کمپیوٹر سائنس

10

پنجاب کریکولم اینڈ ٹیکسٹ بک بورڈ، لاہور

جملہ حقوق بحق پنجاب کریکولم اینڈ ٹیکسٹ بُک بورڈ، لاہور محفوظ ہیں۔
اس کتاب کا کوئی حصہ نقل یا ترجمہ نہیں کیا جا سکتا اور نہ ہی اسے ٹیسٹ پیپر، گائیڈ بکس، خلاصہ جات،
نوٹس یا امدادی کتب کی تیاری میں استعمال کیا جا سکتا ہے۔

فہرست

باب نمبر	عنوانات	صفحہ نمبر
1	پروگرامنگ کا تعارف	2
2	یوزر انٹریکشن (User Interaction)	21
3	مشروط منطق (Conditional Logic)	51
4	ڈیٹا اینڈ ریپیٹیشن (Data and Repetition)	77
5	فنکشنز (Functions)	102
الف	اِصطلاحات	118
ب	انڈیکس (اشاریہ)	121
ج	جوابات	122

مصنّف:
ڈاکٹر عدنان ہاشمی
اسسٹنٹ پروفیسر
شعبہ کمپیوٹر سائنس اینڈ آئی۔ ٹی
دی یونیورسٹی آف لاہور

مدیر:
ڈاکٹر مدثر نصیر
ایسوسی ایٹ پروفیسر
شعبہ کمپیوٹر سائنس اینڈ آئی۔ ٹی
دی یونیورسٹی آف لاہور

نگران:
جہانزیب خان
ایس۔ایس (کمپیوٹر سائنس)
پی۔سی۔ٹی۔بی لاہور

ڈپٹی ڈائریکٹر (گرافکس) / آرٹسٹ:
مسز عائشہ وحید

ڈائریکٹر مسوّدات:
عطا دستگیر

لے آؤٹ / ڈیزائننگ:
علیم الرحمٰن

نظر ثانی:
ڈاکٹر جمیل الرحمٰن
ایس۔ایس۔ایس (اُردو)
پی۔سی۔ٹی۔بی لاہور

کمپوزنگ:
عرفان شاہد
عمیر طارق
محمد ندیم

تاریخ اشاعت	ایڈیشن	طباعت	تعداد اشاعت	قیمت
دسمبر 2020ء	تجرباتی	اول		

فہرست

یونٹ 1: پروگرامنگ کا تعارف 1

1.1 پروگرامنگ انوائرنمنٹ 2

- 1.1.1 انٹیگریٹڈ ڈیویلپمنٹ انوائرنمنٹ (Integrated Development Environment) 3
- 1.1.2 ٹیکسٹ ایڈیٹر (Text Editor) 4
- 1.1.3 کمپائلر (Compiler) 5

1.2 پروگرامنگ کے بنیادی نکات 5

- 1.2.1 مختص کیے گئے الفاظ 6
- 1.2.2 C-پروگرام کی ساخت 6
- 1.2.3 C-پروگرام میں کمنٹس کا سنٹیکس اور مقصد 8

1.3 مستقلات اور متغیرات 9

- 1.3.1 مستقلات 10
- 1.3.2 متغیّرات 11
- 1.3.3 متغیّر کی ڈیٹا ٹائپ 11
- 1.3.4 متغیّرات کا نام 12
- 1.3.5 متغیّر کی ڈکلیریشن (Variable Declaration) 13
- 1.3.6 متغیّر کی انشلائزیشن (Variable Initialization) 14

یونٹ 2: یوزر انٹریکشن 21

2.1 اِن پٹ/آؤٹ پٹ (I/O) فنکشن 23

- 2.1.1 printf( ) 23
- 2.1.2 فارمیٹ سپیسفائر (Formate Specifier) 24
- 2.1.3 scanf( ) 26
- 2.1.4 getch( ) 28
- 2.1.5 سٹیٹمنٹ ٹرمینیٹر (Statement Terminator) 29
- 2.1.6 اسکیپ سیکوئنس (Escape Sequence) 29

2.2 اوپریٹرز 31

- 2.2.1 اسائنمنٹ اوپریٹر (Assignment Operators) 31
- 2.2.2 ارتھمیٹک اوپریٹر (Arithmatic Operators) 32
- 2.2.3 ریلیشنل اوپریٹر (Relational Operators) 37
- 2.2.4 اسائنمنٹ اوپریٹر (=) اور برابر کا اوپریٹر (==) 38
- 2.2.5 منطقی اوپریٹر 39
- 2.2.6 یونری بمقابلہ بائنری اوپریٹرز (Uniry vs Binary Operators) 41

فہرست

2.2.7 اوپریٹرز کی ترجیح 42

یونٹ 3: مشروط منطق 51

3.1 کنٹرول سٹیٹمنٹس (Control Statements) 52

3.2 سلیکشن سٹیٹمنٹس (Selection Statement) 53

- 3.2.1 if سٹیٹمنٹ 59
- 3.2.2 if-else سٹیٹمنٹ 64
- 3.2.3 نیسٹڈ سلیکشن سٹرکچرز (Nested Selection Structures) 64
- 3.2.4 حل شدہ مثالیں 77

یونٹ 4: ڈیٹا اینڈ ریپیٹیشن 78

4.1 ڈیٹا سٹرکچرز 78

- 4.1.1 ارے (Array) 78
- 4.1.2 ارے ڈکلیریشن (Array Declaration) 79
- 4.1.3 ارے انشلائزیشن (Array Initialization) 79
- 4.1.4 ارے ایلمنٹس تک رسائی 80
- 4.1.5 ویری ایبلز کا بطور ارے انڈیکس استعمال 82

4.2 لوپ سٹرکچر 83

- 4.2.1 لوپس کا عام سٹرکچر 83
- 4.2.2 لوپس کا عام سنٹیکس 83
- 4.2.3 نیسٹڈ لوپس (Nested Loops) 87
- 4.2.4 حل شدہ مثالیں 91
- 4.2.5 لوپس اور اریز (Loops and Arays) 93
- 4.2.6 حل شدہ مثالیں 95

یونٹ 5: فنکشنز 101

5.1 فنکشنز 102

- 5.1.1 فنکشنز کی اقسام 102
- 5.1.2 فنکشنز کے فوائد 103
- 5.1.3 فنکشن کا سگنیچر (Signature of Function) 103
- 5.1.4 فنکشن کو ڈیفائن کرنا 104

6۔ اصطلاحات 118

7۔ انڈیکس (اشاریہ) 121

8۔ جوابات 122

باب 1

پروگرامنگ کا تعارف

تدریسی مقاصد:(Students Learning Outcomes)

