرپریٹر کا فرق واضح کر سکیں۔
- انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) کے تصور کو بیان کر سکیں۔
- C لینگوئج کے مختلف اجزاء (ایڈیٹر، کمپائلر، لنکر، لوڈر اور ڈی بگر) کو تفصیلاً بیان کر سکیں۔
- ہیڈر فائلز کی تعریف کر سکیں۔
- ریزروڈ ورڈز (Reserved Words) کی پہچان کر سکیں۔
- C پروگرام کے بنیادی ڈھانچے (Structure) کو بیان کر سکیں۔
- Comments کے گرائمر اور مقصد کو سمجھ سکیں۔
- متغیر (Variable) اور مستقل (Constant) میں فرق کر سکیں۔
- Variable کے نام سے متعلق اصولوں کو تفصیلاً بیان کر سکیں۔
- C پروگرام میں استعمال ہونے والی ڈیٹا ٹائپس (Data Types) کو جان سکیں۔
- ٹائپ کاسٹنگ (Typecasting) کا استعمال کر سکیں۔
- *const* کی تعریف کر سکیں۔
- Variable اور Constant بنانے اور ابتدائی مقدار مقرر کرنے کے عمل کو تفصیلاً بیان کر سکیں۔

NOT FOR SALE

2.1 پروگرامنگ کا ایک تعارف (Introduction to Programming)

کمپیوٹر ایک ایسا آلہ ہے جو دی گئی ہدایات کے مطابق عمل کرتا ہے۔ واضح ہدایات کا ایسا مجموعہ جو کمپیوٹر کو فراہم کیا جائے کمپیوٹر پروگرام کہلاتا ہے۔ کمپیوٹر پروگرامنگ ایک ایسا عمل ہے جس میں ہم کسی بھی مسئلے کو حل کرنے کے لیے ایک کمپیوٹر پروگرام بناتے ہیں۔ دنیا میں کمپیوٹر کے لیے بہت ساری زبانیں استعمال ہوتی ہیں جیسے C ، C C++ اور Java وغیرہ۔ ان میں سے ہر ایک کی اپنی کچھ نہ کچھ خصوصیات ہیں۔ کمپیوٹر پروگرام کمپیوٹر میں ہونے والے مختلف عوامل کو کنٹرول کرتا ہے اور یہ کسی نہ کسی پروگرامنگ لینگویج میں لکھا جاتا ہے تاکہ کوئی مخصوص کام سرانجام دیا جاسکے۔

2.1.1 کمپیوٹر پروگرام (Computer Program)

کمپیوٹر پروگرام ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتا سکے کہ کیا کرنا اور کیسے کرنا ہے۔ کوئی بھی کمپیوٹر، کمپیوٹر پروگرام میں دی گئی ہدایات کے مطابق عمل کرتا ہے۔ کمپیوٹر پروگرام کسی بھی پروگرامنگ لینگویج میں لکھے جاتے ہیں۔ جو فرد کمپیوٹر پروگرام بناتا ہے اسے پروگرامر کہا جاتا ہے۔ پروگرامر پروگرام بنا کر کمپیوٹر کو ہدایات دیتا ہے کہ کس طرح ڈیٹا کو پروسیس کرے تاکہ وہ انفارمیشن میں تبدیل ہو جائے۔ پروگرامر مختلف پروگرامنگ لینگویج یا پروگرامنگ آلات کو استعمال کرکے پروگرام بناتا ہے۔ ذیل میں دی گئی دو اصطلاحات کمپیوٹر پروگرام سے متعلق ہیں۔

پروگرام کی ترکیب (Program Syntax)

ترکیب (Syntax) دراصل کسی بھی پروگرامنگ لینگویج کے وہ اصول ہیں جن کے تحت کوئی بھی پروگرام لکھا جاتا ہے۔ یہ وہ طریقہ کار ہے جس سے ہم پروگرام کے اندر صحیح ہدایات (Statements) لکھ سکتے ہیں۔ کسی بھی قدرتی زبان کی گرائمر اور پروگرامنگ لینگویج کی ترکیب دونوں ایک ہی چیز ہیں۔ جیسا کہ C لینگویج میں آؤٹ پُٹ کو دکھانے کے لیے ()printf اسٹیٹمنٹ استعمال ہوتی ہے، جس کی ترکیب یہ ہے :

printf("escape sequence/format specifier/string", variable/expression/constant);

پروگرام کا مفہوم (Program Semantic)

سیمنٹک (Semantic) پروگرامنگ لینگویج کے معانی کا علم ہے جو ایک لفظ یا پورے جملے پر لاگو ہو سکتا ہے۔ یہ اُن اقدامات کی مرحلہ وار ترتیب ہے جو کمپیوٹر پروگرام کو چلاتے ہوئے ملحوظ رکھی جاتی ہے۔ مثال کے طور پر ذیل میں دی گئی اسٹیٹمنٹ C کے پروگرام میں دو ویری ایبل x اور y کا حاصل ضرب product کے ویری ایبل میں سٹور کرے گی۔

*product = x*y;*

2.1.2 پروگرامنگ لینگویجز (Programming Languages)

الفاظ، علامات اور کوڈ کا مجموعہ جن کی مدد سے آپ کوئی بھی پروگرام لکھ سکتے ہیں انہیں پروگرامنگ لینگویج کہا جاتا ہے۔ مختلف

NOT FOR SALE

باب 2: "C" پروگرامنگ

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- پروگرام کی ترکیب (Syntax) اور مفہوم (Semantic) واضح کر سکیں۔
- کمپیوٹر کی زبانوں کی مختلف اقسام کو تفصیل سے بیان کر سکیں جیسے ہائی لیول لینگویج اور لو لیول لینگویج۔
- ہائی لیول لینگویج کی خصوصیات کو تفصیلاً بیان کر سکیں۔
- مختلف طرح کی ہائی لیول لینگویج کو جان سکیں۔
- کمپائلر اور انٹرپریٹر کا فرق واضح کر سکیں۔
- انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) کے تصور کو بیان کر سکیں۔
- C لینگویج کے مختلف اجزاء (ایڈیٹر، کمپائلر، لنکر، لوڈر اور ڈی بگر) کو تفصیلاً بیان کر سکیں۔
- ہیڈر فائلز کی تعریف کر سکیں۔
- ریزروڈ ورڈز (Reserved Words) کی پہچان کر سکیں۔
- C پروگرام کے بنیادی ڈھانچے (Structure) کو بیان کر سکیں۔
- Comments کے گرائمر اور مقصد کو سمجھ سکیں۔
- متغیر (Variable) اور مستقل (Constant) میں فرق کر سکیں۔
- Variable کے نام سے متعلق اصولوں کو تفصیلاً بیان کر سکیں۔
- C پروگرام میں استعمال ہونے والی ڈیٹا ٹائپس (Data Types) کو جان سکیں۔
- ٹائپ کاسٹنگ (Typecasting) کا استعمال کر سکیں۔
- *const* کی تعریف کر سکیں۔
- Variable اور Constant بنانے اور ابتدائی مقدار مقرر کرنے کے عمل کو تفصیلاً بیان کر سکیں۔

NOT FOR SALE

2.1 پروگرامنگ کا ایک تعارف (Introduction to Programming)

کمپیوٹر ایک ایسا آلہ ہے جو دی گئی ہدایات کے مطابق عمل کرتا ہے۔ واضح ہدایات کا ایسا مجموعہ جو کمپیوٹر کو فراہم کیا جائے کمپیوٹر پروگرام کہلاتا ہے۔ کمپیوٹر پروگرامنگ ایک ایسا عمل ہے جس میں ہم کسی بھی مسئلے کو حل کرنے کے لیے ایک کمپیوٹر پروگرام بناتے ہیں۔ دنیا میں کمپیوٹر کے لیے بہت ساری زبانیں استعمال ہوتی ہیں جیسے C ، C C++ اور Java وغیرہ۔ ان میں سے ہر ایک کی اپنی کچھ نہ کچھ خصوصیات ہیں۔ کمپیوٹر پروگرام کمپیوٹر میں ہونے والے مختلف عوامل کو کنٹرول کرتا ہے اور یہ کسی نہ کسی پروگرامنگ لینگوئج میں لکھا جاتا ہے تاکہ کوئی مخصوص کام سرانجام دیا جاسکے۔

2.1.1 کمپیوٹر پروگرام (Computer Program)

کمپیوٹر پروگرام ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتا سکے کہ کیا کرنا اور کیسے کرنا ہے۔ کوئی بھی کمپیوٹر، کمپیوٹر پروگرام میں دی گئی ہدایات کے مطابق عمل کرتا ہے۔ کمپیوٹر پروگرام کسی بھی پروگرامنگ لینگوئج میں لکھے جاتے ہیں۔ جو فرد کمپیوٹر پروگرام بناتا ہے اسے پروگرامر کہا جاتا ہے۔ پروگرامر پروگرام بنا کر کمپیوٹر کو ہدایات دیتا ہے کہ کس طرح ڈیٹا کو پروسیس کرے تاکہ وہ انفارمیشن میں تبدیل ہو جائے۔ پروگرامر مختلف پروگرامنگ لینگوئج یا پروگرامنگ آلات کو استعمال کر کے پروگرام بناتا ہے۔ ذیل میں دی گئی دو اصطلاحات کمپیوٹر پروگرام سے متعلق ہیں۔

پروگرام کی ترکیب (Program Syntax)

ترکیب (Syntax) دراصل کسی بھی پروگرامنگ لینگوئج کے وہ اصول ہیں جن کے تحت کوئی بھی پروگرام لکھا جاتا ہے۔ یہ وہ طریقہ کار ہے جس سے ہم پروگرام کے اندر صحیح ہدایات (Statements) لکھ سکتے ہیں۔ کسی بھی قدرتی زبان کی گرائمر اور پروگرامنگ لینگوئج کی ترکیب دونوں ایک ہی چیز ہیں۔ جیسا کہ C لینگوئج میں آؤٹ پٹ کو دکھانے کے لیے printf() اسٹیٹمنٹ استعمال ہوتی ہے، جس کی ترکیب یہ ہے :

```
printf("escape sequence/format specifier/string", variable/expression/constant);
```

پروگرام کا مفہوم (Program Semantic)

سیمنٹک (Semantic) پروگرامنگ لینگوئج کے معانی کا علم ہے جو ایک لفظ یا پورے جملے پر لاگو ہو سکتا ہے۔ یہ اُن اقدامات کی مرحلہ وار ترتیب ہے جو کمپیوٹر پروگرام کو چلاتے ہوئے ملحوظ رکھی جاتی ہے۔ مثال کے طور پر ذیل میں دی گئی اسٹیٹمنٹ C کے پروگرام میں دو ویری ایبل x اور y کا حاصل ضرب product کے ویری ایبل میں سٹور کرے گی۔

```
product = x*y;
```

2.1.2 پروگرامنگ لینگویجز (Programming Languages)

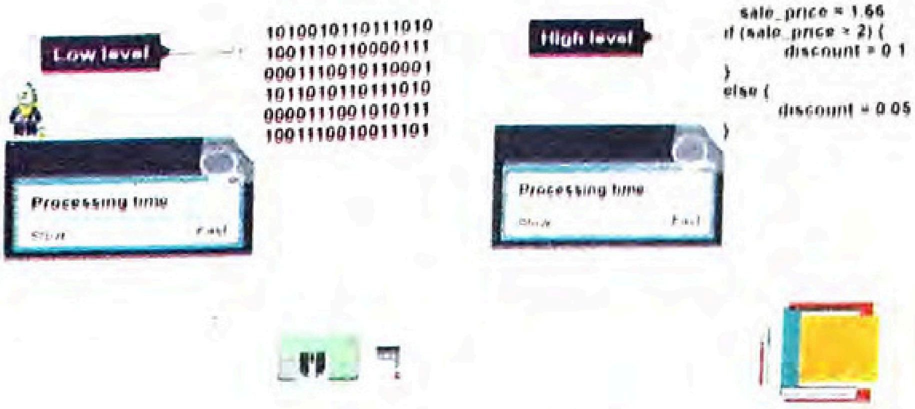
الفاظ، علامات اور کوڈ کا مجموعہ جن کی مدد سے آپ کوئی بھی پروگرام لکھ سکتے ہیں انہیں پروگرامنگ لینگوئج کہا جاتا ہے۔ مختلف

