UNIT – 1

INTRODUCTION TO PROGRAMMING

Q 1.1:Define Software.

Ans: In order to perform all the tasks, computers have to be fed a series of instructions by humans which tell them how to behave and perform when faced with a particular type of problem. These series of instructions are known as a **computer program or software**

Q 1.2: Define Computer programing.

Ans: The process of feeding or storing these instructions in the computer is known as **computer programming.**

Q 1.3: Who is programmer?

Ans: The person who knows how to write a computer program correctly is known **as a programmer.**

Q 1.4: Define programing Language.

Ans: Programmers write computer programs in these special languages called **programming languages.**

Example: Java, C, C + +, C#, Python.

Q 1.5: Why do we need a Programming Environment?

Ans: A collection of all the necessary tools for programming makes up a programming environment. It work as a basic platform for us to write and execute programs.

Example:

For gardening we need gardening tools and for painting we need a collection of paints, brushes and canvas. Similarly we need proper tools for programming.

Importance:

1. In order to correctly perform any task, we need to have proper tools.
2. It is essential to setup a programming environment before we start writing programs.
3. It works as a basic platform for us to write and execute programs

Q 1.6: Explain integrated Development Environment.

Ans: A software that provides a programming environment to facilitate programmers. In writing and executing computer programs is known as an **Integrated Development Environment (IDE).**

Graphical User Interface:

An IDE has a Graphical User Interface (GUI), meaning that a user can interact with it using windows and buttons to provide input and get output. An IDE consists of tools that help a programmer throughout the phases of writing, executing and testing a computer program. This is achieved by combining text editors, compilers and debuggers in a single interface.

IDEs for C programming Language:

Some of the many available IDEs for C programming language are:

1. Visual Studio
2. Xcode
3. Code::Blocks
4. Dev C++

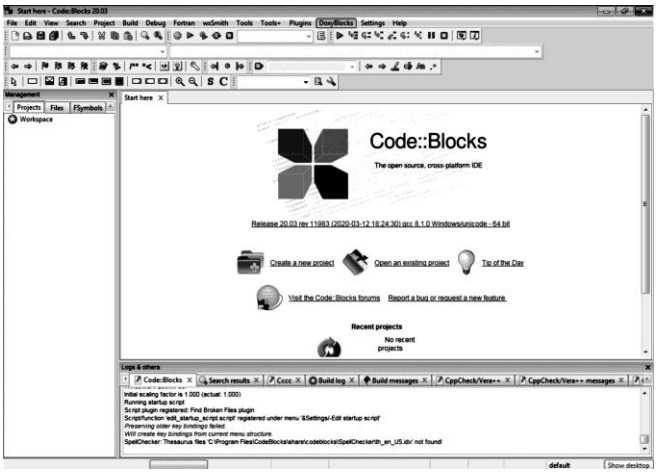


Figure 1.1 shows the main screen of Code::Blocks IDE.

ACTIVITY 1.1	Answer
Use your web browser to find out the names of three different IDEs that can be used for C programming languages.	1. CLion 2. Eclipse 3. Code Lite

Q 1.7: Describe Test Editor. / Write the steps to create a C program file in IDE.

Ans: Text Editor:

An text editor is a software that allows programmers to write and edit computer program. All IDEs have their own specific text editors.

Main Screen of an IDEs:

It is main screen of an IDE where we can write our program.

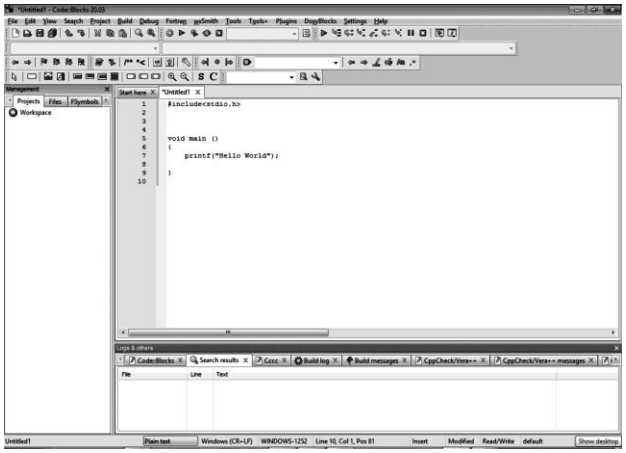


Figure 1.2: Text editor in Code:: Blocks

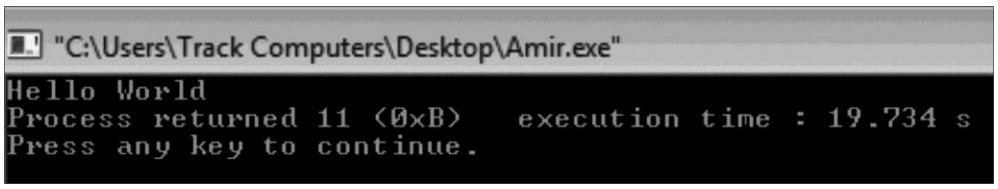
Execution and Saving Program:

When executed, this program displays *Hello World!* on computer screen. We have to save our file before it can be executed. We have named our program file as *"HelloWorld.c"*. We can click on the build and run button to see the program's output, as pointed by an arrow in **Figure 1.3.**

Figure 1.3: Running program in Code:: Blocks

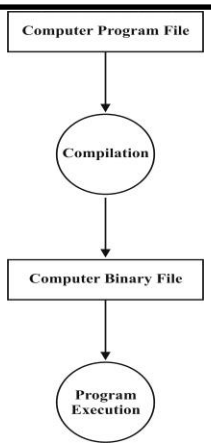
Console screen showing the output is displayed, as shown in Figure 1.4.

Output:



ACTIVITY 1.2	ANSWER
Open the IDE installed on you lab computer. Write the program written in Figure 1.2 in the text editor of your IDE and execute it.	

Q 1.8: Define compiler and describe the purpose of compiler.



Ans: A compiler is software that is responsible for conversion of a computer program written in some high level programming language to machine language code.

Purpose:

Computers only understand and work in machine language consisting of 0s and 1s. They require the conversion of a program written in programming language to machine language, in order to execute it. This is achieved using a compiler.

Example: A missing semicolon (;) at the end if a line.

Q 1.9:Define syntax and syntax error.

Ans: **Syntax:** Each programming language has some primitive building blocks and provides some rules in order to write an accurate program. This set of rules is known as **syntax** of the language.

Syntax Error: While programming, if proper syntax or rules of the programming language are not followed, the program does not get compiled. In this case, the compiler generates an error. This kind of errors are called syntax errors.

Q 1.10: What are Reserved Words?

Ans: Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the compiler. These words are known **as reserved words or keywords.**

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	size of	volatile
do	if	static	while

ACTIVITY 1.3	ANSWER
From the following list, encircle the reserved words in C language: int, pack, create, case, return, small, math, struct, program, library.	int, pack, create, case, return, small, math, struct, program, library.

Q 1.11: Discuss the main parts of the structure of C program?

Ans: Link section or header section:

While writing programs in C language, we make extensive use of functions that are already defined in the language. But before using the existing functions, we need to include the files where these functions have been defined. These files are called **header**

files. We include these header files in our program by writing the include statements at the top of program.

General Structure:

General structure of an include statement is as follows:

#include <header_file_name>

Here header_file_name can be the name of any header file.

Example:

We have included file **stdio.h** that contains information related to **input** and **output** functions. **Main section:**

It consists of a main() function. Every C program must contain a main( ) function and it is the starting point of execution.

Body of main( ) function:

The body of main( ) is enclosed in the curly braces { }. All the statements inside these curly braces make the body of main function.

Example:

```
# include <stdio.h >
void main ( )
{
printf("Hello World!");
}
```

In the above example the statement *printf("Hello world!")* uses a predefined function *printf* to display the statement Hello World! on computer screen. We can also create other functions in our program and use them inside the body of main( ) function.

ACTIVITY 1.4	ANSWER
Identify different parts of the following C program: *#include <stdio.h>* *#include <conio.h>* void main ( ) { *printf("I am a student of class 10") ;* *getch ( );* }	**Header Section:** *#include <stdio.h>* *#include <conio.h>* **Main Section:** void main ( ) **Body Section:** { *printf("I am a student of class 10") ;* *getch ( );* }

Q 1.12: Write any three rules in order to write syntactically correct C language program.

Ans: 1. The sequence of statements in a C language program should be according to the sequence in which we want our program to be executed.

2. C language is **case sensitive**. It means that if a keyword is defined with all small case letters, we cannot capitalize any letter

Example: *int* is different from *Int.*

3. Each statement ends with a semi-colon (;) symbol.

Q 1.13: What are the Purpose and Syntax of Comments in C Programs?

Ans: Comments:

Comments are the statements in a program that are ignored by the complier and do not get executed. Usually comments are written in natural language e.g. in English.

Purpose of writing comments:

Comments can be thought of as documentation of the program. Their purpose is twofold.

1. They facilitate other programmers to understand our code.
2. They help us to understand our own code even after years of writing it.

Syntax of writing comments:

In C programing language, there are two types of comments.

1. Single-line comments 2. Multi-line comments

1. Single-line comments:

Single line comments start with //. Anything after // on the same line, is considered a comment.

Example: *//This is a comment.*

2. Multi-line comments:

Multi-line comments start with /* and end at */- Anything between /* and */ is considered a comment, even on multiple lines.