انٹیگریٹڈ ڈیولپمنٹ انوائرنمنٹ(IDE)کے تصور کی وضاحت

- C- پروگرامنگ انوائرنمنٹ کے ان موڈیولز کی وضاحت:
  - ٹیکسٹ ایڈیٹر (Text Editor)
  - کمپائلر (Compiler)
- کی ورڈ (Keyword) کی شناخت کرنا
- C-پروگرام کی ساخت کی وضاحت جس میں یہ شامل ہیں:
  - انکلیوڈ(include)
  - main() فنکشن
  - main() کی باڈی
- کمینٹس(Comments) کا مقصد اور ان کے سنٹیکس کی وضاحت
- مستقل(Constant)اور متغیر(Variable)میں فرق کی وضاحت
- متغیر(Variable) کے نام رکھنے کے اصولوں کی وضاحت
- C - کی درج ذیل ڈیٹا ٹائپس(data types)اور ان میں موجود بائٹس کی تعداد کا علم ہونا
  - انٹیجر (int(signed)/unsigned)
  - فلوٹنگ پوائنٹ(float)
  - کریکٹر (char)
- متغیرات(Variables) کو واضح(Declare)اور اجرا کرنے (initialize) کے پروسیس کی وضاحت

تعارف (Introduction)

کمپیوٹر ہماری زندگی کا ایک اہم حصہ بن چکے ہیں۔ یہ ہمیں ریاضی کے پیچیدہ سوالات اور انٹرنیٹ پر کھوج لگانے (Searching) سے لے کر سیٹلائٹ اور راکٹ لانچر کو چلانے اور کنٹرول کرنے تک کئی مسائل کے حل میں مدد دیتا ہے۔ دراصل کمپیوٹر خود سوچنے کی صلاحیت نہیں رکھتا۔ تمام امور سرانجام دینے کے لیے انسان کمپیوٹر کو ہدایت دیتے ہیں جن سے اُنھیں معلوم ہوتا ہے کہ ایک خاص قسم کے مسئلے کو کس طرح حل کرنا ہے۔ ہدایت کی اس سیریز یا فہرست کو کمپیوٹر پروگرام یا سافٹ ویئر (Software) کہتے ہیں اور یہ ہدایات کمپیوٹر میں محفوظ کرنے کے عمل کو کمپیوٹر پروگرامنگ کہتے ہیں۔ وہ شخص جو جانتا ہو کہ ایک کمپیوٹر پروگرام کس طرح لکھا جاتا ہے 'پروگرامر' کہلاتا ہے۔

کمپیوٹر انگریزی، اُردو یا دیگر عام زبانیں نہیں سمجھتے جن کے ذریعے سے انسان ایک دوسرے سے بات چیت کرتے ہیں۔ ان کی اپنی خاص زبانیں ہوتی ہیں جنھیں کمپیوٹر کے سائنس دان (Computer Scientists) ڈیزائن کرتے ہیں۔ پروگرامر ان خاص زبانوں میں کمپیوٹر پروگرام لکھتے ہیں جنھیں پروگرامنگ لینگویجز (Programming Languages) کہا جاتا ہے۔ بہت زیادہ استعمال ہونے والی پروگرامنگ لینگویجز میں سے چند ایک JAVA, C++, C, C#, اور Python ہیں۔ اس کتاب میں ہم کمپیوٹر پروگرام لکھنے کے لیے C- لینگویج استعمال کریں گے۔ اس باب میں C- لینگویج کو استعمال کرتے ہوئے کمپیوٹر پروگرامنگ کی کچھ بنیادی چیزوں کا ذکر کیا گیا ہے۔

کیا آپ جانتے تھے؟

C- لینگویج 1949ء اور 1973ء کے درمیان بیل لیبز (Bell Labs) میں ڈینس رِچی (Dennis Ritchie) نے تیار کی تھی۔

1.1 پروگرامنگ انوائرنمنٹ (Programming Environment)

کسی بھی کام کو سرانجام دینے کے لیے ہمیں مناسب ٹولز (Tools) کی ضرورت ہوتی ہے۔ مثلاً باغبانی کرنے کے لیے باغبانی کے آلات اور پینٹنگ کے لیے پینٹ، برش اور کینوس (Canvas) کی ضرورت ہوتی ہے۔ اسی طرح پروگرامنگ کے لیے بھی خاص ٹولز درکار ہوتے ہیں۔ پروگرامنگ کے تمام اہم آلات کو اکٹھا کرنے سے پروگرامنگ انوائرنمنٹ بنتی ہے۔ پروگرام لکھنے سے پہلے پروگرامنگ انوائرنمنٹ تیار کرنا ضروری ہے۔ پروگرام لکھنے اور چلانے کے لیے یہ ہمیں بنیادی پلیٹ فارم فراہم کرتی ہے۔

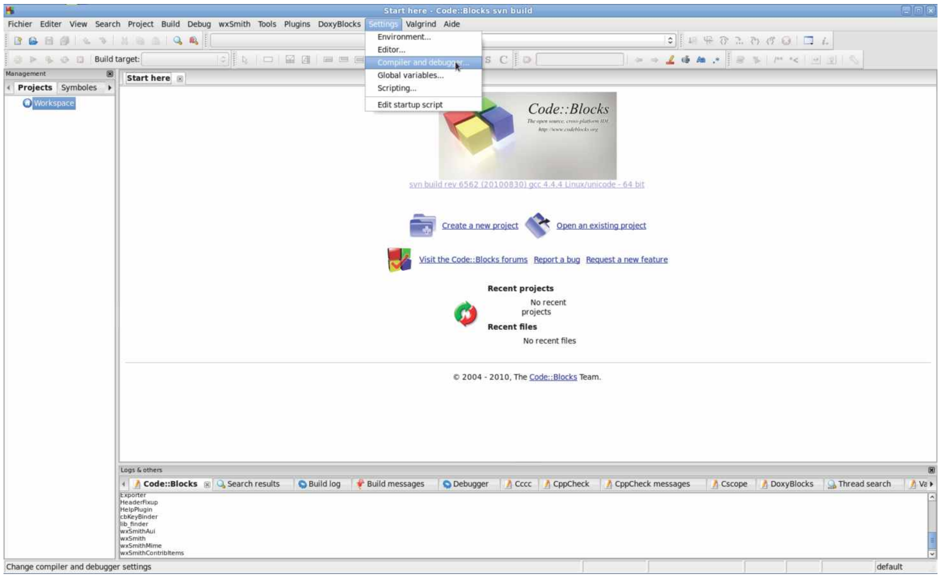
1.1.1 انٹیگریٹڈ ڈیویلپمنٹ انوائرنمنٹ IDE (Integrated Development Environment)

ایک سافٹ ویئر جو پروگرامر کو کمپیوٹر پروگرام لکھنے اور چلانے میں مدد دینے کے لیے پروگرامنگ انوائرنمنٹ فراہم کرتا ہے وہ **انٹیگریٹڈ ڈیویلپمنٹ انوائرنمنٹ (IDE)** کہلاتا ہے۔

