C Theoretical:

1: Define programming?

Ans: Programming is a process of solving a problem with the help of computer system. It prepares different instructions for computer.

2: What is a program?

Ans: A set of instructions that tells a computer what to do is called program. The computer works according to the given instructions in the program. A person who develops a program is called a programmer.

3: Define top down design?

Ans: The technique of dividing a problem in small parts is called top down design.

4: What is an algorithm and pseudo code?

Ans: An algorithm is a step-by-step procedure to solve a problem. It is better to write an algorithm before writing the actual program. Algorithms are written in a language that is similar to simple English called Pseudo code.

5: What are main parts of a pseudo code?

Ans: Pseudo code simplifies program development by separating it into two main parts:

(|)Logic design: In this part logic of the program is designed. We specify different steps required to solve the problem and the sequence of these steps.

(||)Coding: In this part algorithm is converted into program. The steps of algorithm are translated into instructions of any programming language.

6: What is Flow chart and why it is designed to solve a problem?

Ans: A flow chart is the graphical representation of an algorithm. It helps to understand the algorithm easily. It is also helpful in understanding the flow of control and data in algorithm.

7: what is the use of programming languages?

Ans: A set of words, symbols, and codes used to write programs is called program language. Different programming languages are available for writing different types of programs. Some languages are specially used for writing business programs. Others are used for writing scientific programs.

8: List some types of major programming languages?

Ans: There are two main types of programming languages:

Low level languages:

These languages are near to computer hardware and far from human languages. Computer can understand these languages easily. Writing a program in these languages requires a deep knowledge of internal structure of computer hardware. Machine language and assembly languages are examples of low level languages.

High level Languages:

High level languages are close to human languages but far from computer hardware. High level languages are easy to understand. Instructions of these languages are written in English-like words such as input and print etc. C, C++, Java and Pascal are the examples of high level languages.

9: What is machine language?

Ans: A type of low level language in which instructions ae written in binary form is called machine language. It is the only language that is directly understood by computer. It is the fundamental language of the computer. It does not require any compiler. Instructions written in machine language run faster. It disadvantage is that it is machine independent. Each type of hardware requires its own machine language.

10: What is assembly language?

Ans: Assembly language is a low level language. It is one step higher than machine language. In assembly language symbols are used instead of binary code. It is easier to understand than machine language.

11: What are the types of high level languages?

Ans: There are three types of high level languages:

1) Procedural languages: In these languages a program is a predefined set of instructions. Computer executes these instructions in the same sequence in which these instructions are written. Each instruction in this language tells the computer what to do and how to do. Examples of procedural languages are FORTRAN, COBOL, PASCAL, BASIC and C.
2) Object-oriented Languages: OOP is a technique in which programs are written on the basis of objects. An object is a collection of data and functions. Object may represent a person, thing or place in real world.
3) Non-Procedural languages: In non-procedural languages user only needs to tell the computer “what to do” not “how to do”. Its advantage is that they can be used by non-technical user to perform a specific task. Examples are Structured Query Language (SQL) and Report Program Generator (RPG).

12: What is the difference between high level and low level languages?

Ans:

High-level Language	Low-level Language
1. Easy to learn.	1. Difficult to learn.
2. Near to human language.	2.Far from human languages
3. Translator is required.	3. No translator is required.
4. Programs in high level language are slow in execution.	4. Programs in high level language are fast in execution.
5. They are easy to modify.	5. They are difficult to modify.
6. These languages do not provide much facility at hardware level.	6. These languages provide facility to write programs at hardware level.
7. These languages are normally used to write application programs.	7. These languages are normally used to write hardware programs.

13: What are the Main types of code?

Ans: There are the two main types of codes: Source code and Object code.

14: What is source code? Why it can be executed directly?

Ans: Source code is a computer program written in a high-level language like C, C++, Visual Basic or Java. Computer cannot understand the statements of high-level languages. So the source code cannot be executed by the computer directly. It is converted into machine code and then executed.

15: What is object code?

Ans: An object code is a program that is translated by a language processor. It is also called machine code. Computer understands object code directly.

16: What is the purpose of language processor?

Ans: A program written in high-level or assembly language cannot be run on a computer directly. It must be converted into machine language before execution. Language processor or translator is software that converts these programs into machine language.

17: List out different types of language processors?