Example:
```
/*this is
a multi – line
comment*/
```

Q 1.14: Write a program which demonstrates the usage of comments.

```
#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("l am a student of class 10");

} //body of main function ends here
```

ACTIVITY 1.5	ANSWER
Tick valid comments among the following: • *comment goes here* • / comment goes here / • % comment goes here % • /* comment goes here*/ • /* comment goes here/ • // comment goes here*/	*/Comment goes here*/

Q 1.15: Explain Character Set.

Ans: Character Set:

Each language has a basic set of alphabets (character set) that are combined in an allowable manner to form words, and then these words can be used to form sentences. C programming language have a *character set* that includes:

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

Q 1.16: Explain constants and its types.

Constants:

Constants are the values that cannot be changed by a program

Example: 5, 75.7,1500 etc.

Types: In C language, primarily we have three types of constants:

1. Integer Constants
2. Real Constants
3. Character Constants

1. Integer Constants:

These are the values without a decimal point

Example: 7, 1256, 30100, 55555, -54, -2349 etc.

Sign: They can be positive or negative. If the value is not preceded by a sign, it is considered as positive.

2. Real Constants: These are the values including a decimal point.

Example: 3.14, 15.3333, 75.0, -1575.76, -7941.2345 etc.

Sign: They can also be positive or negative.

3. Character Constants:

Any single small case letter, upper case letter, digit, punctuation mark, special symbol enclosed within ' ' is considered a character constant

Example:

'5', 7', 'a', 'X','!',';' etc.

ACTIVITY 1.6	ANSWER
Identify the type of constant for each of the following values:	
i. 12.	i. Integer
ii. 1.2	ii. Real
iii. '*'	iii. Character
iv. -21	iv. Integer
v. 32.768	v. Real
vi. 'a'	vi. Character
vii. -12.3	vii. Real
viii. 41	viii. Integer
ix. 40.0	ix. Real
x. '/'	x. Character

Q 1.17: Explain Variables.

Ans: Variables:

A variable is actually a name given to a memory location as the data is physically stored inside the computer's memory. The value of a variable can be changed in a program. It means that, in a program, if a variable contains value 5, then later we can give it another value that replaces the value 5.

Identifier:

Each variable has a unique name called indemnifier.

Data Type:

Data type describes the type of data that can be stored in the variable C language has different data types such as int, float, and char. The types int, float and char are used to store integer, real and character data respectively.

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

Matching data types against different types of data

Q 1.18: Explain Data Type of a variable.

Ans: Data Type of a Variable:

Each variable in C language has a data type. The data type not only describes the type of data to be stored inside the variable but also the number of bytes that the compiler needs to reserve for data storage.

1. Integer _int (signed/unsigned)

Integer data type is used to store both integer values (whole numbers). Integer takes up 4 bytes of memory.

- **Declaration:**

To declare a variable of type integer, we use the keyword **int**.

Singed int:

A signed int can store both positive and negative values ranging from –2, 147, 483, 648 to 2, 147, 483, 647. By default, type int is considered as a singed integer.

Unsigned int:

An unsigned int can store only positive values and its value ranges from 0 to +4, 294, 967, 295. Keyword **unsigned int** is used to declare an unsigned integer.

2. Floating Point -float:

Float data type is used to store a real number (number with floating point) up to six digits of precision.

- **Declaration:**

To declare a variable of type float, we use the keyword **float**.

- **Range:** A float uses 4 bytes of memory. It value ranges from 3.4×10^{-38} to 3.4×10^{38}.

3. Character:

A variable of type char can store one character only.

- **Declaration:**
  To declare character type variables in C, we use the keyword char. It takes up just 1 byte of memory for storage.

Q 1.19: Write down the rules for naming variables.

Ans: Name of a Variable:

1. Each variable must have a unique name or identifier. Following rules are used to name a variable.
2. A variable name can only contain alphabets (uppercase or lowercase), digits and underscore _ sign.
3. Variable name must begin with a letter or an underscore, it cannot begin with a digit.
4. A reserved word cannot be used as a variable name.
5. There is no strict rule on how long a variable name should be, but we should choose a concise length for variable name to follow good design practice.

Examples: Some examples of valid variable names are height, Average Weight, _var1.

ACTIVITY 1.7	ANSWER
From the following list, encircle the reserved words in C language: _Hello, 1var roll_num Air23Blue float Case $car name = color Float	_Hello, 1var roll_num Air23Blue float Case $car name = color Float

Q 1.20: How can we declare and initialize a variable?

Ans: Variable Declaration:

We need to declare a variable before we can use it in the program. Declaring a variable includes specifying its data type and giving it a valid name.

Syntax: data_type variable_name;

Examples: Some examples of valid variable declarations are as follows.

1. unsigned int age;
2. float height;
3. int salary;
4. char marital_status;

Multiple variables:

Multiple variables of same data type may also be declared in a single statements.

Examples:

1. unsigned int age, basic_salary, gross_salary;
2. int points_scored, steps.
3. float height, marks;
4. char martial_status, gender;

Declaring variable, the range of values allowed by that variable, operations that can be performed on it.