IDE کا ایک گرافکل یوزر انٹرفیس (Graphical User Interface) ہوتا ہے جس کی ونڈوز (windows) اور بٹن استعمال کر کے صارف (user) ان پٹ (input) دے سکتا ہے اور آؤٹ پٹ لے سکتا ہے۔ IDE میں ایسے آلات ہوتے ہیں جو پروگرامر کو کمپیوٹر پروگرام لکھنے، چلانے اور ٹیسٹ کرنے کے مراحل میں مدد دیتے ہیں۔ ایک ہی انٹرفیس میں ٹیکسٹ ایڈیٹرز (Text Editors)، کمپائلرز اور ڈیبگرز (Debuggers) کو اکٹھا کر کے یہ مقاصد حاصل کیے جاتے ہیں۔ C- پروگرامنگ لینگوئج کے لیے موجود کچھ IDEs یہ ہیں:

(1	Visual Studio	(2	X Code
(3	Code:: Blocks	(4	Dev C++

شکل **1.1** میں Code:: Blocks کی main سکرین دکھائی گئی ہے۔



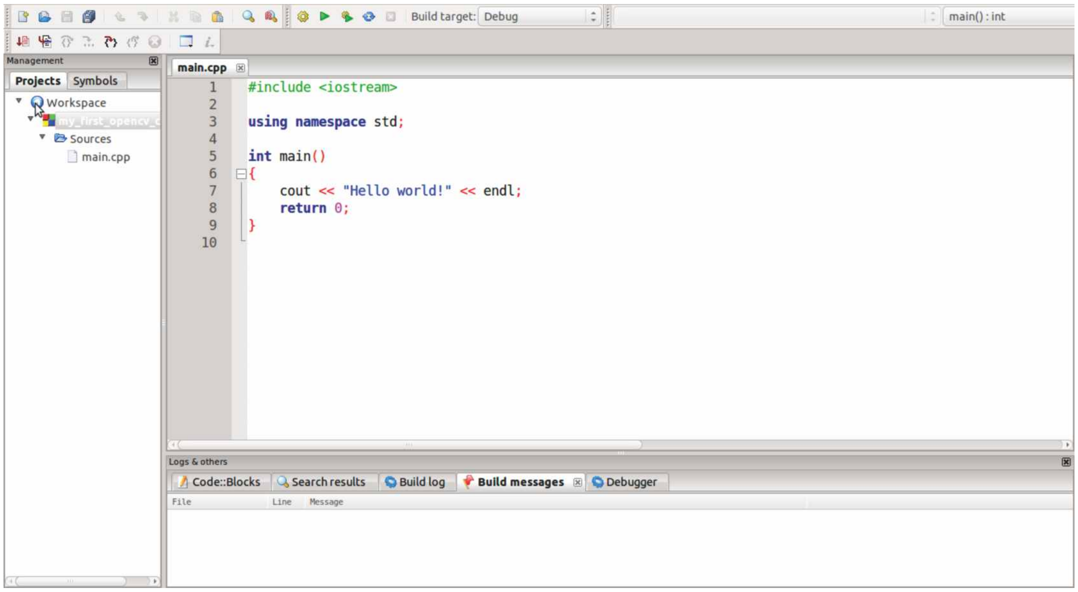
شکل 1.1 Code:: Blocks کا main انٹرفیس۔

سرگرمی 1.1:

اپنے ویب براؤزر (Web Browser) کے ذریعے C- پروگرامنگ لینگوئج کی تین مختلف IDEs کے نام معلوم کریں۔

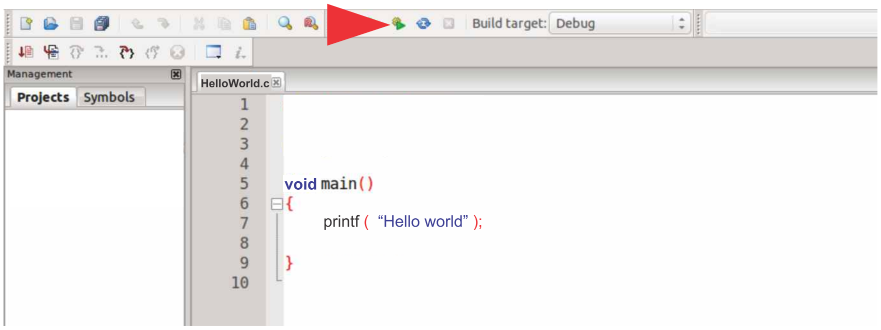
1.1.2 ٹیکسٹ ایڈیٹر (Text Editor)

ٹیکسٹ ایڈیٹر ایسا سافٹ ویئر ہے جو پروگرامر کو کمپیوٹر پروگرام لکھنے اور اس میں ترمیم کرنے میں مدد دیتا ہے۔ تمام IDEs کے اپنے ٹیکسٹ ایڈیٹرز ہوتے ہیں۔ ہم IDE کی مین سکرین پر اپنے پروگرام لکھ سکتے ہیں۔



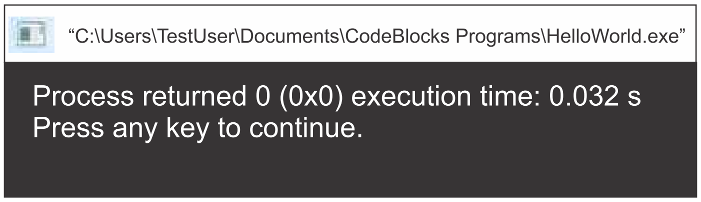
شکل 1.2 Code:: Blocks کا ٹیکسٹ ایڈیٹر

شکل 1.2 میں Code:: Blocks IDE کے ٹیکسٹ ایڈیٹر میں لکھا ہوا C-لینگویج کا ایک بنیادی پروگرام دکھایا گیا ہے۔ جب یہ پروگرام ایگزیکیوٹ ہوتا ہے تو کمپیوٹر سکرین پر "Hello World" ظاہر ہوتا ہے۔ اسے ایگزیکیوٹ کرنے سے پہلے فائل کو محفوظ کرنا ضروری ہے۔ ہم نے اپنے پروگرام کی فائل کا نام "Hello World.c" رکھا ہے۔ پروگرام کی آؤٹ پٹ دیکھنے کے لیے ہم "build and run" بٹن کو کلک کر سکتے ہیں جس کی شکل 1.3 میں نشان دہی کی گئی ہے۔



شکل 1.3 Code:: Blocks میں ایگزیکیوٹ ہوتا ہوا پروگرام

کنسول آؤٹ پٹ سکرین (Console output screen) اس طرح نظر آتی ہے جیسے **شکل 1.4** میں دکھائی گئی ہے۔