NOT FOR SALE

طرح کے کام کمپیوٹر پر انجام دینے کے لیے مختلف اقسام کی پروگرامنگ لینگویج استعمال ہوتی ہیں۔ یہ لینگویجز کاروباری، سائنسی اور عمومی مسائل کے حل کے لیے استعمال ہوتی ہیں۔

پروگرامنگ لینگویجز کی دو اقسام میں درجہ بندی کی جاسکتی ہے۔

- لولیول لینگویجز (Low Level Languages)
- ہائی لیول لینگویجز (High Level Languages)



شکل نمبر 2.1 لولیول اور ہائی لیول پروگرامنگ لینگویجز

لولیول لینگویجز (Low Level Languages)

کمپیوٹر ہائی لیول لینگویجز یا انگریزی میں دی جانے والی ہدایات کو براہ راست نہیں سمجھ سکتا۔ یہ صرف مشین لینگویج جیسے بائنری لینگویج، میں دی گئی ہدایات کو سمجھ اور چلا سکتا ہے۔ یہ لینگویجز کمپیوٹر ہارڈ ویئر کے بہت قریب ہیں اِس لیے کمپیوٹر اِن لینگویجز کو باآسانی سمجھ سکتا ہے۔ ان لینگویج میں پروگرامنگ کرنے کے لیے کمپیوٹر کے ہارڈ ویئر کی ساخت سے متعلق بہت گہرا مطالعہ ہونا ضروری ہے۔

لولیول لینگویجز کی دو اقسام ہیں۔

1. مشین لینگویج (Machine Language)

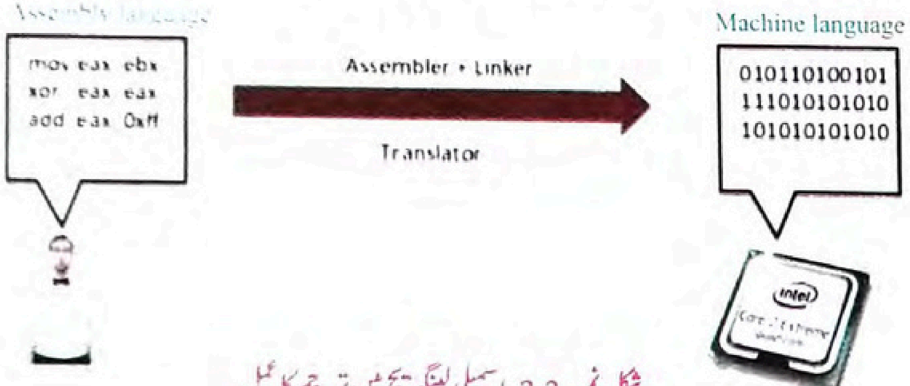
زبان کی وہ قسم جس میں ہم ہدایات کو بائنری شکل میں لکھتے ہیں مشینی زبان کہلاتی ہے۔ یہ کمپیوٹر کی وہ واحد زبان ہے جسے کمپیوٹر براہ راست سمجھ جاتا ہے۔ یہ کمپیوٹر کی بنیادی یا اساسی زبان ہے۔ اس زبان میں لکھے گئے پروگرام کمپیوٹر پر بہت تیزی کے ساتھ چلتے ہیں۔ کمپیوٹر کے اندر مشین لینگویج کو بائنری اعداد (bits) یعنی 0 اور 1 کے تسلسل سے ظاہر کیا جاتا ہے۔ 0 کی علامت چارج کی غیر موجودگی اور 1 چارج کی موجودگی کو ظاہر کرتا ہے۔ انسان کے لیے مشینی زبان سمجھنا زیادہ مشکل ہے۔ اس زبان میں پروگرام لکھنا یا لکھے ہوئے پروگرام میں ترمیم کرنا بہت دیر طلب کام ہے۔ مشینی زبان کو پہلی نسل کی زبان (First Generation Language) بھی کہا جاتا ہے۔

2. اسمبلی لینگویج (Assembly Language)

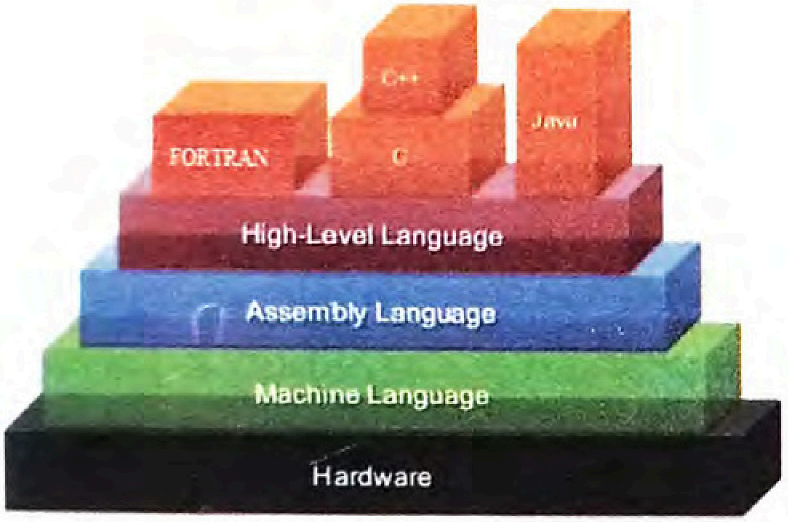
اسمبلی لینگویج بھی لولیول لینگویج ہے۔ یہ مشین لینگویج سے ایک درجہ اوپر ہے۔ اس لینگویج میں بائنری کوڈ کی بجائے علامات کو

NOT FOR SALE

استعمال کیا جاتا ہے۔ ان علامات کو (نمونکس) mnemonics کہا جاتا ہے۔ مثال کے طور پر دو اعداد کی تفریق کے لیے SUB اسٹیٹمنٹ استعمال کی جاتی ہے۔ اسے علاماتی زبان (Symbolic Language) بھی کہا جاتا ہے۔ مشینی زبان کے مقابلے میں اسمبلی لینگویج میں پروگراموں کو لکھنا یا ان میں ترمیم کرنا زیادہ آسان ہے۔ اسمبلی لینگویج میں لکھے کوڈ کو پہلے مشین لینگویج میں ترجمہ کرنے کے لیے اسمبلر کو استعمال کیا جاتا ہے اور پھر اسے کمپیوٹر پر چلایا جاتا ہے۔ اسمبلی لینگویج زیادہ تر سسٹم سافٹ ویئرز بنانے کے لیے استعمال کی جاتی ہے۔ اس کو دوسری نسل کی زبان (Second Generation Language) بھی کہا جاتا ہے۔



شکل نمبر 2.2 اسمبلی لینگویج میں ترجمے کا عمل



شکل نمبر 2.3 مختلف پروگرامنگ لینگویجز

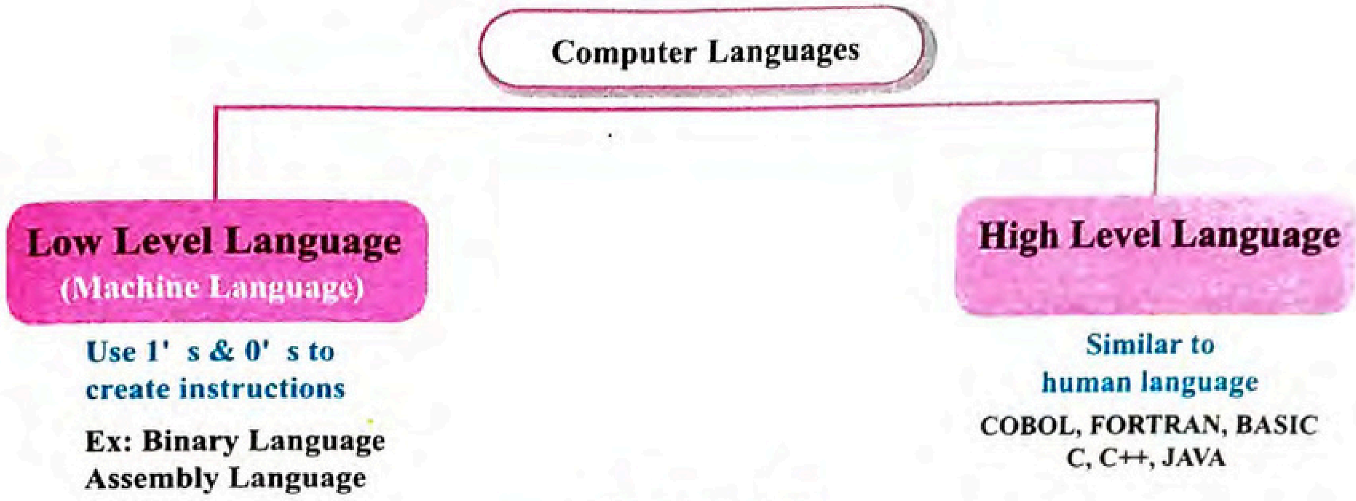
ہائی لیول لینگویجز (High Level Languages)

ہائی لیول لینگویجز انگریزی سے ملتے جلتے الفاظ پر مشتمل ہوتی ہے۔ ہائی لیول لینگویج بنانے کا مقصد یہ تھا کہ لوگ اپنی زبان یعنی انگریزی میں آسانی سے پروگرام لکھ سکیں۔

ہائی لیول لینگویج بنیادی طور پر نمونکس کوڈ کے بجائے انگریزی کے الفاظ اور حسابی علامات اِستعمال کرتی ہے۔ دنیا میں بہت ساری ہائی لیول لینگویجز ہیں جن میں پروگرام لکھنے کے قوانین اور گرائمر ایک دوسرے سے مختلف ہیں۔ ہائی لیول لینگویج میں لکھے گئے پروگرام کو کمپیوٹر پر چلانے سے پہلے مشین کوڈ میں ترجمہ کیا جاتا ہے۔ ہر ہائی لیول لینگویج کا ترجمہ کرنے کے لیے ایک مخصوص مترجم (Translator)

NOT FOR SALE

پروگرام ہوتا ہے۔ ہائی لیول لینگوئج کی چند مثالیں C, Java, Visual Basic اور Pascal ہیں۔ ہائی لیول لینگوئج کو درج ذیل اقسام میں تقسیم کیا جاسکتا ہے۔



شکل نمبر 2.4 کمپیوٹر لینگویجز

- پروسیجرل لینگوئج (Procedural Language)
- سٹرکچرڈ لینگوئج (Structured Language)
- آبجیکٹ اورینٹڈ پروگرامنگ لینگوئج (Object-Oriented Programming Language)

پروسیجرل لینگویجز (Procedural Languages)

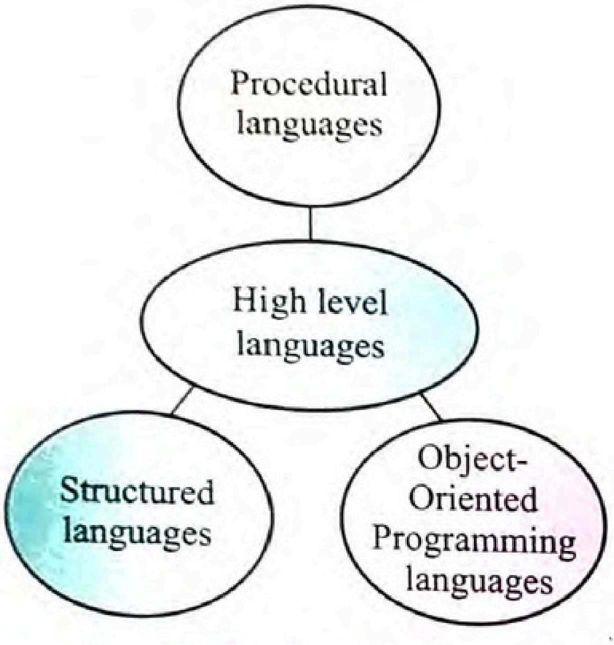
پروسیجرل لینگوئج کو تیسری نسل کی زبان (Third Generation Language) بھی کہا جاتا ہے۔ اس زبان میں پروگرام کو پروسیجرز یا ماڈلز یو کی شکل میں تیار کیا جاتا ہے۔ پروسیجر ہدایات کی وہ ترتیب ہے جس کا کوئی نہ کوئی منفرد نام ہو۔ پروسیجرز کو چلانے کے لیے اس کے نام سے پکارا جاتا ہے۔ پروسیجر ایک دفعہ لکھا جاتا ہے لیکن اسے پروگرام میں جتنی دفعہ درکار ہو استعمال کیا جاسکتا ہے اور اس کا کوڈ بار بار لکھنے کی ضرورت نہیں پڑتی۔

