<u>Chapter 5</u> Programming Concept and Logics

Q1. Define Programming concept.

The thorough understanding regarding the entire process of development of programs is called programming concept. It includes the overall understanding of different types of computer programming languages, compiler, interpreter, control structures, program design tools, syntax and semantic.

Q2. What is Program and programming?

Program is defined as a set of instruction which is written **sequentially** and **logically** to do some specific task. The process of writing a program is called programming.

Q3. What is Programming Language?

The rules or syntax that a programmer use while writing a program is called programming language. The language which is used to establish a communication between human being and machine is called programming language.

Q4. Explain about Programming Statement and explain about its types.

An instruction or expression (also called code) written in high level language to do a specific task in a program is called programming statement. Each programming statement commands the computer to do some specific task such as input, output, calculation or some logical decision. One program statement in high level language may results in several instructions in machine language when the program is compiled. Programming statement may consist of keyword, variables, operators, control structures, data type, library function, user defined function etc. There are three types of programming statement. They are: -

- 1. Simple statement
- 2. Compound Statement
- 3. Control Statement or Control Structures

Simple Statement: - A Single line instruction which perform either calculation, assignment or comparison (logical testing) is called simple statement.

Example: - **a**+**b** (adding the value stored in the variable a and b)

a=5 (the value 5 is assigned into the variable a)

a>60 (testing or comparing whether the value stored into the variable a is greater than 60 or not)

Variable is defined as a name of a location in the memory of computer system where the value is stored.

Rules for naming variables

- It should always start from alphabet.
- It can be only alphabet or combination of alphabet.
- It can be the combination of alphabet and the number.
- There should not be space in between the variables.
- It should not have any symbol other than underscore (_)
- It can be any but it should be suitable according to the program.

Compound Statement: - Combination of two or more than two simple statement is called compound statement.

Example: -**sum=a+b** (at first adding the value stored into the variable a and b and then assigning the result into the variable sum)

In the above statement there are two types of working variable one is input variable and another is output variable.

Control Statement: - It is defined as the basic building block of high level programming language. It affects the flow of execution through the program. There are three types of control statement. They are: -

- Sequence
- Selection
- Iteration(Loop)

Q5. Differentiate between Program and Software

Program	Software	
1. The set of instructions which are written	1. Software is a collection of program that instruct computer t	
sequentially and logically to do some specific task	do some specific task.	
is called Program.		
2. A program is independent of its own.	2. Collection of program are require to develop software.	
3. A Program is created by Programmer.	3. Software is created by group of programmers as team work.	
4. A Program is not generally licensed to buy.	4. Software is generally licensed under a company.	
5. A Program cannot be categorized according to need	5. The software can be categorized according to various needs	
and uses.	and uses. For Example:-Application software, System	
	software etc.	
6. Example:- A program to calculate simple interest.	6. Example:- Microsoft Office. Photoshop, Window10 etc.	

Q6. What are the Qualities of Good Program?

A program can be considered as a good program if it contains the following qualities:

- a. It should be Correct.
- b. It should give desired output.
- c. It should be easy to understand.
- d. It should be easy to maintain and update.
- e. It should be flexible.
- f. It should have easier user interface.
- g. It should be portable.
- h. It should be efficient in terms of processing and memory utilization.

Q7. Explain about different types of programming languages and write down what are the advantages and disadvantages of different types of programming language?

There are two types of programming language which can be categorized into the following ways: -

1. Low Level Language

-Machine Level Language or First Generation Language(1GL) -Assembly Level Language or Second Generation Language(2GL)

2. High level Language

-Procedural Oriented Language or Third Generation Language(3GL) -Problem Oriented Language or Fourth Generation Language(4GL) -Natural Language or Fifth Generation Language(5GL)

Low Level Language

Low Level Languages are difficult and time consuming to develop the program. It is a machine dependent language. Programmers require detailed knowledge of hardware architecture to use this language. It is used for developing hardware oriented software.