شکل 1.4 پروگرام آؤٹ پٹ

سرگرمی 1.2:

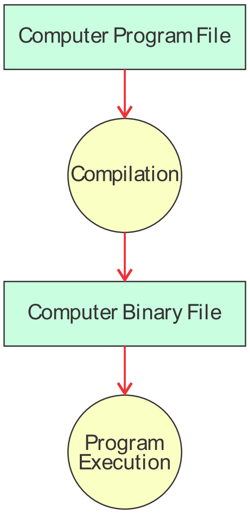
اپنی لیب کے کمپیوٹر میں انسٹال کی ہوئی IDE کھولیں۔ شکل 1.2 میں دیا ہوا پروگرام اپنی IDE کے ٹیکسٹ ایڈیٹر میں لکھیں اور اسے ایگزیکیوٹ کریں۔

1.1.3 کمپائلر (Compiler)

کمپیوٹر صرف مشین کی زبان جو کہ 0 اور 1 پر مبنی ہے کو سمجھتا اور اس میں کام کرتا ہے۔

ایک پروگرام کو ایگزیکیوٹ کرنے کے لیے اِسے مشین کی زبان میں تبدیل کرنا ضروری ہوتا ہے۔

کمپائلر ایسا پروگرام ہے جو کہ ایک اعلیٰ درجے کی زبان میں لکھے گئے پروگرام کو مشین کی زبان یا کوڈ میں تبدیل کرنے کا ذمہ دار ہوتا ہے۔



شکل 1.5: پروگرام ایگزیکیوشن

1.2 پروگرامنگ کے بنیادی نِکات (Programming Basics)

ہر پروگرامنگ لینگوئج کے کچھ ابتدائی تعمیراتی عناصر ہوتے ہیں اور اس میں درست پروگرام لکھنے کے لیے کچھ اصول دیے گئے ہوتے ہیں۔ اصولوں کے اس مجموعے کو لینگوئج کا سینٹیکس (Syntax) کہتے ہیں۔ سینٹیکس "پروگرامنگ لینگوئج" کی گرامر کی طرح ہے۔ پروگرامنگ کرتے ہوئے اگر پروگرامنگ لینگوئج کا سینٹیکس یا اصول درست طرح استعمال نہ کیے جائیں تو پروگرام کمپائل (Compile) نہیں ہوتا۔ اس صورت میں ایک ایرر (error) آتا ہے جسے سینٹیکس ایرر کہا جاتا ہے۔

1.2.1 مختص کیے گئے الفاظ (Reserved words)

ہر پروگرامنگ لینگوئج میں پہلے سے واضح الفاظ کی ایک فہرست ہوتی ہے۔ ہر لفظ کا ایک خاص مطلب ہوتا ہے جو کمپائلر کو پہلے سے معلوم ہوتا ہے۔ ان الفاظ کو مخصوص کیے گئے الفاظ **(Reserved words)** یا کی۔ورڈز **(Keywords)** کہا جاتا ہے۔ اگر پروگرامر ان کی خود کوئی تعریف (Definition) دے تو سینٹیکس ایرر آتا ہے۔

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

ٹیبل 1.1 میں پروگرامنگ لینگوئج کے ریزرو ورڈز کی فہرست دی گئی ہے۔

سرگرمی 1.3:

ان الفاظ میں سے C-لینگوئج کے ریزرو ورڈز یا مختص کیے گئے الفاظ کے گرد دائرہ لگائیں۔

int, pack, create, case, return, small, math, struct, program, library.

1.2.2 C-پروگرام کی ساخت (Structure of Program)

ہم شکل 1.2 میں لکھے ہوئے پروگرام کو دیکھ کر C-پروگرام کی ساخت کو سمجھ سکتے ہیں۔ ہم دیکھ سکتے ہیں کہ ایک پروگرام کو تین حصوں میں تقسیم کیا جاسکتا ہے۔

1-لنک سیکشن یا ہیڈر سیکشن (Link section or Header Section)

C-لینگوئج میں پروگرام لکھتے ہوئے ہم لینگوئج میں پہلے سے ڈیفائن کیے گئے فنکشنز کا بہت زیادہ استعمال کرتے ہیں لیکن ان فنکشنز کو استعمال کرنے کے لیے ہمیں ان فائلوں کو شامل کرنا پڑتا ہے جن میں ان فنکشنز کو بیان کیا گیا ہے۔ یہ فائلیں ہیڈر فائلز (Header files) کہلاتی ہیں۔ ہم پروگرام کے شروع میں include statements لکھ کر اپنے پروگرام میں یہ ہیڈر فائلز شامل کرتے ہیں۔

include statement سٹیٹمنٹ کی ساخت درج ذیل ہے:

#include <header-file-name>

یہاں header-file-name کسی بھی ہیڈر فائل کا نام ہوسکتا ہے۔

درج بالا مثال (**شکل 1.2**) میں ہم نے فائل <stdio.h> شامل کی ہے جس میں ان پٹ اور آؤٹ پٹ فنکشنز سے متعلق معلومات ہیں۔ دیگر بہت سی ہیڈر فائلز ہوتی ہیں مثلاً فائل <math.h> میں ریاضی کے سارے پہلے سے طے شدہ (Pre-defined) فنکشنز موجود ہیں۔

2- main سیکشن:

اس میں main() فنکشن ہوتا ہے۔ ہر C پروگرام میں ایک main() فنکشن ضرور ہوتا ہے اور یہیں سے پروگرام چلنا شروع ہوتا ہے۔

3- main() فنکشن کی باڈی:

main() کی باڈی ٹیڑھی بریکٹوں (curly braces) {} کے اندر ہوتی ہے۔ تمام سٹیٹمنٹس جو ٹیڑھی بریکٹوں {} کے اندر آتی ہیں وہ main فنکشن کی باڈی میں شامل ہوتی ہیں۔ اوپر دیے گئے پروگرام میں سٹیٹمنٹ printf("Hello World!"); کمپیوٹر سکرین پر "Hello World" پرنٹ کرنے کے لیے پری ڈیفائنڈ فنکشن Printf کا استعمال کرتی ہے۔ ہم اپنے پروگرام میں اور فنکشنز بنا کر انھیں بھی main() فنکشن میں استعمال کر سکتے ہیں۔

اہم نوٹ:

C- لینگوئج کے سینٹیکس کے اعتبار سے درست پروگرام لکھنے کے لئے درج ذیل نکات کو ذہن میں رکھنا ضروری ہے۔

- C- لینگوئج پروگرام میں سٹیٹمنٹس کی ترتیب وہی ہونی چاہیے جس ترتیب سے ہم عبارات (Statements) کو چلانا چاہتے ہیں۔
- C- لینگوئج کیس سینسٹو (case sensitive) ہے۔ یعنی اگر کوئی ریزرو ورڈ چھوٹے حروف میں ڈیفائن کیا گیا ہے تو اسے بڑے حروف میں نہیں لکھا جا سکتا مثلاً int اور Int مختلف ہیں۔ پہلا ریزرو ورڈ ہے جب کہ دوسرا نہیں ہے۔
- ہر سٹیٹمنٹ کے آخر پر سیمی کولن (;) آتا ہے۔