Ans: Different types of language processors are as follows:

1) Compiler: Compiler converts the instruction of high-level language into machine language as a whole.
2) Interpreter: An interpreter is a program that converts one statement of a program at one time.
3) Assembler: Assembler translates the instruction of an assembly language into machine language.

18: What is difference between compiler and interpreter?

Ans:

Compiler	Interpreter
1. Compiler converts the program into machine code as a whole.	1. Interpreter converts the program into machine code statement by statement.
2. Compiler creates object code file.	2. Interpreter creates object code file.
3. Program execution is fast.	3. Program execution is slow.
4. It displays error after compiling the whole program.	4. It displays error on each statement of program.

19: What is compiling?

Ans: The process of converting source code into object code is known as compiling.

20: What is C language?

Ans: The C language was evolved from B by Dennis Ritchie at Bell Laboratories. C uses many of the important concepts of BCPL and B while adding data typing and other powerful features. • C initially became widely known as the development language of the UNIX operating system

21: What is C standard library?

Ans: The standard C library is a set of built in functions that a programmer uses in its programs.It contains built in functions like printf, sacnf, getch and etc…

22: What are white spaces?

Ans: White spaces are used in the program to increase readability. White spaces include space, tab and carriage return etc.

23: What is preprocessor directive?

Ans: Preprocessor directive is an instruction given to the compiler before the execution of the actual program. It is also called compiler directive. It modifies C program before compilation. Semicolon is not used at the end of preprocessor directives. It starts with # symbol and are written at the start of the program.

24: What is purpose of the statement # include<stdio.h>?

Ans: The # sign indicates that this is an instruction for the compiler. Stdio.h stands for standard input/output. The word “include” is a preprocessor directive that is used to include a header file stdio.h in user’s program.

25: What is the purpose of header files?

Ans: Header files are collection of standard library functions to perform different tasks. Each header file contains different types of predefined functions. Many header files can be included in the program before calling any of its functions in the program. The extension of a header file is .h. The include preprocessor directive is used to include header files in the program.

26: What is the main function used in C programs?

Ans: The main() function is the place where the execution of a C program starts. When the program is executed the control enters main() function and starts executing its statements. Each program contains a main() function.

27: What do you know about C statements?

Ans: A statement in C language is an instruction for the computer to perform a task. The statements are written in curly brackets. Each statement in C is terminated by a semicolon otherwise the computer generates an error.

28: What do you mean by delimiters?

Ans: The statements are written in curly braces. The curly brace { is called opening brace and } is called closing brace. The braces are also known as delimiters. These statements are collectively called the body of the program.

29: Why are comments used in C programs?

Ans: Comments are used to increase readability of the program. Comments are notes about different lines of code that explain the purpose of the code. The user can insert information notes in the code. It helps in debugging and modifying the program later.

30: Where and how comments can be added in the program?

Ans: Comments can be added anywhere in the code. Comments can be added in programs in two ways.

i. Comments on single line are added by using double slash ”//”. Anything written on the right side of double slash is considered as comments and is ignored during execution.
ii. Comments on multiple lines are added by using “/*” and “/*” symbols.

Compiler ignores these comments so it does not affect the size of executable program.

32: Define a bug?

Ans: An error in a computer program is known as bug. The programmer can make different errors while writing programs. The compiler detects errors and displays error message to describe the cause.

33: Write types of errors?

Ans: There are three types of errors:

i. Syntax error.
ii. Logical error

34: What is a syntax error?

Ans: A collection of rules for writing programs in a programming language is known as syntax. All programs are written according to these rules. Syntax error is a type of error that occurs when an invalid statement is written. The compiler detects syntax errors and display error message to describe the cause of error. A program containing syntax error cannot be compiled successfully. Syntax errors may include semicolon missing at the end of statement, a misspelled keyword or any of the delimiters is missing.

35: What are logical errors? When and why does it occur?

Ans: A type of error that occurs due to poor logic of the programmer is known as logical. A logical error occurs when the program follows a faulty algorithm. A statement with logical error may produce unexpected and wrong result. For example, the programmer may have written a wrong formula to computer an answer. The formula will be executed without any error but will produce wrong result.

36: What is an identifier? What the legal characters for an identifier?

Ans: An identifier is a word that represents a certain program entity. Identifiers are used for variables, constants, functions etc. The legal characters to construct identifier include upper alphabets from A to z, lower alphabets from a to z, digits from 0 to 9 and underscore characters.

37: What is difference between standard identifier and user-defined identifier?