پروسیجرل لینگوئج کی چند مثالیں C، Pascal، FORTRAN اور BASIC ہیں۔

سٹرکچرڈ لینگویجز (Structured languages)

ایسی پروگرامنگ لینگوئج جس میں پروگرام کو چھوٹے چھوٹے حصوں، بلاک یا پروسیجرز میں تقسیم کیا جاسکے اسے سٹرکچرڈ پروگرامنگ لینگوئج کہا جاتا ہے۔ اس طرح کے پروگرام کی ترتیب اوپر سے نیچے کی طرف ہوتی ہے۔ اس طرح کی زبانوں کی عام مثالیں ALGOL، PL/1، Pascal اور ADA ہیں۔

NOT FOR SALE



شکل نمبر 2.5 ہائی لیول لنگویجز

آبجیکٹ اورینٹڈ پروگرامنگ لینگوئج (Object Oriented Programming Language)

آبجیکٹ اورینٹڈ پروگرامنگ میں، پروگرام آبجیکٹس کے ایک سیٹ کا استعمال کر کے تیار کیا جاتا ہے۔ ایک آبجیکٹ پروگرام ایک ایسا جزو ہے جس سے ماڈلز یو کا ایک سیٹ اور ڈیٹا سٹرکچر منسلک ہوتے ہیں۔ یہ پروگرام کو ڈیزائن کرنے کے لئے ایک جدید اور آسان تکنیک ہے۔ ایک بار جب کسی بھی پروگرام کے لئے کوئی آبجیکٹ ڈیزائن ہو جائے تو پھر اسے کسی بھی دوسرے پروگرام میں دوبارہ استعمال کیا جاسکتا ہے۔ کچھ مقبول اور عام استعمال ہونے والی آبجیکٹ اورینٹڈ پروگرامنگ (OOP) لینگویجز کی مثالیں ++ C اور JAVA ہیں۔

2.1.3 ہائی لیول پروگرامنگ لینگویجز کی خصوصیات (Characteristic of High Level Programming Languages)

ہائی لیول پروگرامنگ لینگویجز کی چند خصوصیات درج ذیل ہیں۔

سیکھنے میں آسان (Easy to Learn)

لو لیول پروگرامنگ کے مقابلے میں ہائی لیول پروگرامنگ لینگویجز سیکھنے میں بہت آسان ہوتی ہیں۔ پروگرام کے لیے لکھی گئی اسٹیٹمنٹس بالکل انگریزی زبان جیسے جملوں کی طرح ہی ہوتی ہیں۔

سمجھنے میں آسان (Easy to Understand)

ہائی لیول پروگرامنگ لینگوئج میں ایک پروگرامر کا لکھا ہوا کوئی بھی پروگرام دوسرا پروگرامر آسانی سے سمجھ سکتا ہے۔ کیونکہ پروگرام میں لکھی گئی ہدایات انگریزی زبان کی طرح ہی ہوتی ہیں۔

NOT FOR SALE

پروگرام لکھنے میں آسانی (Easy to Write Program)

ہائی لیول پروگرامنگ لینگویج میں، ایک نیا پروگرام آسانی سے بہت ہی کم وقت میں لکھا جاسکتا ہے۔

بڑے اور پیچیدہ سافٹ ویئر کچھ دنوں یا مہینوں میں تیار کیے جاسکتے ہیں۔

غلطیوں کا پتہ لگانے اور انہیں دور کرنے میں آسانی (Easy to Detect & Remove Errors)

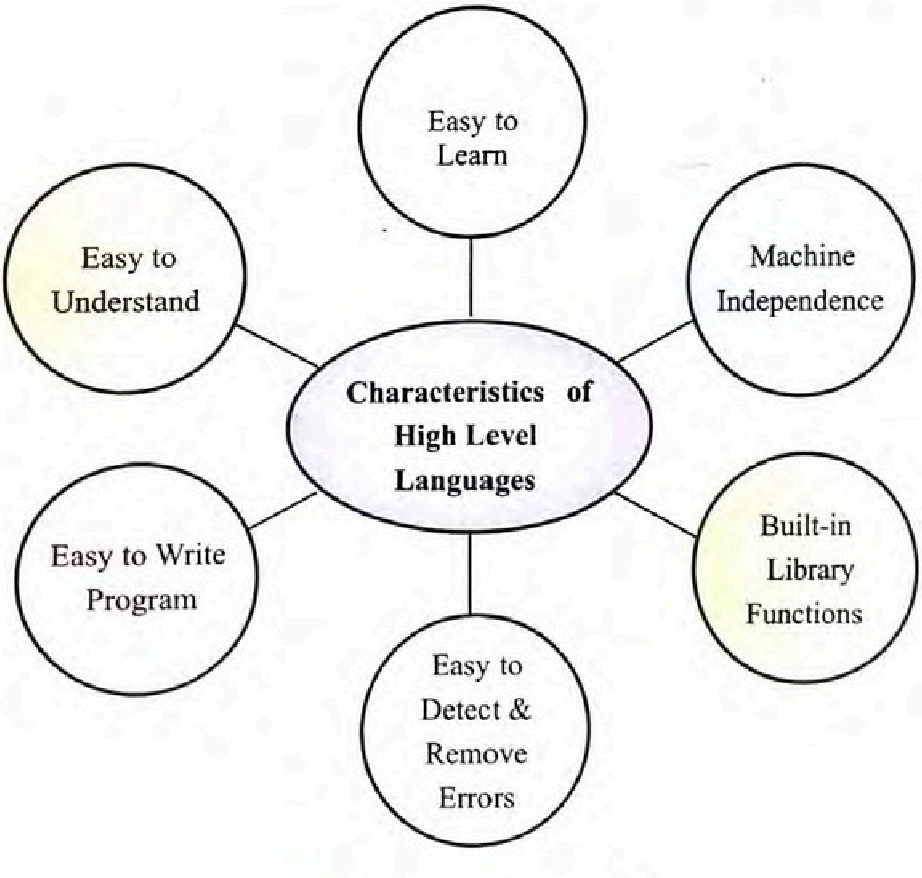
پروگرام میں موجود غلطیوں کو آسانی سے معلوم کیا جاسکتا ہے اور انہیں دور بھی کیا جاسکتا ہے۔ زیادہ تر غلطیاں نئے پروگرام کو کمپائل کرنے کے دوران ظاہر ہو جاتی ہیں۔

بنے بنائے لائبریری فنکشنز (Built-in Library Functions)

ہر ہائی لیول پروگرامنگ لینگویج میں بڑی تعداد میں بلٹ ان فنکشنز یا طریقہ کار مہیا ہوتے ہیں جو نئے پروگراموں کی ڈیزائننگ کے دوران مخصوص کاموں کو انجام دینے کے لئے استعمال ہوتے ہیں۔ اس طرح سے، پروگرامر کا بہت سا وقت بچ جاتا ہے۔

مشین پر غیر انحصاری (Machine Independence)

ہائی لیول لینگویج میں لکھا ہوا پروگرام کسی خاص مشین پر منحصر نہیں ہوتا۔ اس کا مطلب یہ ہے کہ ایک قسم کے کمپیوٹر پر لکھا ہوا ایک پروگرام دوسری قسم کے کمپیوٹر پر بآسانی چلایا جاسکتا ہے۔



شکل نمبر 2.6 ہائی لیول لینگویج کی خصوصیات

2.1.4 چند مقبول ہائی لیول پروگرامنگ لینگویجز (Some Popular High Level Programming Languages)

ڈینس رچی، بانی C لینگویج

C لینگویج (C Language)

C ایک عمومی مقاصد کے لیے استعمال ہونے والی کمپیوٹر پروگرامنگ لینگویج ہے جو 1972ء میں امریکہ کے اے ٹی اینڈ ٹی بیل لیبارٹریز میں ڈینس رچی (Denis Richie) نے تیار کی۔

C ایک سادہ اور سٹرکچرڈ اور ڈیزائنڈ پروسیجرل پروگرامنگ لینگویج ہے۔ یہ مختلف پلیٹ فارم پر آسانی سے منتقل ہو سکتی ہے۔ اور اسی وجہ سے صنعت اور پیشہ ورانہ پروگرامنگ میں وسیع پیمانے پر استعمال ہوتی ہے۔ یونیکس UNIX پر آپریٹنگ سسٹم C میں تیار کیا گیا ہے۔

C ++ لینگویج (C++ Language)

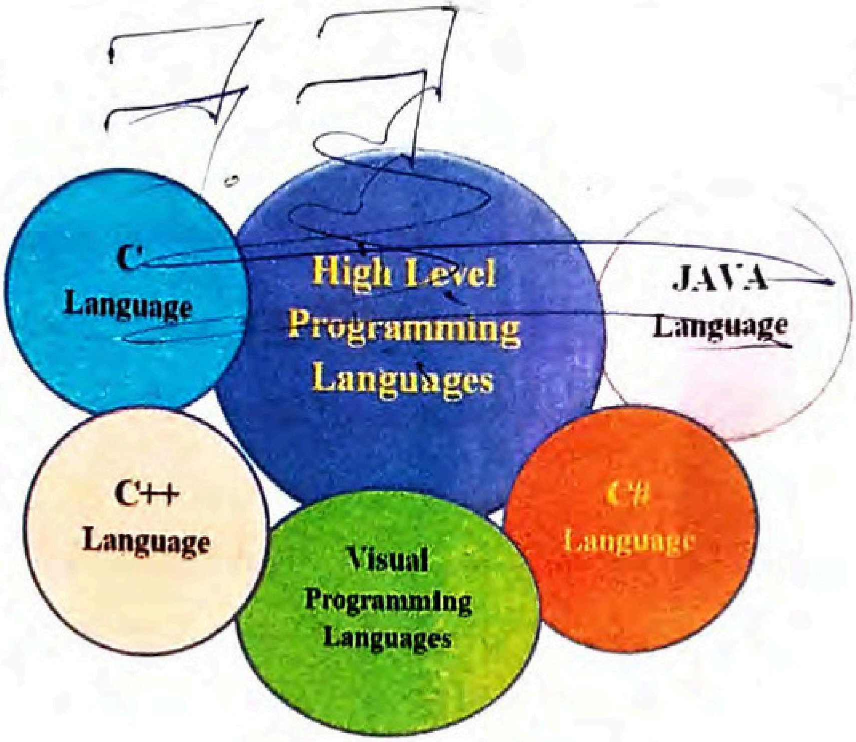
C++ 1980ء کی دہائی کے شروع میں بجرنے سٹراسٹرپ (Bjarne Stroustrup) نے تیار کی تھی۔ یہ ایک آبجیکٹ اورینٹڈ پروگرامنگ (OOP) لینگویج ہے۔ یہ C لینگویج کی ہی توسیع ہے۔ C ++ میں C لینگویج کے تمام عناصر شامل ہیں اور ساتھ ہی ساتھ آبجیکٹ، کلاسز، ایونٹ اور دیگر آبجیکٹ اورینٹڈ کے تمام تصورات کے ساتھ کام کرنے کی اضافی خصوصیات ہیں۔ زیادہ تر ایپلی کیشن پروگرام، جیسے ورڈ پروسیسرز اور اسپریڈ شیٹ C ++ میں ہی تیار کی گئی ہیں۔

وژول پروگرامنگ لینگویجز (Visual Programming Languages)

وژول پروگرامنگ کی اصطلاح ونڈوز پر مبنی ایپلی کیشنز یا گرافیکل یوزر انٹر فیس (GUI) ایپلی کیشنز بنانے یا تیار کرنے کے لیے استعمال ہوتی ہے۔ وژول پروگرامنگ لینگویجز، جن کو بعض اوقات پانچویں نسل کی پروگرامنگ لینگویج بھی کہا جاتا ہے، آبجیکٹ اورینٹڈ پروگرامنگ لینگویجز ہیں۔ اس پروگرامنگ لینگویج میں وژول یا گرافیکل یوزر انٹر فیس (GUI) کی سہولیات ہونے کی وجہ سے یہ GUI والی ایپلی کیشنز تیار کرنے کے لیے نہایت مفید ہے۔