سرگرمی 1.4:

C- پروگرام کے مختلف حصوں کی شناخت کریں۔

```
#include<stdio.h>
#include <conio.h>
void main()
{
        printf("I am a student of class 10");
        getch();
}
```

1.2.3 C- پروگرام میں کمینٹس کا سینٹیکس اور مقصد

کمینٹس پروگرام کی وہ سٹیٹمنٹس ہوتی ہیں جنھیں کمپائلر نظرانداز کر دیتا ہے اور یہ ایگزیکیوٹ نہیں ہوتیں۔ کمینٹس اپنے کوڈ کی وضاحت کرنے کے لیے ہوتے ہیں اور عموماً فطری زبان جیسا کہ انگریزی میں لکھے جاتے ہیں۔

کمینٹس لکھنے کا مقصد:

کمینٹس کو پروگرام کی دستاویز کی طرح سمجھا جا سکتا ہے۔ ان کے دو مقاصد ہوتے ہیں:

1- یہ دوسرے پروگرامرز کے کوڈ سمجھنے میں مدد دیتے ہیں۔

2- ان کے ذریعے سے ہم اپنے کوڈ کو لکھنے کے کئی سال بعد بھی آسانی سے سمجھ سکتے ہیں۔

ہمیں ان سٹیٹمنٹس کو ایگزیکیوٹ کرنے کی ضرورت نہیں ہوتی اگر ہم ان سٹیٹمنٹس کو ایگزیکیوٹ کریں تو سینٹیکس ایررز آ سکتے ہیں کیوں کہ یہ سٹیٹمنٹس فطری زبان میں لکھی جاتی ہیں۔

کمنٹس لکھنے کا سینٹیکس:

C- پروگرامنگ لینگویج میں کمینٹس کی دو قسمیں ہوتی ہیں:

1- سنگل لائن کمینٹس (Single Line Comments)

2- ملٹی لائن کمینٹس (Multi Line Comments)

سنگل لائن کمینٹس // سے شروع ہوتے ہیں۔ اس لائن پر // کے بعد کچھ بھی لکھا جائے وہ کمینٹ ہی سمجھا جاتا ہے۔ مثلاً // This is a comment

ملٹی لائن کمینٹس /* سے شروع ہوتے ہیں اور */ پر ختم ہوتے ہیں۔

/* and */ کے درمیان جو بھی لکھا جائے وہ کمینٹ ہی سمجھا جاتا ہے۔ چاہے ایک سے زیادہ سطروں پر ہو۔ مثلاً

```
/*this is
a multi-line
comment*/
```

درج ذیل کوڈ میں کمینٹس کا استعمال دکھایا گیا ہے۔

EXAMPLE CODE 1.1

```
#include <stdio.h>
/*this program displays "I am a student of class 10" on the
output screen*/
void main()
{ //body of main function starts from here
    printf("I am a student of class 10");
} //body of main function ends here
```

سرگرمی 1.5:

درج ذیل میں سے درست کمینٹس پر (✓) کا نشان لگائیں۔

- *comment goes here*
- /comment goes here/
- %comment goes here%
- /* comment goes
  here*/
- /*comment goes here/
- //comment goes here */

1.3 مستقلات اور متغیّرات:

ہر زبان کے حروف کا ایک بنیادی سیٹ (کریکٹر سیٹ) ہوتا ہے جنھیں قابل اطلاق طریقے سے جوڑ کر الفاظ بنائے جاتے ہیں اور ان الفاظ کو جملے بنانے میں استعمال کیا جاتا ہے۔ اسی طرح C- پروگرامنگ لینگوئج میں بھی کریکٹر سیٹ ہوتا ہے جس میں یہ سب شامل ہوتے ہیں:

1) حروف (A,B,...,Y,Z), (a,b,....y,z)

2) ہندسے (0-9)

3) خصوصی علامات (~`!@#%^&*()_-+=\{}[]:;"'<>,.?/)

ان حروف، ہندسوں اور خصوصی علامات کو جب قابلِ اطلاق طریقے سے جوڑا جائے تو مستقلات، متغیرات اور کی۔ورڈز (جنھیں ریزرو ورڈز بھی کہتے ہیں) بنتے ہیں۔ہم پہلے ہی کی-ورڈ کے بارے میں پڑھ چکے ہیں۔اب ہم مستقلات اور متغیرات کے بارے میں پڑھیں گے۔

1.3.1 مستقلات(Constants):

مستقلات وہ قیمتیں ہوتی ہیں جنھیں پروگرام تبدیل نہیں کرسکتا۔مثلاً 5, 75.7, 1500وغیرہ۔C-پروگرامنگ لینگوئج میں ابتدائی طور پر مستقلات کی تین قسمیں ہیں:

1) انٹیجر مستقلات (Integer constants):

ان کی قیمتوں میں اعشاریہ نہیں آتا مثلاً 7, 1256, 30100, 55555, -54, 2349 وغیرہ۔یہ مثبت بھی ہوسکتی ہیں اور منفی بھی۔اگر قیمت سے پہلے کوئی علامت نہ ہو تو قیمت کو مثبت ہی سمجھا جاتا ہے۔

2) حقیقی مستقلات (Real constants):

ان کی قیمتوں میں اعشاریہ آتا ہے مثلاً:3.14, 15.3333, 75.0, -1575.76, 7941.2345وغیرہ۔یہ بھی مثبت اور منفی ہوسکتی ہیں۔

3) کریکٹر مستقلات (Character Constants):

' ' کے اندر لکھا گیا کوئی بھی چھوٹے یا بڑے حروف، پنکچوایشن (Punctuation) کی علامت یا سپیشل ورڈ مستقل کریکٹر ہوتی ہے مثلاً '5', '7', 'a', 'x', '!', ';'وغیرہ۔

اہم نکتہ:

مستقل کریکٹر کے طور پر لکھا گیا ہندسہ جیسے '9'، مستقل انٹیجر کریکٹر کے طور پر لکھے گئے ہندسے جیسے 9 سے مختلف ہوتا ہے۔ہم دو انٹیجر مستقل کو جمع کرکے ریاضی کے اعتبار سے صحیح جواب حاصل کرسکتے مثلاً 9+8=17 لیکن کریکٹر مستقلات کو جمع کرکے نہیں مثلاً '9'+'8'≠17

سرگرمی 1.6:

مندرجہ ذیل مستقلات کی اقسام کی شناخت کریں۔

32.768	-21	'*'	1.2	12
'\'	40.0	41	-12.3	'a'