Machine language: - The language which is called the language of CPU and which is written in binary code (composed of 0 and 1) is called Machine language. In this language computer can understand every instruction only in the form of 0 and 1. All the valid computer operation are performed in terms of binary operation

Example: - If we want to write an instruction "Add 2 and 3 "then we have to write "01100001 (ADD) 00000010 (2) 00000011(3)"

Advantage of machine language

1) The program written in machine language does not need any translation.

2) Though it is very difficult to write program in machine language but the program written in machine language runs faster than the program written in any other languages.

Disadvantage of machine language

- 1) Machine language is machine dependent. The program written for one processor cannot be run in another processor i.e. a machine code of intel processor doesn't work for Motorola processor.
- 2) Machine language has to be written in the form of 1 and 0. So it is very difficult to do program in machine language.
- 3) Chances of error is high while writing program in machine language.
- 4) It is very difficult to debug the program written in machine language.

Assembly language: - The language in which program is written in short codes (**mnemonics code**) like ADD, MUL or SUB instead of writing instruction in binary number (0 and 1) is called assembly language. It is also called low level language. Assembler converts the assembly language to the machine language to make it understandable by computer hardware because computer can understand the instruction written only in binary form. It is not user friendly language. This language is also processor dependent language. A Program written for Intel processor does not work in Motorola processor. Computer viruses, device driver, traffic light system etc. are mostly written in assembly language.

Assembler: - A computer program which converts the "assembly language" into machine language instruction is known as Assembler.

Advantage of assembly language

- 1) Assembly language is easier to understand as compared to machine language.
- 2) It is easier to modify as compared to Machine language.
- 3) Chances of error is less than machine language.
- 4) In Assembly language program execution time is faster than program written in high level languages.
- 5) It is useful for developing hardware oriented system software like device driver, remote control system, robotics, traffic light system etc.

Disadvantage of Assembly language

- 1) It is a processor or Machine dependent language.
- 2) Since it is a Machine dependent language it requires the knowledge of hardware architecture also while writing program in assembly language.

High Level Language: - The languages which are quite similar to written English and are therefore very easier to use in compare to machine language and assembly language is called high level language. Program written in high level language is translated into machine language by a program is called compiler or interpreter (first this program checks the program's syntax if it is free from errors translator translates into an equivalent code). High level languages are machine independent language i.e. the program written for Intel processor easily run on Motorola processor. It is very easy to understand. It is user friendly language. It is a problem oriented language.

Examples of High Level Language are: - C, C++, FORTRAN, QBasic, Python, PHP, VB.Net etc.

High-Level Language can be further categorized as:

- i. Procedural Oriented Language (3GL)
- ii. Problem Oriented Language(4GL)

Procedural Oriented Language(3GL): - This language uses simple English and mathematical statements to develop a program. It includes languages such as: - C, QBasic, FORTRAN etc. These languages are designed to express the logic and the procedure of a problem. Because of their flexibility, procedural languages are able to solve a variety of problems.

Advantages of Procedural Oriented Language

- i. It is easier to develop and debug program compared to Low Level Language.
- ii. More advanced and user friendly software can be developed.

- iii. It is a machine independent language. So it is portable (i.e. program written in one processor easily run on another processor).
- iv. Programmer do not require knowledge of hardware architecture.

Disadvantage of Procedural Oriented Language

- i. The program written in high level language in less efficient as compared to the program written in assembly and machine language because it requires more memory and more time to execute.
- ii. The task related to the computer hardware can be programmed or handled only by either machine or assembly language so for these purposes high level language cannot be used.
- iii. The execution speed of procedural oriented language is slower than Low Level Language because it cannot be executed directly, it need to be translated into machine language code (Object Code).

Problem Oriented Language(4GL): - The language which works at a higher level than procedural oriented languages such as COBOL, Pascal, C etc. is called problem oriented language or 4GL. It is a non-procedural language. It is a result oriented language and include database Query Languages. 4GL permits programs to be constructed using short English like easily understood and easily remembered commands. 4GL programs are also needed to be translated either by compiler or interpreter. 4GL are dedicated for some particular application development.