بہت مقبول اور وسیع پیمانے پر استعمال ہونے والی وژول پروگرامنگ لینگویجز کی مثال وژول بیسک (VB)، وژول C ++ ، C# اور Java ہیں۔

NOT FOR SALE



شکل نمبر 2.7 مقبول ہائی لیول پروگرامنگ لینگویجز

C # لینگویج (C# Language)

C# یا سی شارپ (C-Sharp) ایک آبجیکٹ اورینٹڈ پروگرامنگ لینگویج ہے۔ اسے مائیکروسافٹ کارپوریشن نے سن 2000ء میں تیار کیا۔ یہ Visual Basic اور Java کے عناصر اور اجزاء کے ساتھ C ++ پر مبنی ہے۔ اس کا استعمال ایسی ویب ایپلی کیشنز میں کیا جاتا ہے جسے انٹرنیٹ پر استعمال کیا جا سکے۔ C# میں کمپیوٹر گیم، یوٹیلٹی پروگرام، آپریٹنگ سسٹم، کمپائلر، کاروباری ایپلی کیشنز اور ویب ایپلی کیشنز سمیت تمام قسم کے پروگرام تیار کیے جاسکتے ہیں۔

جاوا لینگویج (JAVA Language)

Java ایک ہائی لیول آبجیکٹ اورینٹڈ پروگرامنگ لینگویج ہے جو سن مائیکروسسٹمز (Sun Microsystems) نے 1991ء میں تیار کی۔

یہ بھی C ++ کی طرح ہی ہے لیکن یہ خاص طور پر ویب پیج ڈیزائننگ کے لئے استعمال کی جاتی ہے۔

جاوا ایپلیٹس (Applets) کو جاوا لینگویج میں ہی تیار کیا جاتا ہے۔ جاوا ایپلیٹ ایک چھوٹا جاوا پروگرام ہے جو کسی اور پروگرام جیسے ویب براؤزر میں چلایا جاتا ہے۔ جاوا ایپلیٹ کو ویب پیجز کے اندر سمو دیا جاتا ہے تاکہ انٹرنیٹ پر کسی تصویر یا امیج اینیمیشن کو متحرک دکھایا جا سکے اور کسی ڈیٹا بیس تک رسائی حاصل کی جا سکے۔

2.1.5 کمپائلر اور انٹرپرٹر (Compiler and Interpreter)

کمپائلر (Compiler)

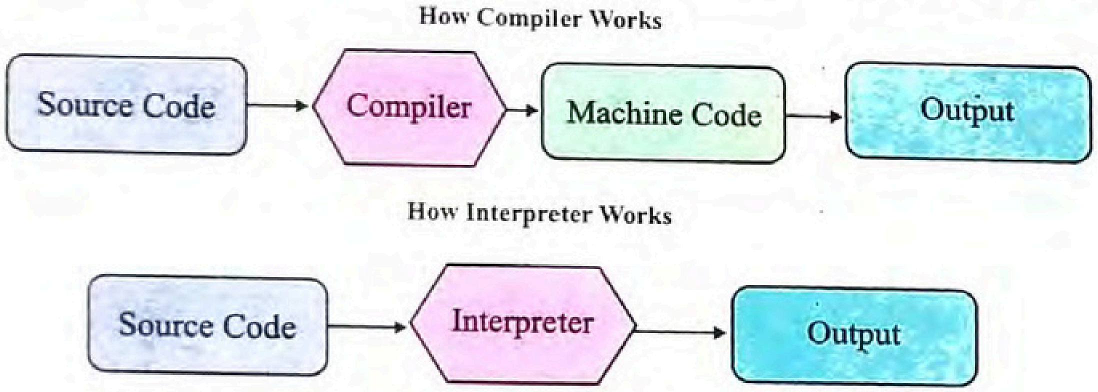
کمپائلر ایک سسٹم سافٹ ویئر ہے جو ہائی لیول پروگرامنگ لینگویج کے سورس کوڈ (Source Code) کو مشین لینگویج آبجیکٹ کوڈ

میں ترجمہ کرتا ہے۔ یہ انگریزی کی طرح ++C لینگویج میں لکھے ہوئے کوڈ کو مشین لینگویج (بائنری بندسے 0 اور 1) میں ترجمہ کرتا ہے جسے کمپیوٹر براہ راست سمجھتے ہیں۔ یہ پہلے پورے پروگرام کو مشین لینگویج میں تبدیل کرتا ہے اور پھر اس کے بعد اسے چلاتا ہے۔ پروگرامنگ لینگویجز جیسے ++C / C، Pascal اور Java کمپائلر کو ہی استعمال کرتی ہیں۔

انٹرپرٹر (Interpreter)

انٹرپرٹر بھی ہائی لیول پروگرامنگ لینگویج کا مترجم ہے جو پروگرام کی اسٹیٹمنٹ کو ایک ایک کرکے ترجمہ کرتا ہے اور ساتھ ہی ساتھ چلاتا بھی ہے۔ یہ پروگرام کی ایک اسٹیٹمنٹ کو لے کر اسے مشین لینگویج میں تبدیل کرتا ہے، اس پر عمل درآمد کرتا ہے، اور پھر اگلی اسٹیٹمنٹ پر چلا جاتا ہے۔

ہر دفعہ جب کوئی اسٹیٹمنٹ پڑھی جائے گی تو اسے چلانے سے پہلے ضرور مشین لینگویج میں ترجمہ کیا جائے گا۔ کمپائلر کمپیوٹر پر کسی پروگرام کو چلانے سے پہلے پورے پروگرام کو ایک ہی بار پڑھتا ہے اور آبجیکٹ پروگرام میں ترجمہ کرتا ہے۔ انٹرپرٹر کا استعمال کرنے والی پروگرامنگ لینگویجز کی مثالوں میں جاوا اسکرپٹ (JAVA Script)، BASIC، Visual Basic اور Perl شامل ہیں۔



شکل نمبر 2.8 کمپائلر اور انٹرپرٹر کیسے کام کرتے ہیں

2.2 پروگرامنگ کا انوائرمنٹ (Programming Environment)

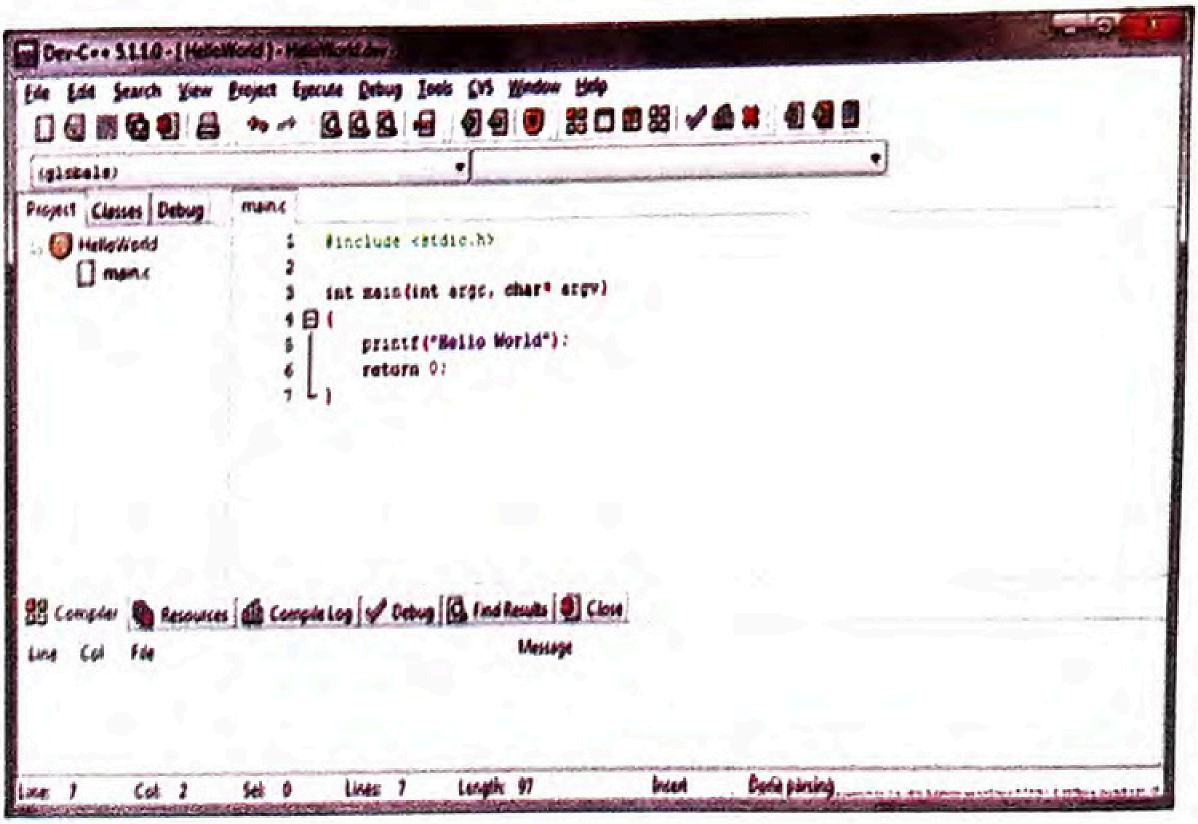
پروگرامنگ انوائرمنٹ پروگرامنگ ٹولز کا ایک مجموعہ ہے جو کمپیوٹر پروگرام بنانے، کمپائل کرنے اور انہیں چلانے کے لیے استعمال کیا جاتا ہے۔ اس میں ایڈیٹر (Editor)، کمپائلر (Compiler)، لنکر (Linker)، ڈی بگر (Debugger) وغیرہ شامل ہیں۔ اسے انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) کا نام بھی دیا جاتا ہے جو کمپیوٹر پروگرام بنانے میں مدد کرتا ہے۔ انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ درحقیقت ایک ایپلیکیشن سافٹ ویئر ہے جو گرافیکل یوزر انٹرفیس (GUI) مہیا کرتا ہے۔

2.2.1 انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE)

انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) پروگراموں کو تیار کرنے، مرتب کرنے اور چلانے کے لئے پروگرامنگ لینگویج

NOT FOR SALE

ڈویلپمنٹ ٹول ہے۔ IDE کمپیوٹر سافٹ ویئر ہے جہاں پروگرام بنانے کے لئے درکار تمام عوامل اور آلات کو یکجا کیا گیا ہے۔ آج کل کے جدید IDEs میں پروگرامرز کے لیے بہت ہی آسان اور خوبصورت GUI موجود ہیں۔C لینگویج کے لیے بھی ایک IDE استعمال ہوتا ہے، جیسا کہ شکل نمبر 2.9 میں دکھایا گیا ہے۔



شکل نمبر 2.9 C لینگویج کا IDE

2.2.2 C پروگرامنگ انوائرمنٹ کے مختلف اجزاء (Modules of the C Programming Environment)

ذیل میں C لینگویج میں پروگرامنگ کے لیے درکار اجزاء بیان کیے گئے ہیں۔

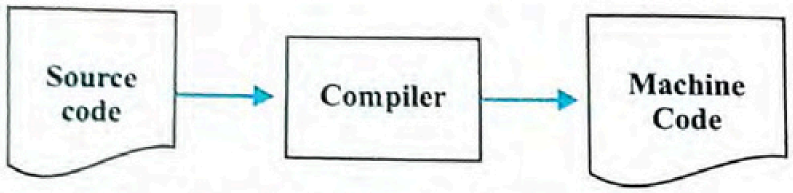
1. ایڈیٹر (Editor)

ایڈیٹر یا ٹیکسٹ ایڈیٹر ایک ایسا پروگرام ہے جو صارفین کو C لینگویج سورس پروگرام بنانے اور اس میں ترمیم کرنے کے قابل بناتا ہے۔C کے پروگرام کی ایکسٹینشن *c.* ہوتی ہے۔ مثلاً student.c ایسا ہی ایک پروگرام ہے۔ تقریباً ہر ایک ایڈیٹر پروگرام کمپائل کرنے کے دوران پروگرام میں موجود تمام غلطیوں کو واضح کرتا ہے تاکہ انہیں آسانی سے دور کیا جاسکے۔

2. کمپائلر (Compiler)