1.3.2 متغیرات (Variables)

متغیر دراصل میموری لوکیشن (Memory Location) کو دیا جانے والا نام ہے۔جیسا کہ ڈیٹا کو کمپیوٹر کی میموری میں محفوظ کیا جاتا ہے۔متغیر کی قیمت پروگرام میں بدلی جا سکتی ہے۔اس سے مراد یہ ہے کہ اگر ایک متغیر کی قیمت 5 ہے تو ہم بعد میں اس کی جگہ کچھ اور رکھ سکتے ہیں۔

ہر متغیر کا ایک منفرد نام ہوتا ہے جسے شناخت کنندہ (Identifier) کہتے ہیں اور ایک ڈیٹا ٹائپ (Data Type) ہوتی ہے۔ ڈیٹا ٹائپ بتاتی ہے کہ متغیر میں کس قسم کا ڈیٹا محفوظ کیا جا سکتا ہے۔C- لینگوئج میں مختلف ڈیٹا ٹائپس ہیں جیسے char,float,int بالترتیب انٹیجر، حقیقی اور کریکٹر ڈیٹا کو محفوظ کرنے کے لیے استعمال ہوتی ہیں۔ٹیبل 1.2 میں ڈیٹا کی مختلف اقسام کے مطابق C لینگوئج کی ڈیٹا ٹائپس بتائی گئی ہیں۔

Type of Data	Matching Data Type in C language	Sample Values
Integer	*int*	123
Real	*float*	23.5
Character	*char*	'a'

ٹیبل 1.2: مختلف قسم کے مستقل اعداد کی ڈیٹا ٹائپس

اب ہم ممکنہ ڈیٹا ٹائپس اور متغیّرات کے ناموں کے بارے میں تفصیل سے پڑھیں گے۔

1.3.3 متغیر کی ڈیٹا ٹائپ (Data Type of a Variable)

C- لینگوئج میں ہر متغیر کی ایک ڈیٹا ٹائپ ہوتی ہے۔ڈیٹا ٹائپ نہ صرف یہ بتاتی ہے کہ اس متغیر میں کس قسم کا ڈیٹا محفوظ کیا جا سکتا ہے بلکہ یہ بھی بتاتی ہے کہ اس متغیر کا ڈیٹا محفوظ کرنے کے لیے کمپائلر کو کتنی بائٹس رکھنی چاہیں۔ نیچے C- لینگوئج کی مختلف ڈیٹا ٹائپس دی گئی ہیں۔

کیا آپ جانتے تھے؟

کچھ کمپائلر انٹیجر کو محفوظ کرنے کے لیے 2 بائٹ استعمال کرتے ہیں۔ ایسے کمپائلرز میں انٹیجر کی حد 32,768- سے 32،768 ہوتی ہے۔

انٹیجر int۔ (Signed/Unsigned)

انٹیجر ڈیٹا ٹائپ انٹیجر کانسٹینٹس کو محفوظ کرنے کے لیے ہوتی ہے۔انٹیجر کے لیے میموری کی 4 بائٹس درکار ہوتی ہیں۔انٹیجر ٹائپ کا متغیر ڈکلیئر کرنے کے لیے ہم کی۔ورڈ intاستعمال کرتے ہیں۔

: Signed int

Signed int میں -2,147,483,648 سے 2147,483,647 تک تمام مثبت اور منفی قیمتیں محفوظ کی جاسکتی ہیں۔اگر صرف int ٹائپ دی جائے تو یہ طے شدہ ہے کہ اسے Signed int ہی سمجھا جائے گا۔

:Unsigned int

Unsigned int میں صرف مثبت قیمتیں ہی آسکتی ہیں جو 0 سے 4,294,967,295 تک ہوسکتی ہیں۔ان سائنڈ انٹیجر ڈکلیئر کرنے کے لیے کی۔ورڈ unsigned int استعمال ہوتا ہے۔

فلوٹنگ پوائنٹ float:

فلوٹ ڈیٹا ٹائپ میں حقیقی نمبر آتے ہیں جن میں اعشاریہ کے بعد زیادہ سے زیادہ چھے ہندسے آسکتے ہیں۔فلوٹ ٹائپ کا متغیر ڈکلیئر کرنے کے لیے کی۔ورڈ float استعمال ہوتا ہے۔ ایک فلوٹ میموری کی 4 بائٹس لیتا ہے۔ اس کی قیمتیں $3.4x10^{-38}$ سے $3.4x10^{38}$ تک ہوسکتی ہیں۔

کریکٹر(char):

C-میں کریکٹر ٹائپ کے ویری ایبلز کو ڈکلیئر کرنے کے لیے کی۔ورڈ char کی ضرورت ہوتی ہے۔اسے محفوظ کرنے کے لیے میموری کی 1 بائٹ درکار ہوتی ہے۔char ٹائپ کے ایک ویری ایبل میں صرف ایک کریکٹر محفوظ کیا جاسکتا ہے۔

1.3.4 متغیرات کا نام (Name of a variable)

ہر متغیر کا ایک مخصوص نام یا شناخت کنندہ ہوتا ہے۔متغیر کا نام رکھنے کے اصول درج ذیل ہیں:

1) متغیر کے نام میں صرف حروف (چھوٹے یا بڑے) ہندسے اور "_" علامت آسکتی ہے۔

2) متغیر کا نام کسی حرف یا "_" علامت سے شروع ہوسکتا ہے ہندسے سے نہیں۔

3) ایک کی۔ورڈ متغیر کا نام نہیں ہوسکتا۔

4) متغیر کے نام کی لمبائی کا کوئی اصول نہیں ہے لیکن بہتر یہ ہے کہ نام مختصر ہو۔

متغیرات کے درست ناموں کی کچھ مثالیں ہیں height ، Average weight ، _var1

سرگرمی 1.7:

درج ذیل میں سے متغیرات کے درست ناموں کے گرد دائرہ لگائیں۔

_Hello,	1var	roll_num	Air23Blue	float
Case	$car	name	=color	Float

اہم نوٹ:

پروگرامنگ کا عمدہ طریقہ یہ ہے کہ متغیرات کے نام ایسے ہونے چاہییں جیسا ڈیٹا اُن میں رکھنا ہو جیسا کہ اگر کسی متغیر میں تنخواہ محفوظ کرنی ہو اس کا نام salary یا wages ہونا چاہیے۔

1.3.5 متغیر کی ڈکلیریشن (Variable Declaration)

پروگرام میں ایک متغیر کو استعمال کرنے سے پہلے اسے ڈکلیئر کرنا ضروری ہے۔ متغیر کو ڈکلیر کرتے ہوئے اس کی ڈیٹا ٹائپ اور نام بتایا جاتا ہے۔