Example of 4GL are: - Visual Basic, PHP, C#, VB.Net etc.

Advantages of Problem Oriented Language(4GL)

- i. It is even easier to develop and debug a program than a Procedural Oriented Language.
- ii. More advance and user friendly software can be developed.
- iii. Web based software can be developed.
- iv. It is a machine independent language.
- v. Program developed in 4th generation language is more portable than those developed in 3rd generation languages.

Disadvantages of Problem Oriented Language (4GL)

- i. Needs longer time to convert into machine language than other languages.
- ii. Execution speed is slower than other languages.
- iii. It requires more resources (Memory, CPU) than Procedural Oriented Language.

Q8. What do you mean by Language Translator? Name the different types of Language Translator.

Language Translator is a program that translates instruction(Program) written in assembly or high level language into machine language. The program is developed in assembly or high level language for the convenience of the programmer, but the computer doesn't understand these statements directly. So, the language translator is used which converts those statements in a computer executable format. There are three types of Language Translator. They are: -

Assembler

- AssemblerCompiler
- Interpreter

Q9. Differentiate between Compiler and Interpreter with example.

Compiler	Interpreter
1. It translates a program written in high level language(Source Code)into machine language(Object Code) at once.	1. It translates a program written in high level language(Source Code) into machine language(Object Code) line by line.
2. It finds the syntax errors after compiling the whole program	2. It finds the syntax errors after translating each line of the program.
3. The compiling process is faster than interpreter.	3. The interpreting process is slower than compiler.
4. It creates the object code after compiling the whole program	4. It creates the object code after translating each line of the program.
5. Compiler based program is difficult to code and debug.	5. Interpreter based program is easy to code and debug.
6. New programming languages don't use compiler	6. New programming languages use interpreter

Q10. Define Syntax and Semantics

- **Syntax:** The rules of programming languages according to which statement of a program are written is called syntax. It is defined as a set of rules that define the structure of the programming language. The rules that programmer fallow while writing a program in any programming language is called Syntax.
- **Semantics:** The logic, concept or idea of the program is called semantics. In another word the meaning of the program is called semantics.

Q11. Define Error. What are the different types of error?

Errors are the mistake in the program. The error can be in syntax or logic of the program. The process of detecting and correcting the error is known as debugging.

Different types of errors are: -

- Syntax Error
- Logical Error (Semantic Error)
- Syntax Error: -A Syntax error is an error in the syntax or format of the program. It happens when there is a mistake in the syntax of the programming language. For example: - In C language whenever we do not use semicolon after writing the simple statement causes a syntax error. It is detected by a translator at the translation process.
- **Logical Error:** A Logical error is an error in a logic of the program. For Example: In C to calculate the area of rectangle the formula is 1*b but if we write the formula 1+b then it will produce a logical error because it is syntactically correct but it produces undesired output. It is not detected by the translator.

Q12.Why it is very difficult to find logical error in comparison to syntactical error?

It is very difficult to find out the logical error in comparison to syntactical error because syntax error violates the rule of writing the programming statement and is detected by compiler at the time of compiling by showing total error occur in computer screen but the logical error is not detected at the time of compiling, after it get executed when we get the wrong output, seeing that wrong output we have to find out the error in logic. On other hand there is no any satisfactory method has been found through which we can detect the logical error.

For example: - if we have to find out the sum of two number say x and y but we subtract the number, then it is called the logical error. Actually the computer assumes that we are going to subtract but actually we want to find the sum of two numbers so there exist an error. So, instead of writing Z=X+Y if we write Z=X-Y, the computer will not show the error, but the result which we want we will not get.

Q13. Differentiate between Syntax Error and Semantics Error.

Syntax Error	Logical Error
1. A Syntax error is an error in the syntax or format of	1. A Logical error is an error in a logic of the program.
the program.	
2. It is identified by the translator (Compiler or	2. It is not identified by the translator (Compiler or
Interpreter) during translation.	Interpreter) during translation.
3. If a program contains a syntax error, it will not	3. If a program contains a logical error, it produce output
produce output.	but it will not be accurate.
4. There is a method to find the syntax error	4. There is no method to find the logical error.
5. Normal programmer can debug syntax error.	5. Experience programmer is required to debug logical
	error.
6. It is cheaper to debug syntax error.	6. It is expensive to debug logical error.
7. It is easier and no time consuming to debug syntax	7. It is complex and time consuming to debug logical
error.	error.