Code Example:

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

Variable Initialization:

Assigning value to a variable for the first time is called **variable initialization**. C language allows us to initiate a variable both at the time of declaration, and after declaring it.

General Structure:

For initializing a variable at the time of declaration, we use the following general structure.

data_type variable_name = value;

Example: Following example shows a program that demonstrates the declaration and initialization of two variables.

```
#includes<stdio.h>

void main ( )

{

        char grade;      // variable grade is declared

        in value = 25; /* variable value is declared and initialized.*/

        grade = 'A'; // variable grade is initialized

}
```

ACTIVITY 1.8	ANSWER
Write a program that declares variables of appropriate data types to store your personal data. Initialize these variables with the following data: • initial letter of your name • initial letter of your gander • your age • your marks in 8th class • your height	# include <stdio.h> voidmain ( ) { char name = 'H'; char gender = 'F'; int age = 23; int marks = 457; float height = 5.4; }

IMPORTNATS DEFINICATIONS

Computer Program or Software:

Computer need to be fed a series of instructions by humans which tell them how to perform a particular task. These series of instructions are known as a computer program or software.

Computer Programing:

The process of feeding or storing the instructions in the computer is known as computer programming and the person who know how to write a computer correctly known as a programmer.

Programing Language:

Computer programs are written in languages called programming languages. Some commonly known programming languages are java, C, C + +, Python.

Programing Environment:

A collection of all the necessary tools for programming makes up a programming environment. Programing environment provides us to basic platform to write and execute programs.

Integrated Development Environment (IDE):

A software that provides us programming environment which facilitates the programmer in writing and executing computer program is known as an Integrated Development Environment (IDE)

Text Editor:

A text editor is a software that allows programmers to write and edit computer programs. All IDEs have their own specific editors.

Complier:

A complier is a software that is responsible for conversion of a computer program written in some programming language to machine language code.

Syntax:

Every programming language has some primitive building blocks and follows some grammar rules known as its syntax.

Reserved words or Keywords:

Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the complier. These words are known as reserved words or Keywords.

Header Section, Main section body of the main function:

A program is divided into three parts. **Header section** is the part where header files are included. **Main section** corresponds to the main function and the body of the main function includes everything enclosed in the curly braces.

Comments:

Comments are the statements that are ignored by the complier and do not get executed. To include additional information about the program, comments can be used.

Constants:

Constants are that do not change. The three types of constants are integer constants, real constants and character constants.

Variables unique name or identifier data type:

Variable is a name given to a memory location as the data is physically stored inside the computer's memory. Each variable has a unique name or **identifier** by which we can refer to that variable, and an associated **data type** that describes the type of constant that can be stored in that variable.

Variable declaration:

A variable must be declared before its use. Variable declaration includes specifying variable's data type and giving it a valid name.

Variable initialization:

Assigning value to a variable for the first time is called variable initialization. The variable can be initialized at the time of declaration or after declaration.

EXERICE

Q.1: Multiple Choice Question:

1. A software that facilitates programmers in writing computer program is known as ____.

(a) a compiler (b) an editor (c) an IDE (d) a debugger

2. _____is a software that is responsible for the conversion of program files to machine understandable and executable code.

(a) Compiler (b) Editor (c) IDE (d) Debugger

3. Every programming language has some primitive building blocks and follows some grammar rules known as its ———.

(a) programming rules (b) syntax (c) building blocks (d) semantic rules

4. A list of words that are predefined and must not be used by the programmer to name his own variables are known as _____.

(a) auto words (b) reserved words (c) restricted words (d) predefined words

5. include statements are written in _____section.

(a) header (b) main (c) comments (d) print

6. _____ are added in the source code to further explain the techniques and algorithms used by the programmer.

(a) Messages (b) Hints (c) Comments (d) Explanations

7. _____ are the values that do not change during the whole execution of program.

(a) Variables (b) Constants (c) Strings (d) Comments

8. A float uses _____ bytes of memory.

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

9. For initializing a variable, we use _____ operator.

(a) → (b) = (c) @ (d) ?

10. _____ can be thought of as a container to store constants.

(a) box (b) jar (c) variable (d) collection

MCQs Keys

1	c	**2**	a	**3**	b	**4**	b	**5**	a	**6**	c	**7**	b	**8**	b	**9**	b	**10**	c

Q.2: True of False

1) An IDE combines text editors, libraries, compilers and debuggers in a single interface. ✓T/F
2) Computers require the conversion of the code written in program file to machine language in order to execute it. ✓T/F
3) Column is a reserved word in C programming language. T/F✓
4) *comment goes here* is a valid comment. T/F✓
5) Float can store a real number up to six digits of precision. ✓T/F

Q.3: Define the following.

1. IDE:

Ans: A software that provides us programming environment which facilitates the programmer in writing and executing computer program is known as an Integrated Development Environment (IDE)

2. Compiler:

Ans: A complier is a software that is responsible for conversion of a computer program written in some programming language to machine language code.