ایک ایسا کمپیوٹر سافٹ ویئر ہے جو C لینگویج پروگرام سورس کوڈ (Source Code) کو مشین کوڈ (آبجیکٹ پروگرام) میں ترجمہ کرتا ہے تاکہ کمپیوٹر اسے سمجھ کر چلا (execute) سکے۔C لینگویج کے کسی بھی کمپائل شدہ پروگرام کی ایکسٹینشن (پہچان) ***obj.*** ہوتی ہے۔ مثلاً ***student.obj***۔ کمپائل کرنے کے عمل کو ذیل میں شکل نمبر 2.10 میں دکھایا گیا ہے۔

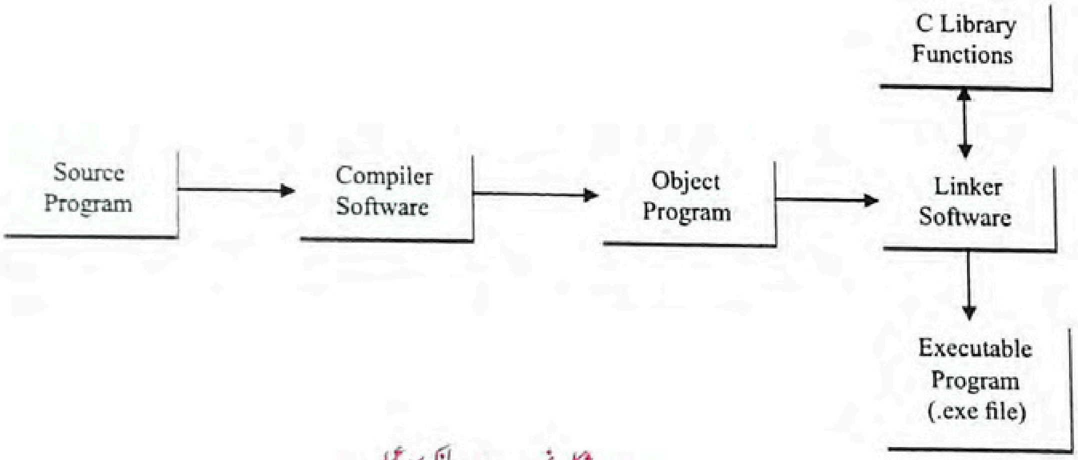
NOT FOR SALE



شکل نمبر 2.10 سورس کوڈ کو کمپائل کر کے مشین کوڈ میں تبدیل کرنے کا عمل

3. لنکر (Linker)

ایک کمپیوٹر پروگرام جو ایک یا ایک سے زیادہ آبجیکٹ کوڈز لیتا ہے اور ان کو ایک ہی قابل عمل پروگرام میں جوڑتا ہے، اسے لنکر کہتے ہیں۔ C لینگوئج ایگزیکیوٹیبل پروگرام کی ایکسٹینشن exe. ہے جیسے student.exe لنکر کے پورے عمل کو شکل نمبر 2.11 میں دکھایا گیا ہے۔



شکل نمبر 2.11 لنکر کا عمل

4. لوڈر (Loader)

یہ ایک پروگرام ہے جو C زبان کے پروگراموں کو میموری میں لوڈ کرتا ہے اور پھر ان پر عمل درآمد کرتا ہے۔

5. ڈیبگر (Debugger)

کمپیوٹرز میں ڈیبگنگ، کمپیوٹر پروگرام کوڈ میں موجود غلطیوں (Errors) کو تلاش کرنے اور ٹھیک کرنے کے عمل کو کہتے ہیں۔

C لینگوئج ڈیبگر ایک کمپیوٹر پروگرام ہے جو صارفین کو پروگرام چلانے دیتا ہے، متغیر کی اقدار کی جانچ پڑتال یا فنکشن کو بھیجی گئی اقدار کو دیکھنے اور پروگرام میں غلطیوں کو ڈھونڈنے اور دور کرنے میں مدد کرتا ہے۔

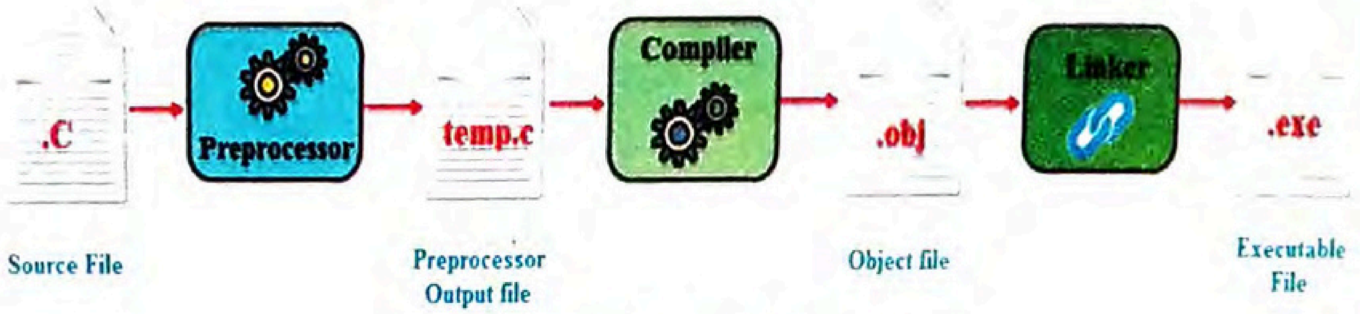
2.3 پروگرامنگ کی اساسیات

C پروگرامنگ کا آغاز (Getting started with C)

C ایک مشہور اور مؤثر پروگرامنگ زبان ہے جو سسٹم اور ایپلی کیشن سافٹ ویئر کے لئے وسیع پیمانے پر استعمال ہوتی ہے۔

NOT FOR SALE

C پروگرام بنانے اور چلانے کے لیے، آپ کو C لینگوئج کمپائیلر کی ضرورت ہوتی ہے۔ C لینگوتج کے لیے بہت سارے مفت لیکن پیشہ ورانہ کمپائلر دستیاب ہیں۔



شکل نمبر 2.12 کمپائل اور لنک کرنے والا C پروگرام

C کے ہر مکمل پروگرام میں ایک فنکشن ہوتا ہے جس کو main() (مین) فنکشن کہا جاتا ہے۔ فنکشن در حقیقت بہت ساری اسٹیٹمنٹس کا مجموعہ ہوتا ہے جو کوئی نہ کوئی کام انجام دینے کے لئے استعمال ہوتا ہے۔ جب پروگرام چلتا ہے تو سب سے پہلے main فنکشن ہی چلایا جاتا ہے۔ اور فنکشن کے ذریعے ہم دوسرے فنکشن چلا سکتے ہیں، چاہے وہ یوزر کے بنائے ہوئے ہوں یا بلٹ ان (Built-in)۔ C کمپائلر کے ساتھ آنے والے سٹینڈرڈ فنکشنز تک رسائی کے لیے آپ کو include# کی مدد سے ہیڈر (header) فائل شامل کرنے کی ضرورت ہوتی ہے۔

2.3.1 ہیڈر فائل (Header File)

مختلف کاموں کو انجام دینے کے لئے ہیڈر فائلیں سٹینڈرڈ لائبریری فنکشن کا ایک مجموعہ ہیں۔ مختلف مقاصد کے لئے بہت ساری ہیڈر فائلیں ہیں۔ ہر ہیڈر فائل میں مختلف قسم کے بنے بنائے فنکشن ہوتے ہیں۔ ایک ہی پروگرام میں بہت ساری ہیڈر فائلوں کو شامل کیا جاسکتا ہے۔ پروگرام میں ہیڈر فائل کو اپنے اندر موجود کسی بھی فنکشن کے استعمال سے پہلے پہلے، شامل کرنا ضروری ہے۔

ہیڈر فائل کی .h ایکسٹینشن ہے۔ پروگراموں میں ہیڈر فائلوں کو شامل کرنے کے لئے include# پری پروسیسر ڈائریکٹو استعمال کیا جاتا ہے۔ یہ فائلیں C کمپائلر سسٹم میں پہلے سے دستیاب ہوتی ہیں۔

ہیڈر فائلوں کو عام طور پر INCLUDE سب ڈائریکٹری میں محفوظ کیا جاتا ہے۔ ہیڈر فائل کا نام Angle Bracket <> میں لکھا جاتا ہے۔

مثال کے طور پر

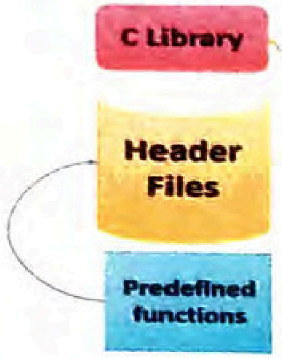
```
# include <stdio.h>
```

ہر ہیڈر فائل میں فنکشن کے ایک خاص گروپ کے لئے معلومات شامل ہوتی ہیں۔ stdio.h ہیڈر فائل کی طرح C میں سٹینڈرڈ اِن پٹ اور آؤٹ پٹ فنکشن کے اعلامیہ پر مشتمل ہے جو اِن پٹ حاصل کرنے اور آؤٹ پٹ کو پرنٹ کرنے کے لئے استعمال ہوتا ہے۔ اسی

طرح، ہیڈر فائل <math.h> ریاضی کے فنکشن کو استعمال کرنے کے لیے شامل کی جاتی ہیں۔ مثلاً

```
# include <math.h>
```

C Program



شکل نمبر 2.13 C زبان میں ہیڈر فائلیں

2.3.2 کلیدی الفاظ (Reserved Words)

وہ الفاظ جو کسی بھی زبان کے ذریعہ خصوصی مقاصد کے لئے استعمال ہوتے ہیں انہیں کلیدی الفاظ یا ریزروڈ ورڈز (Reserved Words) یا (Keywords) کہتے ہیں۔ ہر Keyword کے معنی مستقل ہوتے ہیں اور یوزر کے ذریعہ اسے تبدیل نہیں کیا جاسکتا۔ ہر Keyword کو چھوٹے حروف میں لکھا جاتا ہے جیسے آٹو (auto)، بریک (break) وغیرہ۔ C لینگویج 32 Keywords کو سپورٹ کرتی ہے۔ ذیل میں دیئے گئے جدول میں چند ایک Keywords لکھے گئے ہیں۔

C لینگویج میں 32 کلیدی الفاظ

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

2.3.3 C پروگرام کا بنیادی ڈھانچہ

C پروگرام میں تین اہم حصے ہوتے ہیں:

- پری پروسیسر ڈائریکٹیوز (Preprocessor Directives)
- مین فنکشن (The main () function)
- مین فنکشن کی باڈی (Body of main { })

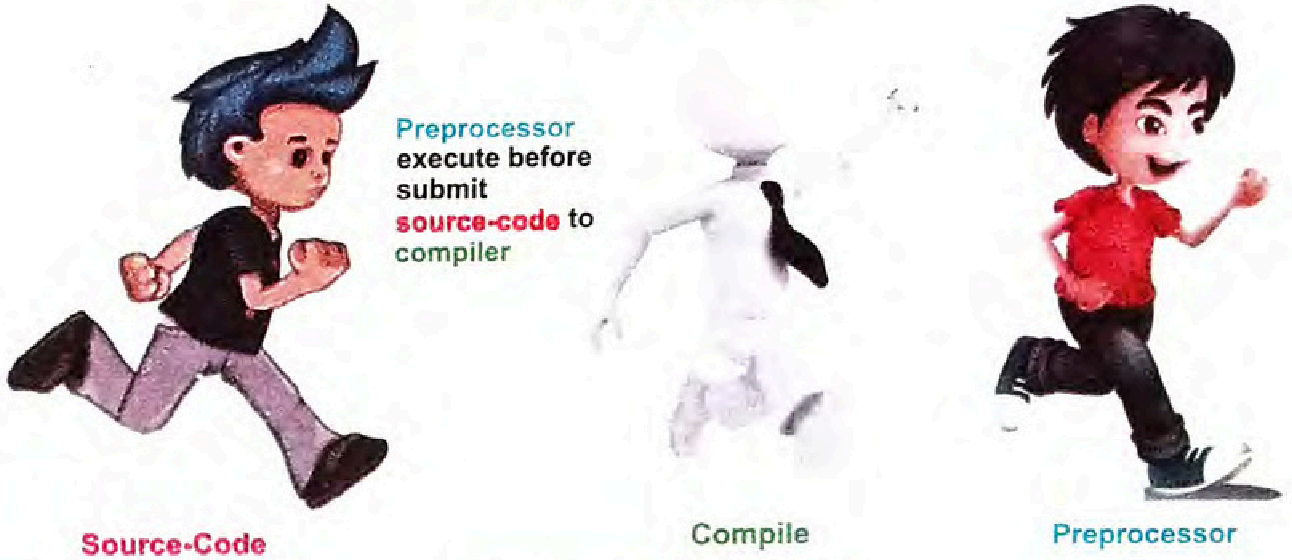
NOT FOR SALE

پری پروسیسر ڈائریکٹیوز (Preprocessor Directives)