Q14. Define data type? Explain each of them with suitable example?

Ans: - A data type defines a set of values that a variable can store along with a set of operations that can be performed on that variable.

There are mainly four data types. They are as follows: -

- a) Integers
- b) Characters
- c) Real
- d) Boolean (logical)
- a) **Integers:** Integer is a basic data type which contains discrete numbers (both +ve and -ve) like 2,-5 etc. The range of value for integer differs from one programming language to another. An integer data-type cannot have fraction or exponent etc
- b) **Characters:** A data type which contain any printable alpha-numeric character plus other special character like #, @, % etc are called character data type.

Example: -

Alphabets: a to z, A to Z Numbers: - 0 to 9

- c) **Real:** The data type which contain any numeric representation is called Real data type. The value may be signed or unsigned, fractions, exponential as well. Example: 0, 5.0, 4.7234, 3.0e⁻⁴ etc. are real numbers
- d) **Boolean:** The data type which only take one of two possible values at any one time is called Boolean data type. The two values are true and false. It is useful for checking the condition inside the program.

Q15. Define Variable and Constant and Keyword

A variable is an identifier which value may change during the execution of the program. A variable is defined as a name of a space in a memory of the computer where we store the value. For example: - in a statement sum=a+b, a, b and sum is a variable.

Constant is an identifier which value remains fixed or unchanged throughout the program. For example: - in a statement area= 3.14*r*r, 3.14 is the constant.

Keyword: - The keyword is a reserved word that has standard and predefined meanings in C. we cannot use keyword for any other purpose other than as a predefined task in a C. Its meaning has already been defined by the compiler. There are 32 keywords in C language. They are: -

Auto	Const	Double	Float	Int	Short	Struct	Insigned
Break	Continue	Else	For	Long	Signed	Switch	Void
Case	Default	Enunm	Goto	Register	Sizeof	Typedef	Volatile
Char	Do	Extern	If	Return	Static	Union	while

Q16. Define Operand, Operator and Operation

Operand: - In computer programming, an operand is defined as an item that takes part in operation. In another word operand is a term used to explain any object that is capable of being manipulated. **For example:-**In statement X=a+b, a and b are operand.

Operator: - An operator is a symbol that is used to define the operations to be performed on operands. For example:-In statement X=a+b, + is operator. The different types of operator are:-

- Arithmetic Operator (+, -, *, /)
- Relational Operator (>, <, >=, <=)
- Logical Operator (Logical AND, Logical OR, Logical NOT)

Operation: - An operation is defined as the result of operator on operand. In another word, an action upon a given data is called operation. The combination of operator and operand is called operation. **For example:** - In Statement X=a+b, a+b is called operation.

Q17. Define Program Design Tools. What are the different types of program design tools? Explain with example.

The tools that is used to design a program before it is actually developed is called program design tools. It is used by the designers.

The name of the different program design tools are as follows: -

- 1. Algorithm
- 2. Flowchart
- 3. Pseudo code

Algorithm: - An algorithm is a sequence of instructions or step by step instruction to find the solution of a problem. Algorithm should be simple and clear. It must be to the point and should lead to the solution of the problem in a finite number of steps.

In other words, the set of rules that define how a particular problem can be solved in finite number of steps is known as algorithm. An algorithm is composed of a finite set of steps, each of which may require one or more operations. Each operation must be definite, effective and finite. The success of a program depends upon the algorithm. Therefore, the logic of the problem must be clearly expressed in an algorithm.