Ans: A type of identifier that has a special meaning is C is known as standard identifier. Count, while and do are the examples of standard identifiers.

The type of identifier that is defined by the programmer to access memory location is known as user-defined identifier. These are used to store data and program results. Some examples are age, marks, area etc.

38: What is a keyword?

Ans: Keyword is a word in C language that has a predefined meaning and purpose. It cannot be changed or defined by the user. It can be used for the same purpose for which it is defined. Examples include if, while, int, const etc.

39: What is meant by data type?

Ans: The data type defines a set of values and set of operations on those values. The data and its type are defined before designing the actual program used to process data.

40: Name and give the purpose of different data types available in C?

Ans: int is used to store integers. 10, 25 and -250 are examples of integer data types. Float is used to store a real or floating point value (value with decimal point). It is also called real data. 10.5, 5.2 and -10.5 are the examples of real data. A char data type stores a single character including letters, punctuation marks and digits. 'a', '5' and '#' are its examples.

41: What is a variable?

Ans: A variable is a named memory location or a memory cell. It is used to store program's input and computational results during the execution of program. The value of the variable can be changed during the execution of the program. However, the name of the variable cannot be changed.

42: What do you mean by variable declaration?

Ans: The process of verifying variable's name and its type is called variable declaration. I program can have as many variables needed. A program can be declared anywhere in the program before its use.

Syntax: data_type variable_name;

Examle: int marks;

float average;

43: Write some rules for declaring variables?

- Ans: Variable may include letters, numbers and underscore (_).

- A variable can be up to 31 characters long for many compilers. If a character consists of more than 31 characters, only first 31 characters will be used. The remaining characters will be ignored.
- Blank spaces are not allowed in variable names. The variables “my var” and “your car” are invalid.

44: Define variable initialization?

Ans: The process of assigning a value to a variable at the time of declaration is known as variable initialization. The equal sign = is used to initialize a variable. Variable name is given on left side and value is given on the right side of equal sign.

```
Syntax: data_type variable=value;
Example: int marks=100;
         float average=0.5;
```

44: What are constants?

Ans: Constant is a quantity that cannot be changed during program execution.

45: What are escape sequence?

Ans: Escape sequences are the characters or symbols used to modify the output style. For example \n is used to jump on the next line. \t is used to insert a tab.

46: Why is it important to assign a data type to a variable?

Ans: Variables are assigned data types to specify the amount of memory occupied by the variable. Different data types occupy different amount of memory.

47: What are operators?

Ans: Operators are the symbols that are used to perform certain operations on data. C provides a variety of operators. These include arithmetic operators, relational operators, logical operators, bitwise operators etc.

48: What is arithmetic operator?

Ans: Arithmetic operator is a symbol that performs mathematical operations on data. Arithmetic operators include +, -, *, / and %.

49: What does the symbol = do in C?

Ans: The symbol =is called assignment operator. It is used to assign a value to the variable.

50: What is the difference between pre increment and post increment operators ++x and x++, respectively?

Ans: Pre increment operator increases the value of x then uses the value of x in the statement. Post increment operator uses the current value of x in the statement and then performs the increment.

51: What is difference between unary and binary operators?

Ans: The unary operator works with one operand such as ++x. The binary operator works with two operands such as x+y.

52: List out two forms of increment operator?

Ans: Increment operator can be used in two forms of prefix and postfix form. In prefix form, the increment operator is written before variable like ++y; It increments value of variable by 1. In postfix, increment operator is written after the variable like y++; It increments the value of variable y by 1.

53: List out two forms of decrement operator?

Ans: Increment operator can be used in two forms of prefix and postfix form. In prefix form, the decrement operator is written before variable like --y; It decrements value of variable by 1. In postfix, decrement operator is written after the variable like y--; It decrements the value of variable y by 1.

54: Define operator precedence?

Ans: The order in which different types operators are evaluated in an expression is known as operator precedence. It is also known as hierarchy of operators.

55: Define associativity of operators?

Ans: The order in which operators of same precedence are evaluated is known as associativity. The operators are evaluated from left-to-right or right-to-left in an expression.

56: What is type casting?

Ans: Type casting is the process of converting the type of variable during execution. Type casting is performed in two ways: Implicit type casting and explicit type casting.

57: What is implicit type casting? Why is it used?