3. Reserved Words:

Ans: Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the complier. These words are known as reserved words or Keywords.

4. Main section of a program:

Ans: Main section corresponds to the main function and the body of the main function includes everything enclosed in the curly braces.

5. char data type:

Ans: A variable of type char can store one character only. To declare character type variables in C, we use the keyword char. It takes up just 1 byte of memory for storage.

Q.4 Briefly answer the following questions.

1) Why do we need a programming environment? See (Q: 1.5)
2) Write the steps to create a C program file in the IDE of your lab computer. See (Q: 1.6)
3) Describe the purpose of a compiler. See (Q: 1.7)
4) List down five reserved words in C programming language. See (Q: 1.9)
5) Discuss the main parts of the structure of a C program. See (Q: 1.10)
6) Why do we use comments in programming? See (Q: 1.12)

7) Differentiate between constants and variables. See (Q: 1.15 & Q: 1.16)

8) Write down the rules for naming variables. See (Q: 1.18)

9) Differentiate between char and int.

char	Int
1. To declare character type variable in C, we use the keyword char. 2. It takes up just i byte of memory for storage. 3. A variable of type char can store one character only.	1. To declare a variable of type integer, we use the keyword int. 2. Integer takes up 4 bytes of memory. 3. Integer data type is used to store integer values (whole numbers).

10) How can we declare and initialize a variable? See (Q: 1.19)

Q.5 Match the columns.

A	B	C
(1) IDE	(a) Machine executable code	(d)
(2) Text Editor	(b) include statement	(f)
(3) Complier	(c) Python	(a)
(4) Programing Language	(d) CLion	(c)
(5) Reserved words	(e) /*(a + b)*/	(g)
(6) Link Section	(f) Notepad	(b)
(7) Body of main ( )	(g) struct	(f)
(8) Comment	(h) { }	(e)

PROGRAMMING EXERICE

Exercise # 1

- With the help of your teacher open the IDE installed on your lab computer for writing C programs.
- Write the following program in the editor and save it as "welcome.c".

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

- Run the program to see Welcome to C language printed on the screen as output.

Ans:

Exercise # 2

Write a program that declares variables of appropriate data types to store the personal data about your best friend. Initialize these variables with the following data:

- initial letter of his name

- initial letter of his gender
- his age
- his height

Ans:
```
# include <stdio.h>
void main ( )
{
char name = 'H';
char gender = 'F';
int age = 23;
int marks = 457;
float height = 5.4;
}
```

SHORT QUESTIONS

1. Define Software.

Ans: In order to perform all the tasks, computers have to be fed a series of instructions by humans which tell them how to behave and perform when faced with a particular type of problem. These series of instructions are known as a **computer program or software**.

2. Define Computer Programming.

Ans: The process of feeding or storing these instructions in the computer is known as **computer programming.**

3. Define Programmer.

Ans: The person who knows how to write a computer program correctly is known **as a programmer.**

4. Define Programing Language.

Ans: Programmers write computer programs in special languages called **programming languages.**

5. Give at least three examples of Programing Language.

Ans: The examples of programming language are following.

i.. Java ii. C
iii. C + +

6. Which languages are used most commonly in programing?

Ans: Java, C, C + +, C# and Python are some of the most commonly used programming languages.

7. Who developed the C language and when?

Ans: C language was developed by **Dennis Ritchie** between 1969 and 1973 at **Bell Labs**.

8. Define programming environment.

Ans: A collection of all the necessary tools for programming makes up a programming environment. It work as a basic platform for us to write and execute programs.

9. Write down the importance of programing environment.

Ans:
1. In order to correctly perform any task, we need to have proper tools.
2. It is essential to setup a programming environment before we start writing programs.
3. It works as a basic platform for us to write and execute programs.

10. Define IDE.

Ans: A software that provides us programming environment which facilitates the programmer in writing and executing computer program is known as an Integrated Development Environment (IDE).

11. Define. GUI

Ans: An IDE has a graphical user interface (GUI), meaning that a user can interact with it using windows and buttons to provide input and get output.

12. Write down the tools of IDE.

Ans: An IDE consists of tools that help a programmer throughout the phases of writing, executing and testing a computer program. This is achieved by combining text editors, compilers and debuggers in a single interface.

13. Give at least three examples of IDE.

Ans: Some of the many available IDEs for C programming language are:

1. Visual Studio
2. Xcode
3. Code::Blocks
4. Dev C++

14. Define text editor.

Ans: An text editor is a software that allows programmers to write and edit computer program. All IDEs have their own specific text editors.

15. Define complier.

Ans: A compiler is software that is responsible for conversion of a computer program written in some high level programming language to machine language code.

16. In which language a computer can work?

Ans: Computers only understand and work in **machine language** consisting of **0s** and **1s**.