Example:

Algorithm for adding the two numbers

Step1: Start the program Step2: Input x, y Step4: Sum=x+y Step5: Print Sum Step6: Stop the program

Flowchart:

A flowchart is a pictorial representation of an algorithm. We can also define it as a program planning tool for organizing the sequence of steps necessary to solve a problem which is shown in terms of symbols. Flowchart use symbols that has geometrical shapes to indicate the different operation and these symbols connected by flow lines which indicate the order of execution of the various activities. It is used by the programmers to develop the program and system designer to represent the overall system while analyzing and developing the system.

Advantage of flowchart

- > Once the flowchart is prepared it is very easy to do the coding by looking at the flowchart.
- > It is easier for a programmer to understand the logic of the program through the flowchart.
- > It provides good documentation for organization.
- > It is useful in debugging the program.
- > It provides valuable information which cannot be expressed by thousands of words.
- ▶ It is user friendly. It can be more easily understood.
- ▶ It expresses clearly the logic of a given procedure.
- ▶ It is used as working models in the design of a new programs and system.

Disadvantage of flowchart

- Preparing flowchart is a time consuming process
- > If there is a change in logic again we have to develop the new flowchart
- Flowchart are quite costly to produce and difficult to use and manage.
- Complex and detailed charts can be laborious to plan and draw
- > The translation of flowchart into computer programs is not easy.
- Most software developers believe that flowcharts are a waste of time and that these slow down the process of software development and they do not start the software development with flowchart.

Example: Flowchart to calculate the sum of two numbers



Pseudo- Code:

The general meaning of pseudocode is false code or fake code. Before we write a real program, we write a program that look like a code on the basis of algorithm and flowchart, which may not be syntactically correct but looks like program code is called pseudocode. The instruction of pseudocode is written by using English phrase and mathematical expression. It has no hard and fast rules for writing instruction in pseudocode but the instruction are closer to high level language instructions. The pseudocode designers should have basic knowledge about high level language before writing pseudo-code. It is also independent of any programming language. **Example:**

Pseudo code for calculating area of rectangle

Input length, breadth Calculate area = length*breadth Output area

Q18. What are the symbols used in flowchart? Mention their name with symbol, shape purpose and description

There are various flowchart symbols that carry different messages and are used for different purposes. These symbols are shown below:

Symbol	Shape	Purpose	Description
	Oval	Start /end	This symbol indicates the beginning or end of the program. every flowchart begins with a start symbol and ends with a stop symbol
	Parallelogram	Input/output	This symbol indicates an input or output operation. It will appear at any point where a data is to be entered into the

	Postangla	Dragoog	computer or at times when result is to be obtained from the computer
	Kettangie	Flocess	calculation or manipulation .i.e processing of data
\bigcirc	Diamond	Decision/condition	It is used to check logical conditions. It has one entry and two exit point which are for true or false
	Line with direction	Flow-line	It is used to indicate direction of flow of information.
	Circle	Connector	Connectors are used for the long flowchart that does not fit in the same page. In this condition the flowchart can be
A A	End of page Beginning of page		broken into parts and an arrow is drawn into a connector symbol containing a unique number or letter
	Large bracket with line	Comment box	Comment box can be used to indicate any comments for better documentation, clarity and in case of explanatory notes.

Q19. Differentiate between Algorithm and Flowchart with example.

Algorithm	Flowchart
1. It is a finite set of step by step instruction of a program.	 It is a diagrammatic representation of a program.
2. It takes less time to write the steps of instruction.	2. It takes more time because we need to draw the symbol.
3. English language is used to write the algorithm, so it is easy to understand	3. To understand the flowchart we should know the meaning of the symbols used in flowchart.
4. No need of special knowledge to write an algorithm,	4. Need special knowledge to draw flowchart.
5. Maintaining the algorithm is easy.	5. Maintaining the flowchart is difficult.
6. It is difficult to understand the program flow	6. It is easy to understand the program flow at a
at a glance.	glance.
7. In algorithm there are steps.	7. In flowchart there are symbol.

Q20. What are the different types of Flowchart? Explain.