Ans: A type of type casting that is performed automatically by the C compiler is called type casting. The operands in arithmetic operation must be of similar types. If the data types of the operands are different, the operand with lower data type is converted into higher data type.

58: What is stream? What are two types of streams?

Ans: A stream can be defined as flow of data. It is an object that is used by a program to insert or extract characters. There are two types of stream known as input stream and output stream. A sequence of characters from an input device to computer is called input device. A sequence of characters from computer to an output device is called output stream.

59: What is a control structure?

Ans: A statement used to control the flow of execution in a program or function is called control structure. The control structures in C are used to combine individual instruction in a single logical unit. The logical unit has one entry point and 1 exit point.

60: What are the basic control structures for writing a program?

Ans: The basic control structures for writing a program are sequence, selection and repetition.

61: Write three selection and three repetition statements?

Ans: Selection statements are if, if…..else and switch. The repetition statements are while, do….while, for.

62: What is relational operator? List out different relational operators?

Ans: The relational operators are used to specify conditions in programs. A relational operator compares two values. It produces result as true or false. The relational operators are sometimes called conditional operators or comparison operators as they test conditions that are either true or false. C provides different relational operators. These are >, <, >=, <= and !=.

63: What is relational expression? Give some examples of relational expressions?

Ans: Relational expression is a statement that uses relational operator to compare two values. Examples of relational expressions are A>B, A<B, A<=B, A>=B, A==B and A!=B.

64: When must use curly braces {} with selection or repetition statement?

Ans: The curly braces are required if the selection or repetition statement has more than one statement in its body.

65: What is compound condition? Give its examples?

Ans: A type of comparison in which more than one conditions are evaluated is called compound condition. It is used to execute a statement or set of statements by testing many conditions. For example, a program inputs two numbers. It displays OK if one number is greater than 100 and second number is less than 100. Compound condition is executed by using logical operators.

66: What is dangling else?

Ans: When an if…else statement is used as the statement part of another if statement, it is not clear that else is associated with which if statement. It is known as dangling else.

67: What is switch statement and why it is used?

Ans: Switch statement is a good alternative of nested if…else. It can be used easily when there are many choices available and only one should be executed. Nested if becomes very difficult in such situations.

68: What are logical operators?

Ans: Logical operators are used to evaluate compound conditions. There are three logical operators in C language.

- AND operator (&&)= It produces true result if both conditions are true.
- OR operator (||) = It produces true if either condition is true.
- NOT operator (!) = It is used to reverse the result of a condition. It produces true result if the condition is false.

69: Syntax of if statement?

```
Ans: if (condition){
Statement1…..;
Statement2……;
.
.
.
Statement N……;
}
```

70: Multiple ‘if else if’ structure.

Ans: if-else-if statement can be used to choose one block of statement from many blocks of statements. Syntax:

```
    If (condition)
    {
    Block 1;
    }
    else if (condition)
    {
    block 2;
    }
    .
    .
    .
    else
    {
    Block N;
    }
```

71: Nested if Structure:

Ans: An if statement within an if statement is called nested if statement. In nested structure, the control enters into the inner if only when the outer condition is true. Syntax:

```
    If (condition)
        If (condition)
        {
            Statement…..;
        }
        Else
        {
            Statement….;
        }
    Else
    {
        Statement….;
```

}

72: Switch Syntax?

```
Ans:    switch (expression)
        {
                Case 1:
                        Statement;
                        Break;
                Case 2:
                        Statement;
                        Break;
                .
                .
                .
                Case N:
                        Statement;
                        Break;
                Default:
                        statement;
        }
```

73: Why break statement is used in switch structure?

Ans: The break statement in each case label is used to exit from switch body. If break is not used all case blocks coming after matching

74: Why should we use default label in switch statement?

Ans: The default block is used to execute a statement or set of statements when the result of expression in switch statement does not match with any case. It makes sure that switch executes properly if the selector is not in the range of case labels.

75: What is conditional operator? Write syntax of conditional operator?

Ans: Conditional operator is a decision-making structure. It can be used in place of simple “if-else” structure. It is also called ternary operator because it uses three operands. The syntax of ternary operator is a as follows:

```
        (Condition)? true-case statement : false-case statement;
```

76: Write one similarity and one difference between if-else and conditional operator?

Ans: The similarity between conditional statement and conditional operator is that both select a statement depending on a condition. The difference is that conditional operator can choose between two values but if-else can be extended to check multiple conditions to select one statement from many available statements.