17. Define syntax.

Ans: Each programming language has some primitive building blocks and provides some rules in order to write an accurate program. This set of rules is known as **syntax** of the language.

18. Define syntax error.

Ans: While programming, if proper syntax or rules of the programming language are not followed, the program does not get compiled. In this case, the compiler generates an error. This kind of errors are called syntax errors.

Example: A missing semicolon (;) at the end of a line.

19. Define reserved words.

Ans: Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the complier. These words are known as reserved words or Keywords.

20. List down five reserved words in C programming language.

Ans: 1. auto 2. break 3. int
4. char 5. float

21. Write down the name of main parts of C language.

Ans: The main parts of C language are following:

1. Link Section / Header Section
2. Main Section
3. Body Section

22. Define link section.

Ans: While writing programs in C language, we make extensive use of functions that are already defined in the language. But before using the existing functions, we need to include the files where these functions have been defined. These files are called header files.

23. Write general structure of header section.

Ans: General structure of an include statement is as follows:
#include<header_file_name>

24. Define main section.

Ans: It consists of a main() function. Every C program must contain a main() function and it is the starting point of execution.

25. Define body section with example.

Ans: The body of main() is enclosed in the curly braces { }. All the statements inside these curly braces make the body of main function.

26. Write a C program and show header section, main section, body section.

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

In the above example the sections of C language are following:

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

27. Define comments.

Ans: Comments are the statements in a program that are ignored by the complier and do not get executed.

28. Write down the purpose of comments in C language.

Ans: Comments can be through of as documentation of the program. Their purpose is twofold.

1. They facilitate other programmers to understand our code.
2. They help us to understand our own code even after years of writing it.

29. How many types of comments in C language?

Ans: There are types of comments in C language.

1. Single-line comment 2. Multi-line comment

30. Describe single line comment with example.

Ans: Start with //. Anything after // on the same line, is considered a comment.

Example:

//This is a comment.

31. Describe multi line comment with example.

Ans: Start with /* and end at */- Anything between /* and */ is considered a comment, even on multiple lines.

Example:

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

32. Write a C language program with single line comment.

Ans:

```
#include <stdio.h>
void main( )
{ //body of main function starts from here
printf("l am a student of class 10");
} //body of main function ends here
```

33. Write a C language program with multi line comment.

Ans:

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

34. Identify the given instruction.

/ * This is a
comment * /

Ans: This is a multi-line comment.

35. Write any five special symbols.

Ans: **i.** **/** **ii.** ***** **iii.** **!** **iv.** **@** **v.** **&**

36. Define constants. Give example.

Ans: Constants are that do not change. The three types of constants are integer constants, real constants and character constants.

Examples: 15, 27, 35, 481

37. How many types of constants?

Ans: There are three types of constants.

i. Real constants ii. Integer constants
iii. Character constants

38. Define integer constants. Also give an example.

Ans: Integer Constants:

These are the values without a decimal point

Example: 7, 1256, 30100, 55555, -54, -2349 etc.

39. Define real constants, give example.

Ans: Real Constants:

These are the values including a decimal point.

Example: 3.14, 15.3333, 75.0, -1575.76, -7941.2345 etc. They can be positive or negative.

40. Define character constants. Also give an example.

Ans: Character Constants:

Any single small case letter, upper case letter, digit, punctuation mark, special symbol enclosed within ' ' is considered a character constant

Example: '5', 7', 'a', 'X','!',';' etc.

41. Write any three character constants.

Ans: i. 'X' ii. '9'
iii. '!'

42. Can we find the sum of two integer constants?

Ans: We can add two integer constants to get the obvious mathematical result.

Examples:

9 + 8 = 17

43. Write any three integer constants.

Ans: i. 55555 ii. 1256
iii. 30100

44. Write any three real constants.

Ans: i. 3.14 ii. 15.3333
iii. 55555

45. Write the sum of two character constants.

Ans: We cannot add a character constant to another character constant to get the obvious mathematical result e.g. '9' + '8' =17.

46. Explain ('9' + '8' = 17).

Ans: We cannot add a character constant to another character constant to get the obvious mathematical result.

47. Define variable.

Ans: Variables is a name given to a memory location as the data is physically stored inside the computer's memory.

48. Define identifier.

Ans: Each variable has a unique name or **identifier** by which we can refer to that variable.

49. Define data type.

Ans: An associated data type that describes the type of constant that can be stored in that variable is called data type of that variable.

50. Write any three data type.

Ans: i. Real ii. Integer iii. Character

51. Define integer data type.

Ans: Integer data type is used to store both integer values (whole numbers), we use takes up 4 bytes of memory.

52. What is the range of signed integer data type?

Ans: A signed int can store both positive and negative values ranging from –2, 147, 483, 648 to 2, 147, 483, 647.