There are two different types of flowchart. They are: -

- System Flowchart
- Program Flowchart

System Flowchart: - A System flowchart is a diagram that shows a broad overview of the data flow and sequence of operation in a system. It is design by the system designer while analyzing and designing the system. It shows the different elements and characteristics of the system or organization diagrammatically. It focus on the overall system, not the solution of the problem. **The different symbol used in system flowchart are:** -



Program Flowchart: - Program flowchart is a pictorial representation of the steps involved in the procedure of a program. It is designed by the programmer while making a new program. It also shows the logical sequence in which the steps are performed. It is a detailed diagram showing how individual processing steps will be performed within the computer to convert input data into the output. It shows the different elements and characteristics of a program diagrammatically. It focus on the solution of the problem not the overall system.

The different symbols used in program flowcharts are: -



Q21. Differentiate between Program Flowchart and System Flowchart?

System Flowchart		Program Flowchart
1.	It gives the board overview of the overall	1. It gives a specific view of the control flow of a
	system.	program.
2.	It shows the data flow of a system.	2. It shows the instruction flow of a program.
3.	The System flowchart is designed by a system	3. The Program flowchart is designed by a
	designer.	programmer.
4.	It focus on the overall system.	4. It focus on the solution of the problem.
5.	It is difficult to design the system flowchart	5. It is easy to design the program flowchart
	because it has complex symbol	because it has simple symbol.
6.	It shows the elements and characteristics of	6. It shows the elements and characteristics of the
	the system diagrammatically.	program diagrammatically.

Q22. Describe about different coding techniques.

- The different types of coding techniques are:
 - a) Absolute binary
 - b) BCD (Binary coded decimal)
 - c) ASCII (American standard code for information interchange)
 - d) EBCDIC (Extended binary coded decimal interchange code)
 - e) Unicode
- a) **Absolute binary:** In absolute binary method 0 is placed before the binary number to represent positive number and 1 is placed before the binary number to represent negative number. The most significant bit in binary number denotes the sign bit and the rest bits represent the actual number. The binary number is expressed in 8,16,32,64 etc bit format.
- b) **BCD** (**Binary coded decimal**):- It is a simple system for converting decimal numbers into binary form where each decimal number is converted separately into binary and placed spaces in between numbers. In BCD each decimal digit occupies 4 bit. For example, the decimal number 24 can be represented in BCD as (0010 0100)₂
- c) ASCII (American Standard Code for Information Interchange): ASCII is a standard coding system that assigns numeric values to letter, numbers, punctuation marks and control characters to achieve compatibility among different hardware and peripherals. ASCII was developed in 1968.ASCII has two sets: Standard ASCII (7 bits code, 128 characters) and extended ASCII (8 bits code, 256 characters). Most system use 8 bit extended ASCII to represent foreign language character and other graphical symbols. In ASCII, each character is represented by a unique integer value from 0 to 255. The value 0 to 31 is used for non-printing control characters and the range from 32 to 127 is used to represent the letters of the alphabet and common punctuation symbols. For example: ASCII code for capital letter A is 65, ASCII code for * is 42 etc. Since ASCII code uses 8 bits, each character represented in ASCII code occupies 1 bytes storage space in computer.

- d) **EBCDIC** (**Extended Binary Coded Decimal Interchange Code**):-It is an 8-bit code system commonly used on large IBM mainframe computers, most IBM minicomputers and computers from many other manufactures. It allows 256 characters to be represented in computer. In this code placement of the letters of the alphabet is discontinuous and there is no direct character to character match when converting from EBCDIC to ASCII and vice versa.
- e) Unicode: It is a 16-bit character code, defined by the Unicode Consortium and International Standard Organization (ISO) that supports up to 65,536 characters. It allows all the characters and symbols in any language in the world to be represented by a single code. For example, the Chinese language has almost 10000 characters which can be represented by Unicode only. If Unicode is universally adopted, then it will make multilingual software much easier to write and maintain. Since Unicode uses 16 bits, each character represented in Unicode occupies 2 bytes storage space in computer. This coding system has been developed to overcome the drawback of ASCII code that supports only 256 different characters which is sufficient only for English language but not for all the languages like Chinese, Japanese etc which has more than 256 characters.