77: What is loop? What are two uses of loop?

Ans: A control structure that executes a statement or set of statements repeatedly is known as loop. Loops can be used to execute a statement or number of statements for a specified number of times. Loops can also be used to access a sequence of values such as 1, 2, 3 and so on.

78: What are the program control statements used to control iterations?

Ans: The program control statements used to control iterations are while loop, do-while loop and for loop.

79: What are the important points in selecting a loop for the program?

- The for loop is best choice if loop is a simple counter controlled loop.
- The do-while loop is appropriate if the loop is sentinel controlled loop whose body is always executed at least once.
- The while and for loops are appropriate if the loop is an event-controlled loop and nothing is known about the first execution.

80: Explain the difference between pretest and posttest?

Ans: A pretest is a condition that is used to test for the completion at the top of the loop. In this type of the loop the statements inside the loop body can never execute if the terminating condition is true the first time it is tested. A posttest is a condition that is used to test for the completion of the loop at the bottom of the loop. It means that the statements in the loop body will always executed at least once.

81: What is difference between while and do-while loop?

Ans: In while loop condition comes before the body of the loop. In do-while loop condition comes after the body of the loop. If condition is false in the beginning while loop is never executed. Do-while loop is executed at least once even if condition is false in the beginning.

82: What is sentinel value?

Ans: Sentinel value is a value that is used to terminate a loop. It is used to control the loops when the number of repetitions is unknown. A sentinel value is commonly used with while and do-while loop.

83: Break statement:

Ans: The break statement is used in the body of the loop to exit from the loop. When this statement is executed the remaining iterations of the loop are skipped. The control directly moves outside the body and the statement that comes after the body is executed.

Example:

```
Int x;

For(x=1;x<=5;x++){

        Printf("Question:");

        break;

        Printf("Gateway to knowledge.");

        }

        Printf("Bye");
```

84: Continue statement:

Ans: The continue statement is used in the body of the loop to move the control to the start of the loop body. When this statement is executed in the loop body the remaining statements of current iteration are not executed. The control directly moves to the next operation.

Example:

```
Int x;

For(x=1;x<=5;x++){

        Printf("Hello World");

        Continue;

        Printf("Knowledge is power");

        }
```

85: What is nested loop?

Ans: A loop within a loop is called nested loop. In nested loop the inner loop is executed completely with each change in the value of counter variable of outer loop. Any loop can be used as inner loop of another loop.

86: What is an infinite loop?

Ans: A loop in which the ending condition never occurs is called indefinite or infinite loop. It repeats forever until the user intervenes to stop the loop.

87: Syntax of while loop:

Ans:

```
while(condition){
        Statement;
        }
```

88: Syntax of do-while loop:

Ans:

```
do{
Statement1;
Statement2;
Statement n;
}while(condition);
```

89: What are three steps that must be performed using the loop control variable?

Ans: These steps are initialization, test/condition and increment/decrement.

90: Define an array?

Ans: An array is a collection of data with same type. It is a group of consecutive memory locations with same name and type. All these memory locations have one collective name and type. The memory locations in the array are known as elements of array. The total number of elements in the array is called its length.

91: Accessing array elements:

Ans: Each element in the array is accessed with reference to its position of location in the array. This position is called index or subscript. The index of first element is 0 and index of last element is length-1. The value of the index is written in brackets along with the name of the array. Syntax: Array-Name[Index];

Example: int marks[5];

```
        Marks[0] =20;

        Marks[1] =50;

        Marks[2] =70;

        Marks[3] =80;

        Marks[4] =90;
```

92: Declaring one dimensional array:

Ans: A type of array in which all elements are arranged in the form of a list is called one-dimensional array. The process of specifying array name, length and data type is called array declaration. Syntax: Data-Type Identifier[Length];

Example: int marks[5];

93: Array initialization:

Ans: The process of assigning values to array elements at the time pf array declaration is called array initialization. Initialization process provides a list of initial values for array elements. The values are separated with commas and enclosed within braces.

Syntax: Data-Type Identifier[Length]={List of values};

Example: int marks[5]={70,54,82,96,49};

94: Searching in Arrays And types of searching:

Ans: Searching is the process of finding the required data in the array. There are basicaly two types of searching: Linear search and Binary search.

95: Linear search?

Ans: Ur is also called sequential search or serial search. It follows the following steps to search a value.

- Visit the first element of the array and compare its values with the required value.
- If the value of the array matches with the desired value the search is complete.