53. Define singed int.

Ans: A signed int can store both positive and negative values ranging from –2, 147, 483, 648 to 2, 147, 483, 647. By default, type int is considered as a singed integer.

54. Write down the keyword used to declare singed.

Ans: To declare a variable of type integer, we use the keyword int.

55. Define unsigned int.

Ans: An unsigned int can store only positive values and its value ranges from 0 to +4, 294, 967, 295. Keyword unsigned int is used to declare an unsigned integer.

56. Write down the keyword used to declare unsigned integer.

Ans: To declare a variable of type integer, we use the keyword unsigned int.

57. Define float data type.

Ans: Float data type us used to store a real number (number with floating point) up to six digits of precision.

58. How many bytes used to store float data type?

Ans: A float uses 4 bytes of memory.

59. Define character data type.

Ans: A variable of type char can store one character only.

60. How many bytes used to store character data type?

Ans: It takes up just 1 byte of memory for storage.

61. Write down any two rules for a name of a variable.

Ans:
1. A variable name can only contain alphabets (uppercase or lowercase), digits and underscore sign.
2. Variable name must begin with a letter or an underscore, it cannot begin with a digit.

62. Write syntax to declare a variable.

Ans: We need to declare a variable before we can use it in the program. Declaring a variable includes specifying its data type and giving it a valid name.

Syntax: data_type variable name;.

63. Write a program in C language using float.

Ans:
```
# include <stdio.h>
voidmain ( )
{
float height = 5.4;
printf ("%f", & height);
}
```

64. Write a program in C language using singed integer.

Ans:
```
# include <stdio.h>
voidmain ( )
{
int age = 23;
printf ("%d", & age);
}
```

65. Describe variable initialization.

Ans: Assigning value to a variable for the first time is called variable initialization. C language allows us to initiate a variable both at the time of declaration and after declaring it.

66. Write methods of variable initialization.

Ans: C language allows us to initiate a variable both at the time of declaration and after declaring it.

67. Write syntax for variable initialization.

Ans: For initializing a variable at the time of declaration, we use the following general structure.

data_type variable_name = value;

68. Write a program that shows the declaration and initialization of two variables.

Ans:
```
#includes<stdio.h>
void main ( )
{
        char grade;      // variable grade is declared
        in value = 25; /* variable value is declared and initialized.*/
        grade = 'A'; // variable grade is initialized
}
```

ADDITIONAL EXERICE

1. The series of instruction called.
(a) Software (b) Computer Program (c) IDE (d) GUI

2. The language that are used most commonly in programming.
(a) Java (b) HTML (c) C ++ (d) Python

3. The name of developer of C language.
(a) Dennis Ritchie (b) Newton (c) Hashim (d) Einstein

4. Programing environment is essential to set up before _____ program.
(a) start (b) end (c) compile (d) Both a and b

5. IDE stand for.
(a) Integrated development environment (b) Integrated developed environment
(c) Intermediate developed environment (d) None of these

6. GUI stand for.
(a) Graphical user inter fare (b) Geological used integrated
(c) Graphical uses integrate (d) None of these

7. A software that allow computer to write program.
(a) Notepad (b) Text editor (c) Not pad ++ (d) None of these

8. Computer can understand.
(a) 0 and 1 (b) 0 to 9 (c) 0 to 16 (d) None

9. A software that convert programing language to machine language is _____.
(a) Text editor (b) complier (c) execute (d) None

10. If rules of programing are not followed, the program.
(a) complied (b) not complied (c) not execute (d) None of these

11. Every programing language has a list of words that are.
(a) not defined (b) fully defined (c) predefined (d) None of these

12. Every programing language has a list of words that predefined called.
(a) Reserved word (b) Alphabet word (c) Keywords (d) Both a and c

13. If a programmer the definition of his own it cause.
(a) Syntax (b) error (c) Syntax error (d) None of these

14. double is a.
(a) Reserved word (b) Keywords (c) Alphabet word (d) Both a and c

15. The Dennis Ritchie develop the C language between.
(a) 1953 to 1963 (b) 1973 to 1983 (c) 1963 and 1973 (d) None

16. The example of IDE.
(a) Code :: Block (b) Java (c) C++ (d) None

17. How many main parts of C program?
(a) 3 (b) 4 (c) 7 (d) 6

18. Link section is also called.
(a) Body (b) Header (c) Main (d) None

19. Which statement used in C program?
(a) .C (b) .html (c) include (d) None

20. Math is used for.
(a) English words (b) Urdu (c) Arabic functions (d) Mathematical

21. Main section consist on.
(a) main ( ) (b) include (c) Body (d) None

22. The body of main ( ) enclosed in.
(a) < > (b) { } (c) [ ] (d) ( )

23. C language statement ends with.
(a) , (b) : (c) ; (d) -

24. Comments are the statements that are ignored by _____.
(a) Text editor (b) Runner (c) a, b both (d) complier

25. Comments are written in.
(a) b, c both (b) Natural language (c) English language (d) None

26. Comments have _____ purposes.
(a) 6 (b) 2 (c) 3 (d) 9

27. Comments have _____ types.
(a) 3 (b) 4 (c) 2 (d) 5

28. // is _____ comment.
(a) Dabble slash (b) Multiline (c) Back slash (d) Single line