متغیر کو ڈکلیر کرنے کا سینٹیکس درج ذیل ہے:

data_type variable_name;

متغیرات کو ڈکلیر کرنے کی چند مثالیں درج ذیل ہیں:

```
unsigned int age;
float height;
int salary;
char marital_status;
```

ایک سٹیٹمنٹ میں ایک سے زیادہ متغیرات کو بھی ڈکلیئر کیا جاسکتا ہے جیسے درج ذیل مثالوں میں کیا گیا ہے۔

```
unsigned int age, basic_salary, gross_salary;
int points_scored, steps;
```

```
float height, marks;
char marital_status, gender;
```

ڈیٹا ٹائپ بتائے بغیر ایک متغیر ڈکلیئر نہیں کیا جا سکتا۔ ڈکلیئر کرنے کے بعد اس متغیر کی ڈیٹا ٹائپ تبدیل نہیں کی جا سکتی۔ متغیر کو ڈکلیئر کرنے سے پتا چلتا ہے کہ اس کی قسم کیا ہے، اس میں کہاں سے کہاں تک قیمتیں آ سکتی ہیں اور اس پر کس قسم کے اوپریشنز انجام دیے جا سکتے ہیں درج ذیل مثال میں ایک پروگرام دیا گیا ہے، جس میں دو متغیرات ڈکلیئر کیے گئے ہیں۔

EXAMPLE CODE 1.2

```
void main()
{
        char grade;
        int value;
}
```

1.3.6 متغیر کی انشلائزیشن (Variable Initialization)

پہلی مرتبہ ایک متغیر سے قیمت کو منسوب کرنا متغیر کو انیشلائزیشن کہلاتا ہے۔ C- لینگوئج میں متغیر کو ڈکلیئر کرتے ہوئے یا اس کے بعد انیشلائز کیا جا سکتا ہے۔ متغیر کو ڈکلیریشن کے وقت انیشلائز کرنے کے لیے عام سنٹیکس یہ ہے:

```
data_type  variable_name = value;
```

درج ذیل مثال میں ایک پروگرام دیا گیا ہے جس میں دو متغیرات کو ڈکلیئر اور انیشلائز کیا گیا ہے۔

EXAMPLE CODE 1.3

```
#include<stdio.h>
void main()
{
        char grade; //Variable grade is declared
        int  value  =  25;  /*Variable  value  is  declared  and
        initialized.*/
        grade = 'A'; //Variable grade is initialized
}
```

سرگرمی 1.8:

ایک پروگرام لکھیں جو مناسب ڈیٹا ٹائپس کے متغیرات میں آپ کا ذاتی ڈیٹا محفوظ کرے۔

- نام کا پہلا حرف
- جنس کا پہلا حرف
- آپ کی عمر
- آپ کے آٹھویں جماعت کے نمبر
- آپ کی قامت

خلاصہ:

- کمپیوٹر کو انسان ہدایات دیتے ہیں جن سے انھیں پتا چلتا ہے کہ ایک خاص مسئلے کو کیسے حل کرنا ہے۔ ہدایات کی اس فہرست کو **کمپیوٹر پروگرام** یا **سافٹ ویئر** کہا جاتا ہے۔

- کمپیوٹر کو ہدایات دینے یا محفوظ کرنے کا عمل کمپیوٹر پروگرامنگ کہلاتا ہے۔ اور وہ شخص جو جانتا ہو کہ ایک **کمپیوٹر پروگرام** کس طرح لکھا جاتا ہے **پروگرامر** کہلاتا ہے۔

- کمپیوٹر پروگرام جن زبانوں میں لکھے جاتے ہیں وہ پروگرامنگ لینگویجز کہلاتی ہیں۔ عام طور پر جانی جانے والی چند پروگرامنگ لینگویجز JAVA, PYTHON,C++, C ہیں۔

- پروگرامنگ کے لیے ضروری آلات کو اگر اکٹھا کر دیا جائے تو پروگرامنگ انوائرنمنٹ بنتی ہے۔ پروگرامنگ انوائرنمنٹ پروگرام لکھنے اور چلانے کے لیے بنیادی پلیٹ فارم فراہم کرتی ہے۔

- ایک سافٹ ویئر جو پروگرامر کو پروگرام لکھنے اور چلانے میں مدد دے **انٹیگریٹیڈ ڈویلپمنٹ انوائرنمنٹ (IDE)** کہلاتا ہے۔

- ایڈیٹر یا ٹیکسٹ ایڈیٹر ایسا سافٹ ویئر ہوتا ہے جس میں پروگرامر پروگرام لکھ سکتا ہے اور اس میں ترمیم کرسکتا ہے۔ تمام IDE's کے اپنے مخصوص ایڈیٹرز ہوتے ہیں۔

- ہر پروگرامنگ لینگویج میں چند ابتدائی تعمیراتی عناصر ہوتے ہیں اور یہ چند گرامر کے اصولوں کی پابند ہوتی ہے جنھیں سنٹیکس کہا جاتا ہے۔

- ہر پروگرامنگ لینگویج میں پہلے سے ڈیفائن کیے ہوئے الفاظ کی ایک فہرست ہوتی ہے۔ اگر ایک پروگرامر انھیں کسی اور کام کے لیے استعمال کرنے کی کوشش کرے یا انھیں خود کوئی ڈیفینیشن دے تو سنٹیکس ایرر آتا ہے۔ یہ الفاظ کی ورڈز **(Keywords)** یا ذخیرہ الفاظ **(Reserved Words)** کہلاتے ہیں۔

- ایک پروگرام کو تین حصوں میں تقسیم کیا جاتا ہے۔ ہیڈر سیکشن وہ حصہ ہے جس میں ہیڈر فائلز شامل کی جاتی ہیں۔ مین سیکشن میں مین فنکشن آتا ہے۔ Main باڈی میں وہ سب کچھ آتا ہے جو { } قوسین میں لکھا گیا ہو۔

- کمنٹس وہ سٹیٹمنٹس ہوتی ہیں جنھیں کمپائلر نظرانداز کردیتا ہے اور وہ چلتی نہیں ہیں۔ پروگرام کے بارے میں اضافی معلومات شامل کرنے کے لیے کمینٹس کو استعمال کیا جاسکتا ہے۔

- کانسٹینسیس کی قیمتیں تبدیل نہیں ہوتی۔ کانسٹینسیس کی تین اقسام انٹیجر کانسٹینس، ریئل کانسٹینس اور کریکٹر کانسٹینس ہیں۔

- ۔ ایک متغیر کو استعمال کرنے سے پہلے ڈکلیئر کرنا ضروری ہے۔متغیر کی ڈکلیریشن کے دوران اس کا نام اور ڈیٹا ٹائپ بتائی جاتی ہے۔
- ۔پہلی مرتبہ ایک متغیر سے قیمت منسوب کرنا،متغیر کی انشلائزیشن کہلاتا ہے۔متغیر کو ڈکلریشن کے وقت یا اس کے بعد انشلائز کیا جاسکتا ہے۔