- If the value of array does not match move to next element and repeat same process.

Loops are frequently used to visit elements of array for searching a value. It can be applied on sorted and unsorted array.

96: Binary search:

Ans: Binary search can only be applied on a sorted array.

- It locates the middle element of the array and compares with the key value.
- If they are equal search is successful and index of middle element is returned.
- If they are not equal it reduces the search to half of the array.
- If key number is less than the middle element it searches the first half of the array. Otherwise it searches the second half of the array. The process continues until the required number is found or loop completes without successful search.

97: Sorting arrays:

Ans: Sorting is a process of arranging the values of array in a particular order. An array can be sorted in two orders: Ascending and descending.

98: Types of sorting:

Ans: Selection Sort: Selection sort is a technique that sorts an array. It selects an element in the array and moves it to its proper position. Selection sort works as follows:

- Find the minimum value in the list.
- Swap it with the value in the first position.
- Sort the remainder of the list excluding the first value.

Bubble Sort: Bubble sort is also known as exchange sort. It repeatedly visits the arrays and compares two items at a time. It swaps these two items if they are in wrong order. It continues to visit the array until no swaps are needed. Steps of bubble sort are:

- Compare adjacent elements. If the first is greater than second then swap them.
- Repeat this for each pair of adjacent elements, starting with the first two and ending with the last two. At this point last number should be greatest.
- Repeat the steps for all elements except the last one.

99: Two-Dimensional array:

Ans: Each element in 2-D array is referred with the help of two indexes. One index is used to indicate the row and the second index indicates the column of the element.

100: Declaring 2-D array:

Ans: Syntax: Data-type Identifier [Rows] [Columns];

101: Functions:

Ans: A function is a named block of code that is used to perform some specific action. The statements written in a function are executed when it is called by its name. Each function has a unique name.

102: Types of functions:

Ans: There are two types of functions:

- Built-in function: A type of function that is available as a part of language is known as built-in function or library function. These functions are stored in different header files. For example printf, scanf etc.
- User-defined function: A type of function written by the programmer is known as user-defined function. User-defined function has a unique name. A program may contain many user-defined functions.

103: Function prototype:

Ans: Function prototype provides information to compiler about the structure of the function to be used in the program. It consists of function name, function return type and number and type of parameters. It is terminated with semicolon.

104: What is a function header?

Ans: The first line of a function definition is known as function header. The function header consists of return-type, function name and parameters.

105: Function body:

Ans: The set of statements which are executed inside the function is known as function body. The body of function appears after function header. The statements are written in curly braces {}. The variable declaration and program logic are implemented in function body.

106: What is difference between function definition and declaration?

Ans: A set of statements that explains what a function does is called function definition. The function definition consists of function header and function body. It must be defined before or after the main function. It cannot be written inside the main function.

Function declaration is a model of a function. It is also known as function prototype. Function prototype provides information to compiler about the structure of the function to be used in the program. It consists of function name, function return type and number and type of parameters. It is terminated with semicolon.

107: Function call:

Ans: The statement that activates a function is known as function call. A name is called with its name. Function name is followed by necessary parameters in parentheses. If there are many parameters these are separated by commas.

108: how a function returns a value?

Ans: A function can return a single value. The return type in function declaration indicates the type of value returned by function. The key word return is used to return the value back to calling function.

109: What is difference between local variable and global variable?

Ans: A variable declared inside a function is called local variable. Local variable can be used only in the function in which it is declared. The life time of a local variable starts when the control enters the function in which it is declared. Local variable is automatically destroyed when control exits function.

A variable declared outside a function is known as global variable. Global variables can be used by all functions in the program. Global variables exit in the memory as long as the program is running. The variables are destroyed from the memory when the program terminates.

110: Explain the difference between call by reference and call by value.

Ans: The call by value makes a copy of the parameter. The formal parameters of a function are initialized by the values of actual parameters given in the function call. A parameter passing mechanism in which the address of actual parameter is passed to call function is known as pass by reference. Formal parameter becomes a second name of actual parameter.

111: Formal Parameter: Parameters in function declaration are called formal parameters.
Actual parameters: Parameters in function call are called actual parameters.

112: Recursion: A programming technique in which a function calls itself is known as recursion. A function that calls itself is called recursive function. A recursive function must have at least one termination condition that can be satisfied. Otherwise, the function will call itself indefinitely.