29. // is used for _____ comment.**
(a) Multiline (b) Steric (c) a, b both (d) None

30. Single line comment enclosed in.
(a) /**/ (b) // (c) \\ (d) / \

31. (!) is _____.
(a) sign (b) exclamation (c) special symbol (d) None

32. 1500 is _____.
(a) Real (b) Integer (c) Constant (d) b, b both

33. How many types of constants?
(a) 3 (b) 4 (c) 5 (d) 6

34. A value do not have decimal.
(a) Real (b) Integer (c) character (d) None

35. A value have decimal.
(a) Integer (b) Character (c) Real (d) None

36. Character constant is enclosed in.
(a) " " (b) ( ) (c) { } (d) ' '

37. 'X' is.
(a) Character (b) Real (c) Integer (d) None

38. 'a' + '8' _____17.
(a) = (b) ≠ (c) a, b both (d) None

39. 9 + 8 _____ 17.
(a) ≠ (b) a, c both (c) = (d) None

40. 40.0 is.
(a) Integer (b) Character (c) All (d) Real

41. Integer takes up _____ bytes of memory.
(a) 1 (b) 2 (c) 3 (d) 4

42. To declare a variable of type integer we use the keyword _____.
(a) int (b) Char (c) Float (d) Struct

43. A singed int can store _____ values.
(a) positive (b) negative (c) a and b both (d) None

44. The smallest range of signed int is _____.
(a) –2147, 483648 (b) 2435 (c) 9399 (d) –893

45. The largest range of signed int is _____.
(a) 2147483647 (b) 2435 (c) 9399 (d) 9999

46. To declare a variable of type singed int we use the keyword _____.
(a) singed integer (b) int (c) Char (d) Float

47. An unsigned int can store _____ values.
(a) positive (b) negative (c) a and b both (d) None

48. The smallest range of unsigned int is _____.
(a) 0 (b) 2 (c) 3 (d) 4

49. The greatest range of unsigned int is _____.
(a) 45678,9 (b) +4294967295 (c) 123 (d) 456

50. To declare a variable of type unsigned integer we use the keyword _____.
(a) unsigned int (b) int (c) char (d) Struct

51. Float data type is used to store _____.
(a) real number (b) binary (c) Decimal (d) Hexadecimal

52. _____ data type is used to store real number.
(a) float (b) int (c) char (d) Struct

53. To declare a variable of type real number we use the keyword _____.
(a) int (b) char (c) Struct (d) float

54. A float takes up _____ bytes of memory.
(a) 1 (b) 5 (c) 4 (d) 3

55. The smallest range of float is _____.
(a) 3.4×10^{-38} (b) 3.4×10^{-3} (c) 10^{-3} (d) 3.5×10^{-6}

56. The largest range of float is _____.
(a) 3.4×10^{-36} (b) 3.4×10^{38} (c) 3.3×10^{38} (d) 3.5×10^{3}

57. To declare a variable of type character we use the keyword _____.
(a) char (b) int (c) float (d) Struct

58. A character takes up _____ bytes of memory.
(a) 1 (b) 2 (c) 3 (d) 4

59. _____ word cannot be used as a variable name.
(a) reserved (b) real (c) single (d) Binary

60. A variable can only contain _____ and digit underscore _ sign.
(a) alphabets (b) symbol (c) both (d) None

61. A variable can only contain alphabets, _____ and underscore _ sign.
(a) digit (b) symbol (c) both (d) None

62. A variable name can only contain alphabets digits and _____ sign.
(a) underscore (b) symbols (c) both (d) None

63. There are _____ methods for variable initialization.
(a) two (b) three (c) four (d) five

MCQs Keys

1	b	**2**	d	**3**	a	**4**	a	**5**	a	**6**	d	**7**	b	**8**	a	**9**	b	**10**	b
11	c	**12**	d	**13**	c	**14**	d	**15**	c	**16**	a	**17**	a	**18**	b	**19**	c	**20**	d
21	a	**22**	b	**23**	c	**24**	d	**25**	a	**26**	b	**27**	c	**28**	d	**29**	a	**30**	b
31	c	**32**	c	**33**	a	**34**	b	**35**	c	**36**	d	**37**	a	**38**	b	**39**	c	**40**	d
41	d	**42**	a	**43**	c	**44**	a	**45**	a	**46**	a	**47**	a	**48**	a	**49**	b	**50**	a
51	a	**52**	a	**53**	d	**54**	c	**55**	a	**56**	b	**57**	a	**58**	a	**59**	a	**60**	a
61	a	**62**	a	**63**	a														