مشق

سوال نمبر 1: کثیر الانتخابی سوالات:

1) ایک سافٹ ویئر جو پروگرامر کو کمپیوٹر پروگرام لکھنے میں مدد دیتا ہے ________ کہلاتا ہے۔

a) کمپائلر   b) ایڈیٹر   c) IDE   d) ڈیبگر

2) ________ ایک ایسا سافٹ ویئر ہوتا ہے جو پروگرام کی فائلز کو ایسے کوڈ میں تبدیل کر دیتا ہے جسے مشین سمجھ سکے اور چلا سکے۔

a) کمپائلر   b) ایڈیٹر   c) IDE   d) ڈیبگر

3) ہر پروگرامنگ لینگویج میں چند ابتدائی تعمیراتی عناصر ہوتے ہیں اور یہ گرامر کے چند اصولوں کے پابند ہوتے ہیں جنھیں ______ کہا جاتا ہے۔

a) پروگرامنگ رُولز   b) سنٹیکس   c) تعمیراتی عناصر   d) سیمانٹک رُولز

4) ایسے الفاظ کی فہرست جو پہلے سے ڈیفائنڈ ہیں اور جنھیں پروگرامر اپنے متغیرات کے ناموں کے طور پر استعمال نہیں کر سکتا ______ کہلاتے ہیں۔

a) آٹو ورڈز   b) کی۔ورڈز   c) محدود الفاظ   d) پہلے سے ڈیفائن کیے ہوئے الفاظ

5) include سٹیٹمنٹس __________ سیکشن میں لکھی جاتی ہیں۔

a) ہیڈر   b) مین(Main)   c) کیمنٹس   d) پرنٹ

6) ________ کو سورس(source) کوڈ میں پروگرام کے استعمال کیے ہوئے الگورتھم اور طریقہ کار کی مزید وضاحت کرنے کے لیے استعمال کیا جاتا ہے۔

a) پیغامات   b) اشارات   c) کمینٹس   d) وضاحتیں

7) ________ وہ قیمتیں جو پروگرام کے چلتے ہوئے تبدیل نہیں ہوتی۔

a) متغیرات   b) کانسٹینٹس   c) سٹرنگز   d) کمینٹس

8) ایک فلوٹ میموری کی ________ بائٹس استعمال کرتا ہے۔

a) 3   b) 4   c) 5   d) 6

9) ایک متغیر کو انیشلائز کرنے کے لیے ہم ________ اوپریٹر استعمال کرتے ہیں۔

a) -->   b) =   c) @   d) ?

10) ________ کو کانسٹینٹس محفوظ کرنے کے لیے ایک مرتبان سمجھا جا سکتا ہے۔

a) باکس   b) جار   c) متغیر   d) مجموعہ

سوال نمبر 2 : غلط/درست کی نشاندہی کریں۔

1) ایک IDE ٹیکسٹ ایڈیٹرز، کمپائلرز اور ڈیبگرز کو ایک انٹرفیس میں اکٹھا کرتی ہے۔ غلط/درست
2) کمپیوٹر میں پروگرام فائل میں لکھے ہوئے کوڈ کو چلانے کے لیے بائنری لینگویج میں تبدیل کرنا پڑتا ہے۔ غلط/درست
3) C-پروگرامنگ لینگویج میں column ایک ذخیرۂ الفاظ ہے۔ غلط/درست
4) *comment goes here* ایک درست کمنٹ ہے۔ غلط/درست
5) فلوٹ میں چھے ہندسوں تک پریسا ئز رئیل نمبر محفوظ کیا جاسکتا ہے۔ غلط/درست

سوال نمبر 3 : درج ذیل کی تعریف کریں۔

1) IDE 2) کمپائلر 3) کی۔ ورڈز 4) پروگرام کا مین سیکشن 5) Char ڈیٹا ٹائپ

سوال نمبر 4 : درج ذیل سوالات کے مختصر جوابات تحریر کریں۔

1) ہمیں ایک پروگرامنگ انوائرنمنٹ کی ضرورت کیوں ہوتی ہے؟
2) اپنی لیب کے کمپیوٹر میں موجود IDE میں C-پروگرام فائل بنانے کے مراحل لکھیں؟
3) کمپائلر کے مقصد کی وضاحت کریں؟
4) C-پروگرامنگ لینگویج کے پانچ کی۔ ورڈز کی فہرست تحریر کریں؟
5) C-پروگرام کی ساخت کے اہم حصے بتائیں؟
6) پروگرامنگ میں کمینٹس کیوں استعمال کرتے ہیں؟
7) کانسٹینٹس اور متغیرات میں فرق کریں۔
8) متغیرات کے نام رکھنے کے اصول تحریر کریں۔
9) char اور int میں فرق بتائیں۔
10) ہم ایک متغیر کو کس طرح ڈکلیئر اور انیشلا ئز کر سکتے ہیں۔

سوال نمبر 5 : کالم ملائیں۔

A	B	C
1) IDE	a) ایسا کوڈ جو مشین پہ چلایا جا سکے	
2) ایڈیٹر	b) Include سٹیٹمنٹ	
3) کمپائلر	c) Pythone	
4) پروگرامنگ لینگویج	d) CLion	
5) ذخیرۂ الفاظ	e) /*(a+b)*/	
6) لنک سیکشن	f) Notepad	
7) Main() کی باڈی	g) Int weight	

8) کمنٹ	struct(h	
9) انٹیجر متغیر	4 (i بائیٹس	
10) فلوٹ	{ } (j	

پروگرامنگ کی مشقیں

مشق نمبر 1:

- استادِ محترم کی مدد سے اپنی لیب کے کمپیوٹر پر C- پروگرام لکھنے کے لیے انسٹال کی ہوئی IDE کھولیں۔
- ایڈیٹر میں یہ پروگرام لکھیں اور welcome نام رکھ کر محفوظ (save) کر لیں۔

```
#include <stdio.h>
#include <conio.h>
void main()
{
	/*A simple C language program*/
	printf(“Welcome to C language”);
	getch();
}
```

- سکرین پر "welcome to c language" بطور آؤٹ پٹ دیکھنے کے لیے پروگرام چلائیں۔

مشق نمبر 2:

ایک پروگرام لکھیں جو آپ کے بہترین دوست کا ذاتی ڈیٹا محفوظ کرنے کے لیے مناسب ڈیٹا ٹائپس کے متغیرات ڈکلیئر کرے۔ان متغیرات کو درج ذیل ڈیٹا سے انیشلائز کریں۔

- اس کے نام کا پہلا حرف
- اس کی جنس کا پہلا حرف
- اس کی عمر
- اس کے قد کی لمبائی