اصل پروگرام کے آغاز سے پہلے جو ہدایات کمپائلر کو دی جاتی ہیں ان کو پری پروسیسر ڈائریکٹو کہا جاتا ہے۔ انھیں کمپائلر ڈائریکٹیو بھی کہتے ہیں۔ پری پروسیسر ڈائریکٹیو کمپائلر کی ہدایات پر مشتمل ہوتی ہیں۔ کمپائلیشن کے وقت ان ہدایات سے خصوصی ہدایات یا کوڈ کو پروگرام میں شامل کیا جاتا ہے۔

یہ پری پروسیسر ڈائریکٹیو عام طور پر ہیش سائن # اور مطلوبہ الفاظ، جیسے، include#، define# وغیرہ سے شروع ہوتی ہیں۔

include# پری پروسیسر ڈائریکٹیوز



شکل نمبر 2.14 C لینگویج میں پری پروسیسر کی مثال

include# کمپائلر کو کسی بھی بیرونی (External) ہیڈر فائلوں کو پروسیس کرنے کی اجازت دیتی ہے۔ جب یہ پری پروسیسر C کمپائلر کے ذریعہ انجام پاتا ہے، تو وہ ہیڈر فائل تلاش کرتا ہے جو زاویائی بریکٹ < > میں لکھا ہوا ہے اور اسے سورس فائل کے اندر کاپی کر دیتا ہے۔

سائنٹکس:

```
#include <header-file.h>
یا
#include "header-file.h"
```

مثالیں:

```
#include<stdio.h>
#include<conio.h>
```

NOT FOR SALE

کچھ اہم ہیڈر فائلیں:

<stdio.h>

stdio.h ہیڈر فائل سٹینڈرڈ ان پٹ آؤٹ پٹ ہیڈر کے لئے ہے۔ اس میں پروگرام میں سٹینڈرڈ ان پٹ / آؤٹ پٹ فنکشنز یعنی () scanf اور () printf شامل ہیں۔

<conio.h>

یہ کنسول ان پٹ / آؤٹ پٹ کے لئے ہے۔ C پروگرامنگ میں استعمال ہونے والی conio.h ہیڈر فائل کنسول ان پٹ / آؤٹ پٹ فنکشنز پر مشتمل ہوتی ہے۔ conio.h کے عام طور پر استعمال ہونے والے کچھ فنکشنز ہیں () clrscr ، () getch ، getche () وغیرہ۔ conio.h کے فنکشنز کو اسکرین کو صاف کرنے، ٹیکسٹ اور رنگ تبدیل کرنے، ٹیکسٹ کو حرکت دینے، کونسی key دبائی گئی ہے اس کا پتہ لگانے اور اس کے علاوہ بہت سارے کاموں کے لیے استعمال کیا جاتا ہے۔

<math.h>

math.h ہیڈر فائل ریاضی کے عمل کو انجام دینے کے لئے استعمال ہوتے ہیں۔ یوزر پروگراموں میں ریاضی کے فنکشنز استعمال کرسکتے ہیں، اعداد کی مطلق قیمت کا حساب لگانے، لاگرتھم معلوم کرنے اور کسی زاویہ کے sine ، cosine معلوم کرنے کے لئے ٹرگونومیٹرک فنکشن استعمال کرتے ہیں۔

<string.h>

سٹرنگ لائبریری string.h میں ایسے مفید فنکشنز ہوتے ہیں جن کو سٹرنگ کے لیے استعمال کیا جاتا ہے۔ جیسے () strcpy ، () strcat ، () strcmp ، () strlen ، وغیرہ۔

main() فنکشن

یہ فنکشن ہر C پروگرام میں ہمیشہ موجود رہنا چاہئے۔ main فنکشن سے ہی C پروگرام پر عمل درآمد شروع ہوتا ہے۔
جب C کا کوئی بھی پروگرام چلایا جاتا ہے تو کنٹرول سب سے پہلے براہ راست () main فنکشن میں منتقل ہو جاتا ہے۔ اگر پروگرام میں مین فنکشن نہ ہو تو پروگرام کمپائل نہیں ہوگا اور غلطی کا پیغام (Error Message) ظاہر ہوگا۔

main() فنکشن کا سائنٹکس یہ ہے:

```
main( )
      {
            Program statement..........
      }
```


NOT FOR SALE

main( ) فنکشن کی باڈی (Body of main( ) Function)

پروگرام کی اسٹیٹمنٹس main( ) فنکشن کے اندر لکھی جاتی ہیں، main فنکشن کی باڈی { } بریکٹ پر مشتمل ہوتی ہے۔ یہ اسٹیٹمنٹس پروگرام کی باڈی کہلاتی ہیں۔ جب فنکشن چلایا جاتا ہے تو اِس میں موجود اسٹیٹمنٹس اپنا کام انجام دیتی ہیں۔

C اسٹیٹمنٹس (C Statements)

C اسٹیٹمنٹ ایک ایسی عبارت ہے جو حقیقت میں کچھ کر کے دکھاتی ہے۔ اصل میں کمپیوٹر کو کوئی کام کرنے کی ہدایت ہے۔ مثال کے طور پر; *("Hello World")* ایک C اسٹیٹمنٹ ہے جو اسکرین پر Hello World لکھا ہوا دکھاتی ہے۔ C کے اسٹیٹمنٹ عام طور پر چھوٹے حروف میں لکھے جاتے ہیں۔ C میں ہر اسٹیٹمنٹ سیمی کالن (Semicolon) کے ساتھ ختم ہوتی ہے۔

2.3.4 C میں تبصرے یا کمنٹس کا استعمال (Comments in C)

ایک کمنٹ پروگرام کے سورس کوڈ کی وضاحت ہے۔ یہ ڈویلپر کو کوڈ کی لاجک کی وضاحت کرنے میں مدد کرتا ہے اور پروگرام کو بہتر بناتا ہے۔ اس کا استعمال کمپیوٹر پروگرام / اسٹیٹمنٹ کے مقصد اور عمل کی ایک مختصر وضاحت فراہم کرنے کے لئے کیا جاتا ہے۔ جو پروگرام کو صارفین کے لئے زیادہ قابل فہم بناتا ہے۔ کمنٹس عملدرآمد نہ کرنے والی لائنیں ہیں اور ترجمے کے دوران کمپائلر ان کو نظر انداز کرتا ہے۔ یہ نوٹ کرنا ضروری ہے کہ کمنٹس لائن صرف پروگرام کے کوڈ میں ظاہر ہوتی ہیں اور پروگرام کے آؤٹ پٹ پر ان کا کوئی اثر نہیں ہوتا ہے۔ C پروگرام میں کہیں بھی کمنٹس ظاہر ہو سکتے ہیں اور پروگرامز کمنٹ لائن میں کچھ بھی لکھنے کے لئے آزاد ہے۔

C میں کمنٹس لکھنے کے دو طریقے ہیں۔

1۔ سنگل لائن کمنٹ: سنگل لائن پر کمنٹ // (دو سلیش) استعمال کر کے شامل کیے جاتے ہیں۔ // (دو سلیش) کے دائیں طرف لکھی ہوئی عبارت کمنٹس کے طور پر سمجھی جاتی ہے۔ یہ انداز کسی پروگرام کی ایک اسٹیٹمنٹ پر کمنٹ لکھنے کے لئے مفید ہے۔ مثال کے طور پر:

```
//this is a comment line
// Sorting process
//Function 1 starts here
```

2۔ ملٹی لائن کمنٹس: کمنٹ کے آغاز میں ملٹی لائن کمنٹ کو */ سے کوڈ میں شامل کیا جاتا ہے۔ کثیر لائن تبصرے ختم کرنے کے لئے /* کا استعمال کیا جاتا ہے۔ جب کوئی کمنٹ کئی لائنوں پر مشتمل ہو تو یہ انداز سب سے زیادہ کارآمد ہے۔ مثال کے طور پر:

```
/*
Program-Name:         Examination Record
Programmer-Name:      M. Khalid
Date:                 6th May
Purpose:              Program to show Examination Record
*/
```

سلیش۔ایسٹیریک */ کمنٹ کمپائلر کو بتاتا ہے کہ اس کے بعد آنے والی ہر چیز کو نظر انداز کر دے جب تک کہ اس کو /* کا نشان دوبارہ نہیں مل جاتا۔

NOT FOR SALE

comment.cpp

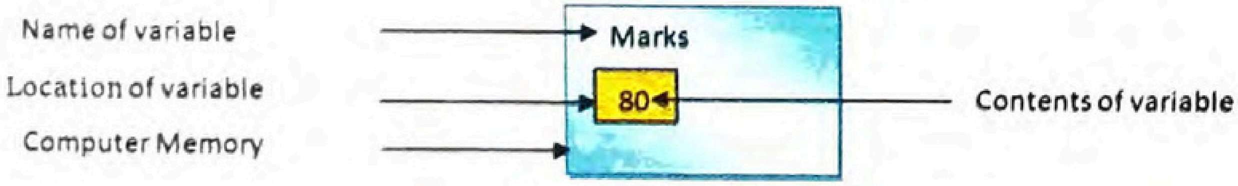
```
#include <iostream>
using namespace std;

int main()
{
    // C++ program to shows how to comment in C++
    // This is the single line comment

    /*  This is
        Multiple Line
        Comment
     */

     cout<<"Single And Multiple line comment will not display ";

    return 0;
}
```

شکل نمبر 2.15 سنگل لائن اور ملٹی لائن کمنٹس

2.4 متغیرات اور مستقلات (Variables and Constants)

2.4.1 متغیر (Variable)

ایک ایسی مقدار جس کی قیمت پروگرام کے عمل کے دوران تبدیل ہو جائے اسے متغیر یا ویری ایبل (Variable) کہا جاتا ہے۔ ایک ویری ایبل کمپیوٹر میموری میں اسٹوریج یا میموری والے مقام کو ظاہر کرتا ہے۔ اسی وجہ سے ویری ایبل میں ذخیرہ شدہ ڈیٹا بھی عارضی ہوتا ہے۔ یہ صرف پروگرام چلنے کے دوران استعمال ہوتا ہے اور اس پر کارروائی کی جاسکتی ہے۔ ویری ایبل میں ذخیرہ شدہ ڈیٹا پروگرام ختم ہونے پر خود بخود ختم ہو جاتا ہے۔



شکل نمبر 2.16 کمپیوٹر میموری میں ویری ایبل

ویری ایبل کی اقسام

ویری ایبل دو قسم کے ہوتے ہیں، عددی اور حرفی (Numeric & Character)۔

عددی (Numeric) ویری ایبل کا استعمال کمپیوٹر پروگراموں میں عددی اقدار کو ظاہر کرنے کے لئے کیا جاتا ہے۔ وہ انٹیجر اور فلوٹنگ پوائنٹ ویلیو کی نمائندگی کرتے ہیں۔ عددی ویری ایبل کی کچھ مثالوں میں marks، sum, n, width وغیرہ ہیں۔

NOT FOR SALE

کریکٹر ویری ایبل کمپیوٹر پروگراموں میں کیریکٹر ویلیو کی نمائندگی کرتے ہیں۔ یہ کسی ایک کیریکٹر یا حروف کے مجموعے کو ظاہر کرتے ہے۔ کیریکٹر ویری ایبل کی کچھ مثالیں Name، Gender، وغیرہ ہیں۔

2.4.2 مستقل (Constant)

کانسٹنٹ ایک مقدار ہے جو پروگرام کے چلنے کے دوران تبدیل نہیں کی جاسکتی ہے۔

حقیقت میں کانسٹنٹ ایک ایسی قیمت ہوتی ہے جو کسی پروگرام میں براہ راست ٹائپ کی جاتی ہے۔ یہ پروگرام میں اس جگہ ظاہر ہوتی ہے جہاں اس کی ضرورت ہو۔ مثال کے طور پر، مندرجہ ذیل اسٹیٹمنٹ میں ایک سٹرنگ کانسٹنٹ موجود ہے:

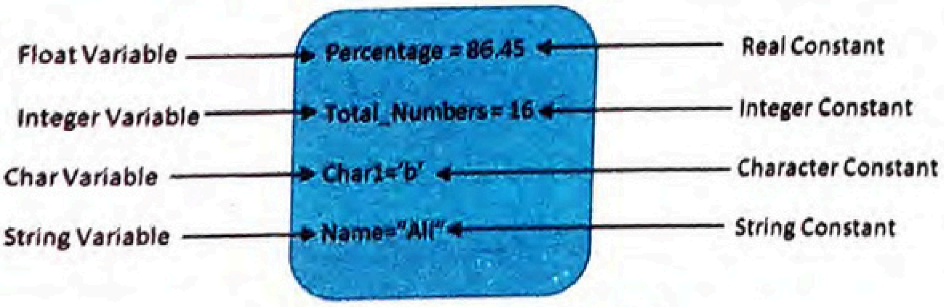
"Pakistan Zinda Bad"

```
printf("Pakistan Zinda Bad");
```

اسی طرح ذیل کی مثال میں بھی:

```
int age= 19;
```

Age ایک عددی ویری ایبل ہے اور 19 وہ کانسٹنٹ قیمت ہے جو Age میں سٹور کی گئی ہے۔



شکل نمبر 2.17 ویری ایبل اور کانسٹنٹ کی اقسام

2.4.3 متغیر کا نام رکھنے کیلئے اُصول (Rules for Writing Variable Names)

C پروگرام میں استعمال ہونے والے کسی بھی متغیر کو درج ذیل اصولوں کے مطابق ہونا چاہئے:

1. انہیں ہمیشہ ایک حرف (Letter) سے شروع ہونا چاہئے، تاہم کچھ سسٹم پہلے کیریکٹر کی حیثیت سے انڈر سکور کی اجازت بھی دیتے ہیں۔
2. ویری ایبل کے نام کی لمبائی C کے کمپائلر پر منحصر ہے۔
3. ویری ایبل کے نام میں خالی جگہ (Blank Space) رکھنے کی اجازت نہیں ہوتی۔
4. ویری ایبل کا نام کوئی ریزرو ورڈ / کی ورڈ نہیں ہونا چاہئے۔
5. اس میں کوئی خاص علامت نہیں ہونا چاہئے جیسے، #، ^ وغیرہ۔

NOT FOR SALE

6. ایک ہی پروگرام میں ایک ویری ایبل کا نام جو ایک ڈیٹا ٹائپ کے لئے استعمال ہو چکا ہے وہ کسی دوسری ڈیٹا ٹائپ کے لئے استعمال نہیں کیا جا سکتا۔

C حروف کے اعتبار سے ایک حساس زبان ہے۔ اس میں چھوٹے اور بڑے حروف میں لکھے گئے ایک جیسے نام مختلف ویری ایبل سمجھے جاتے ہیں۔ مثال کے طور پر، ویری ایبل Pay اور pay دو مختلف ویری ایبل ہیں۔

درست اور غلط ویری ایبل ناموں کی کچھ مثالیں درج ذیل ہیں۔

ویری ایبل کا نام	درست / غلط	وجوہات
Num	درست	
Perform	درست	
Switch	غلط	کلیدی لفظ Keyword
Aysha	درست	
int	غلط	کلیدی لفظ Keyword
3pak	غلط	پہلا حرف عدد ہے۔
x-y	غلط	Underscore کے علاوہ خاص علامات کی اجازت نہیں ہے
Roll No	غلط	نام کے درمیان میں خالی جگہ کی اجازت نہیں۔

2.4.4 C میں استعمال ہونے والی ڈیٹا کے اقسام (Datatypes in C)

C کے تمام کمپائلر میں ویری ایبل کو درج ذیل بنیادی ڈیٹا ٹائپ (Datatypes) تفویض کر سکتے ہیں۔

- انٹیجر (int) Integer
- فلوٹنگ پوائنٹ (Float) اور ڈبل پریسین فلوٹنگ پوائنٹ (double)
- کیریکٹر (char) Character

1. انٹیجر (int, long) ٹائپ (Integer Type)

انٹیجر صحیح اعداد ہوتے ہیں ان کی مقداروں کی حدود مشین پر منحصر ہوتی ہیں۔ پروگرامنگ کی ایک اچھی زبان وہ ہوتی ہے جو پروگرامر کو مختلف اقسام اور حدود (range) والے اعداد کے استعمال کے لیے سہولت فراہم کرے تاکہ اعداد کو انکی درکار میموری کے مطابق کمپیوٹر میں سٹور کیا جا سکے۔ C میں انٹیجر اسٹوریج کے لیے 3 مختلف کلاسیں استعمال ہوتی ہیں۔ جیسے *short*، *int* اور *long*۔ یہ تمام ڈیٹا ٹائپ دو طرح کی ہیں *signed* اور *unsigned*۔

NOT FOR SALE

پروگرام میں انٹیجر کی تعریف کرنے کے لیے درج ذیل اسٹیٹمنٹ لکھی جاسکتی ہے:

int variable;

مندرجہ ذیل جدول میں ان تمام ڈیٹا اقسام کے لیے درکار میموری اور قیمتوں کی حدود لکھی گئی ہیں۔

ویری ایبل کی ڈیٹا قسم	میموری میں درکار بائٹس	قیمت کی حد (Range)
int	2	-32,768 سے +32,767
long	4	-2,147,483,648 سے +2,147,483,647
unsigned int	2	0 سے 65, 535

2. فلوٹنگ پوائنٹ ڈیٹا ٹائپ (float, double, long double)

جاننا ضروری ہے۔

32 بٹ سسٹم میں، انٹیجر ویری ایبل کا سائز 2 بائٹس سے 4 بائٹس تک بڑھایا گیا ہے۔

نقطہ اعشاریہ والے اعداد اصل میں حقیقی اعداد کے اظہار کے لیے استعمال ہوتے ہیں۔

ان اعداد کی نقطہ اشاریہ کے بعد 7 اعداد تک درست قیمت ظاہر ہوتی ہے۔فلوٹنگ پوائنٹ نمبرز کو کی ورڈ float کے ذریعے ظاہر کیا جاتا ہے۔جب فلوٹنگ پوائنٹ نمبر کی درستگی ناکافی ہوتی ہے تو، ہم نمبر کی وضاحت کے لئے double اور long double کا استعمال کرسکتے ہیں۔ پروگرام میں فلوٹنگ پوائنٹ کی تعریف کرنے کے لیے درج ذیل اسٹیٹمنٹ لکھی جاسکتی ہے۔

float variable;

عموماً مندرجہ بالا فلوٹنگ پوائنٹ ڈیٹا ٹائپس کے لئے کمپائلر بائٹس کی تعداد مختص کرتا ہے جو کہ ذیل میں دی گئی ہیں۔

ویری ایبل کی ڈیٹا قسم	میموری میں درکار بائٹس	قیمت کی حد
float	4	3.4×10^{-38} to 3.4×10^{38}
double	8	1.7×10^{-308} to 1.7×10^{308}
long double	10	3.4×10^{-4932} to 1.1×10^{4932}

3. کریکٹر (char) ڈیٹا ٹائپ (Character datatype)

کیریکٹر ویری ایبل میں صرف ایک حرف (Character) ہی سٹور ہو سکتا ہے۔

کیریکٹر ویری ایبل میں ایک حرف کو سٹور کرنے کے لیے C کمپائلر، میموری میں صرف ایک بائٹ مختص کرتا ہے۔

پروگرام میں کیریکٹر ویری ایبل کی تعریف درج ذیل اسٹیٹمنٹ سے کی جاسکتی ہے:

char ch;

NOT FOR SALE

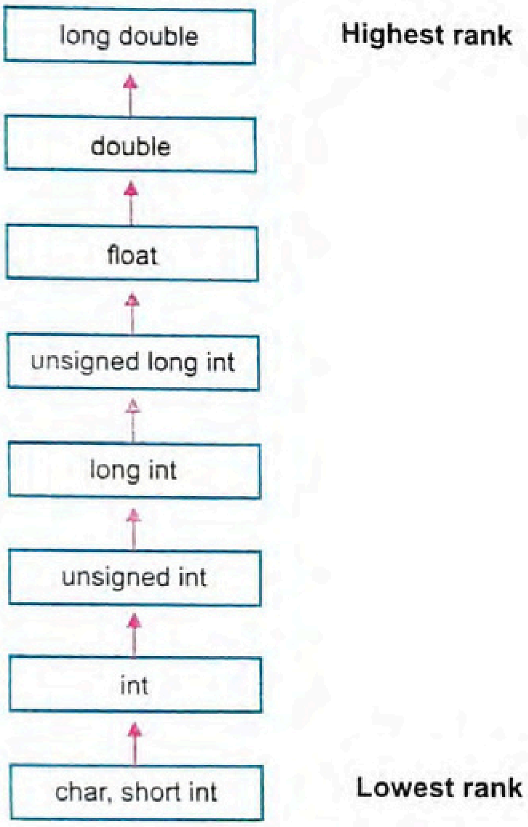
2.4.5 ٹائپ کاسٹنگ (Type Casting)

پروگرام چلنے کے عمل کے دوران کسی ایک ڈیٹا ٹائپ کو کسی دوسری ڈیٹا ٹائپ میں تبدیل کرنے کے عمل کو ٹائپ کاسٹنگ کے نام سے جانا جاتا ہے۔ ٹائپ کاسٹنگ دو طریقوں سے کی جاسکتی ہے۔

- مخفی ٹائپ کاسٹنگ
- ظاہری ٹائپ کاسٹنگ

1. مخفی ٹائپ کاسٹنگ (Implicit Type Casting)

C کمپائلر اس قسم کی ٹائپ کاسٹنگ خود کر لیتا ہے۔ ریاضی کے عمل میں شامل آپرینڈز کو ایک ہی قسم کا ہونا چاہیئے۔ اگر پر آینڈز (operands) کی ڈیٹا ٹائپ مختلف ہیں تو، چھوٹی ڈیٹا ٹائپ والی ویلیو کو بڑی ڈیٹا ٹائپ میں تبدیل کیا جاتا ہے۔



شکل نمبر 2.18 ٹائپ کاسٹنگ کی ترتیب

مثال:

فرض کریں x کوئی انٹیجر ہے اور y ایک long ویری ایبل ہے اور اگر درج ذیل کو ہم حل کرنا چاہیں:

x+y

اگرچہ x کی ڈیٹا ٹائپ y کی ڈیٹا ٹائپ سے چھوٹی ہے اس لیے جمع کے عمل کے دوران x کی قیمت کو بھی long ڈیٹا ٹائپ میں تبدیل کر دیا جائے گا۔

NOT FOR SALE

2. ظاہری ٹائپ کاسٹنگ (Explicit Type Casting)

اس طرح کی ٹائپ کاسٹنگ پروگرامر خود کرتا ہے۔ یہ کاسٹ آپریٹر (Cast operator) کا استعمال کرکے کی جاتی ہے۔ کاسٹ آپریٹر کمپیوٹر کو یہ بتاتا ہے کہ ڈیٹا کو ایک قسم سے دوسری قسم میں تبدیل کرنا ہے۔

کاسٹ آپریٹر کا سانٹکس یہ ہے:

(type) expression;

(type): یہ ڈیٹا کی قسم کو ظاہر کرتا ہے جس میں اوپیرینڈ (operand) کو تبدیل کرنا ہے۔

expression: یہ کانسٹنٹ، ویری ایبل یا expression کو ظاہر کرتا ہے جس کی ڈیٹا ٹائپ کو تبدیل کرنا ہے۔

مثال:

فرض کیا x اور y دو float قسم کے ویری ایبل ہیں۔ x کی قیمت 10.3 ہے اور y کی قیمت 5.2 ہے اور ہم نے ذیل کی عبارت کو حل کرنا ہے:

x%y

مندرجہ بالا عمل پروگرام میں ایک غلطی (error) پیدا کر دے گا کیونکہ x اور y کی ڈیٹا قسم float ہے اور موڈولس (Modulus) پر آپریٹر % کسی float ویری ایبل کے ساتھ کام نہیں کر سکتا۔ یہ صرف انٹیجر پر ہی عمل کرتا ہے۔ اس لیے مندرجہ بالا عبارت کو اس طرح لکھا جاسکتا ہے تاکہ غلطی پیدا نہ ہو۔

(int) x% (int) y

اس عبارت میں x اور y کی قیمتیں پہلے انٹیجر میں تبدیل کر دی جائیں گی اور پھر ان کو حل کیا جائے گا۔ x کی قیمت کو 10 میں اور y کی قیمت کو 5 میں تبدیل کر دیا جائے گا۔ اس عبارت کا نتیجہ 0 (صفر) ہو گا۔

2.4.6 const کا استعمال (Constant Qualifier)

const کو پروگرام کے اندر کانسٹنٹ (Constant) کی تعریف کرنے کے لیے استعمال کیا جاتا ہے۔ کانسٹنٹ کی تعریف کرنے کے لیے اس کا نام اور اس کی ڈیٹا ٹائپ لکھنا لازمی ہے۔ کانسٹنٹ کو ابتدائی قیمت دینا ضروری ہے۔ ابتدائی قیمت پروگرام چلنے کے دوران تبدیل نہیں کی جاسکتی۔

سانٹکس:

const کی مدد سے کانسٹنٹ بنانے کو ذیل میں دکھایا گیا ہے:

const data type identifier = value;

جبکہ؛

const	ایک کی ورڈ ہے جس سے یہ بتایا جاتا ہے کہ کانسٹنٹ بنایا جا رہا ہے۔
data type	بتاتا ہے کہ کانسٹنٹ کی ڈیٹا ٹائپ کونسی ہے۔
identifier	جو کانسٹنٹ بنایا جا رہا ہے یہ اس کا نام ہے۔
value	یہ کانسٹنٹ کی ابتدائی قیمت کو ظاہر کرتا ہے۔

مثالیں:

```
const int N = 100;
const float R = 3.5;
```

مذکورہ بالا مثالوں میں ایک انٹیجر کانسٹنٹ N ہے جس کی ابتدائی قیمت 100 اور ایک فلوٹ کانسٹنٹ R ہے جس کی ابتدائی قیمت 3.5 ہے۔ پروگرام کے چلنے کے دوران ان قیمتوں کو تبدیل نہیں کیا جا سکتا۔

یہی کام define# پری پروسیسر کا استعمال کرتے ہوئے بھی کیا جا سکتا ہے:

```
#define N 100
#define R 3.5
```

2.4.7 ویری ایبل بنانا (Declaration) اور ابتدائی قیمت رکھنا (Initialization)

ویری ایبل بنانا (Declartion of Variables)

ویری ایبل کا نام رکھنا اور اس کی ڈیٹا ٹائپ متعین کرنے کو ویری ایبل بنانا یا ویری ایبل ڈکلریشن کہا جاتا ہے۔ تمام ویری ایبل جو پروگرام میں استعمال ہوں گے انہیں ایک اسٹیٹمنٹ کے ذریعے بنایا جاتا ہے جسے ویری ایبل ڈیکلریشن اسٹیٹمنٹ کہا جاتا ہے۔ ویری ایبل بنانے کے لیے ذیل کا سائنٹکس استعمال ہوتا ہے:

```
Datatype List of variables;
```

جبکہ؛

Datatype	ڈیٹا ٹائپ ویری ایبل کی ڈیٹا ٹائپ بتاتا ہے۔ مثال کے طور پر *int*، *float* وغیرہ۔
List of variables	یہ کافی سارے ویری ایبل کی فہرست ہوگی جو ایک دوسرے سے کاما (Comma) کے ذریعے جدا ہوں گے۔

مختلف قسم کے ویری ایبل بنانے کے لیے علیحدہ علیحدہ اسٹیٹمنٹ استعمال کی جائیں گی۔ مثال کے طور پر انٹیجر ویری ایبل abc، xyz، d اور s بنانے کے لیے درج ذیل اسٹیٹمنٹ استعمال ہوگی۔

```
int abc, xyz, d, s;
```


NOT FOR SALE

انٹیجر ویری ایبل a اور xy بنانے کے لیے، اور فلوٹ ویری ایبل b بنانے کے لیے، کیریکٹر ویری ایبل ch بنانے کے لیے اور doubleویری ایبل sum بنانے کے لیے ذیل میں دی گئی اسٹیٹمنٹ استعمال ہو نگی۔

```
int a, xy;
float b;
char ch;
double sum;
```

ویری ایبل کی ابتدائی قیمت رکھنا (Intialization of Variable)

جب ایک ویری ایبل بنایا جاتا ہے تو میموری میں اس کے لیے جگہ رکھ دی جاتی ہے۔ ویری ایبل کو بناتے وقت کوئی قیمت رکھنے کے عمل کو انیشیلائزیشن (Initialization of Variable) کہا جاتا ہے۔

مثال کے طور پر، انٹیجر ویری ایبل a، b اور c کو 60، 43 اور 77 کی قیمتیں اور n کو 3.35 ویلیو تفویض کر کے فلوٹ ٹائپ کے ویری ایبل بنانے کے عمل کو ذیل کی اسٹیٹمنٹ میں دکھایا گیا ہے :

```
int a = 60, b = 43, c = 77;
float n=3.35;
```

اہم نکات

- ہدایات (instructions) کا ایک مجموعہ جو ایک کمپیوٹر کو بتاتا ہے کہ کیا کرنا ہے اور کیسے کرنا ہے اسے پروگرام کہتے ہیں۔
- سائنٹکس (Syntax) دراصل کسی بھی پروگرامنگ لینگوئج کے وہ قواعد یا اصول ہیں جن کے تحت کوئی بھی پروگرام لکھا جاتا ہے۔
- سیمنٹک (Semantic) ایک پروگرامنگ لینگوئج کی اسٹیٹمنٹ کے مطلب کی وضاحت کرتا ہے۔
- پروگرام لکھنے کے لئے استعمال ہونے والے الفاظ، علامات اور کوڈز کا ایک مجموعہ پروگرامنگ لینگوئج کہلاتا ہے۔
- لو لیول لینگوئج کمپیوٹر ہارڈویئر کے زیادہ قریب ہیں۔
- کمپیوٹر زبان کی وہ قسم جس میں ہم ہدایات کو بائنری شکل میں لکھتے ہیں اسے مشینی زبان (Machine Language) کہا جاتا ہے۔
- اسمبلی لینگوئج بھی لو لیول لینگوئج ہے۔
- ہائی لیول لینگوئج انگریزی الفاظ پر مبنی زبانیں ہیں۔
- انٹرپرٹر ہائی لیول پروگرامنگ لینگوئج کا مترجم ہوتا ہے۔
- کمپائلر سسٹم سافٹ ویئر ہے جو ہائی لیول لینگوئج کے سورس کوڈ کو مشین لینگوج آبجیکٹ کوڈ میں ترجمہ کرتا ہے۔

NOT FOR SALE

- انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) پروگراموں کو تیار کرنے، مرتب کرنے اور چلانے کے لئے پروگرامنگ لینگوئج ڈویلپمنٹ ٹول ہے۔
- مختلف کاموں کو انجام دینے کے لئے ہیڈر فائلیں سٹینڈرڈ لائبریری کے فنکشن کا مجموعہ ہیں۔
- وہ الفاظ جو کسی بھی زبان کے ذریعہ خصوصی مقاصد کے لئے استعمال ہوتے ہیں انہیں Reserved Words کہتے ہیں۔
- پروگرام کے حقیقی آغاز سے پہلے کمپائلر کو جو ہدایات دی جاتی ہیں انھیں پری پروسیسر ڈائریکٹیوز کہتے ہیں۔
- ایک ایسی مقدار جس کی قیمت پروگرام کے عمل کے دوران تبدیل ہو سکتی ہے اسے متغیر/ویری ایبل کہا جاتا ہے۔
- کانسٹنٹ وہ مقدار ہے جسے پروگرام کے چلنے کے دوران تبدیل نہیں کیا جا سکتا۔
- پرواگرم کے چلنے کے دوران کسی ایک ڈیٹا ٹائپ کو دوسرے ڈیٹا ٹائپ میں تبدیل کرنے کے عمل ٹائپ کاسٹنگ کے نام سے جانا جاتا ہے۔

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. ______________ ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتا سکے کہ کیا کرنا اور کیسے کرنا ہے۔

(الف) کمپائلر (ب) پروگرام (ج) کمنٹس (د) ہارڈویئر

ii. لو لیول لینگوئج ______________ کے قریب تر ہیں۔

(الف) انسانی زبان (ب) مشینی زبان (ج) کمپیوٹر ہارڈویئر (د) کمپیوٹر سافٹ ویئر

iii. ذیل میں کونسا C پروگرامنگ کا جزو نہیں ہے۔

(الف) کمپائلر (ب) ایڈیٹر (ج) لنکر (د) کنورٹر

iv. کمپائلر کے آؤٹ پُٹ کو ______________ کہا جاتا ہے۔

(الف) پروگرام (ب) سورس کوڈ (ج) لنکڈ کوڈ (د) آبجیکٹ کوڈ

v. کونسی لینگوئج میں ہدایات (statements) کو بائنری کوڈ میں لکھا جاتا ہے؟

(الف) مشین لینگوئج (ب) اسمبلی لینگوئج

(ج) ہائی لیول لینگوئج (د) ان میں سے کوئی نہیں

vi. کونسی زبان ہائی لیول لینگوئج نہیں ہے؟

(الف) BASIC (ب) اسمبلی لینگوئج (ج) Pascal (د) FORTRAN

NOT FOR SALE

vii. ان میں کون سا کمپائلر ڈائریکٹیو ہے۔

(الف) #include<stdio.h> (ب) Using namespace std;

(ج) int main( ) (د) یہ تمام

viii. سورس کوڈ کو آبجیکٹ کوڈ میں تبدیل کرنے کے عمل کو ______________ کہا جاتا ہے۔

(الف) Compilation (ب) Executing

(ج) Linking (د) Saving

ix. ذیل میں کس کی قیمت پروگرام چلنے کے دوران تبدیل ہو سکتی ہے؟

(الف) کانسٹنٹ (ب) ویری ایبل (ج) نام (د) علامت

x. پروگرام کا آغاز ______________ فنکشن سے ہوتا ہے۔

(الف) const( ) (ب) main( ) (ج) name( ) (د) start( )

سوال نمبر 2۔ ذیل میں دیئے گئے مختصر سوالات کے جوابات تحریر کریں۔

i. پروگرام سائنٹکس اور سیمنٹک میں فرق بیان کریں۔

ii. ہائی لیول لینگوئج اور لو لیول لینگوئج میں فرق بیان کریں۔

iii. IDE کیا ہے؟

iv. OOP کیا ہے؟

v. ہائی لیول لینگوئج کی خصوصیات بیان کریں۔

vi. کمپائلر اور انٹرپریٹر میں فرق بیان کریں۔

vii. ہیڈر فائل کیا ہے؟

viii. سورس پروگرام اور آبجیکٹ پروگرام میں کیا فرق ہے؟

ix. کلیدی الفاظ یا ریزروڈ ورڈز کیا ہوتے ہیں؟

x. ویری ایبل کے نام کے لیے قواعد تحریر کریں۔

xi. *const* کس مقصد کے لیے استعمال ہوتا ہے؟

سوال نمبر 3۔ پروگرامنگ لینگوئج کیا ہے؟ اس کی مختلف اقسام کے بارے میں تفصیل سے لکھیں۔

سوال نمبر 4۔ ہائی لیول لینگوئج کیا ہے؟ کوئی سی پانچ ہائی لیول لینگوئج کے بارے میں تفصیل سے لکھیں۔

سوال نمبر 5۔ C پروگرام کا بنیادی ڈھانچہ کیا ہے؟ پری پروسیسر ڈائریکٹیو کی مختلف اقسام کی تفصیل بیان کریں۔

سوال نمبر 6۔ کانسٹنٹ اور ویری ایبل کیا ہیں؟ ان میں فرق کو مثالوں سے واضح کریں۔

سوال نمبر 7۔ C پروگرامنگ لینگوئج میں استعمال ہونے والی ڈیٹا ٹائپ کو مثالوں سے تفصیلاً بیان کریں۔

سوال نمبر 8۔ Type Casting کیا ہے؟ مثالوں سے واضح کریں۔

سوال نمبر 9۔ C لینگوئج میں ویری ایبل کیسے بنائے جاتے ہیں اور ان کی ابتدائی قیمت کیسے رکھی جاتی ہے؟ مثالیں دیں۔

NOT FOR SALE