

Programming Methodology

TOPIC – 1

General Concepts Of Programming

Very Short Answer Type Questions (1 mark each)

Question 1:

On which step programming methodology used ?

Answer:

Before writing program. **1**

Question 2:

What do you mean by modular approach ?

Answer:

Breaking of a problem into sub-problems. **1**

Question 3:

Which characteristics of programming methodology refers to the overall readability of the programme ?

Answer:

Clarity. **1**

Question 4:

Are comments executed ?

Answer:

Comments are not executed **1**

Question 5:

Do comments increase execution time ?

Answer:

No, comments do not increase execution time. **1**

Question 6:

How many types of comments ?

Answer:

Comments are of two types : single line and multiple line. **1**

Question 7:

Why do we use comments ?

Answer:

Comments helps the reader to understand the programme easily. **1**

Question 8:

Which characteristics refer to insertion of extra blank spaces before declaration ?

Answer:

Indentation **1**

Short Answer Type Questions-I (2 marks each)

Question 1:

Define Clarity of source code.

Answer:

The extent to which inherent language features support source code that is readable and understandable and that clearly reflects the underlying logical structure of the program.

Question 2:

Explain clarity of expression

Answer: Clarity refers to the overall readability of the program, with emphasis on program logic. It allows other programmers to follow the program, if the program is written clearly. It also makes the programmer to grasp his or her own program logic even after a long period of time.

Question 3:

Give an example of clarity of expression.

Answer:

Suppose we wish to write a program to calculate the area of a circle. We require two variables; radius and area. Instead of writing r and a to represent the radius and area respectively, we use radius and area to represent themselves. It increases the readability of program, Similarly, the constant can be written as pi instead of denoting by p.

Question 4:

Explain types of comments.

Answer:

There are two types of comments in a C++ program; single line comment and multi-line comment. Single line comment begin with double slash (//) and end in the same line. We can write a program statement and a comment in same line. Whatever written after the // is ignored by the compiler.

Question 5:

Write any two characteristic of comments.

Answer:

Comments are not executed in a program. Since the compiler ignores comments, there will be no increase in the file size and execution time.

Question 6:

Write any two purpose of comments.

Answer:

1. Comments help the reader to understand the program easily.
2. When a program consists of several functions, comments can be added to each function to indicate the purpose of the function.

Question 7:

Define indentation.

Answer:

Indentation refers to the insertion of extra blank , spaces before declarations and statements. It enhances the readability of programs.

Long Answer Type Questions (4 marks each)

Question 1:

Define clarity and simplicity of expression with example.

Answer:

Clarity and Simplicity of Expression: In the early days most of the computer programs were written to carry out difficult and complex calculations. It is not so any more. However, you may still have to write programs involving complex calculations.

It is advised that if any expression becomes very big or very complex then you must write the same in two steps rather than doing it in one single step. This will make the expression simple and any programmer will be more accurate in writing simple expressions.

Simple expressions are also easily readable and can be understood by others without making any extra efforts. For example, a Statement like :

$A = \text{square root}((x^y/d) + (c+z)^2 - (u*v))/L$ may be written in two steps as :

$$A1 = x^y/d$$

$$A2 = (c+z)^2$$

$$A3 = u * v$$

$$A = \text{square root}(A1 + A2 - A3)/L$$

Topic – 2

Documentation And Programming Maintenance

Very Short Answer Type Questions (1 mark each)

Question 1:

How many types of documentation are there ?

Answer:

- (a) internal documentation
- (b) external documentation

Question 2:

Which documents helps in programme maintainance ?

Answer:

internal documentation

Question 3:

Write the alternate name of error.

Answer:

Bugs

Question 4:

What is debugging ?

Answer:

The process of correcting the error.

Short Answer Type Questions-I (2 marks each)

Question 1:

Explain documentation

Answer:

All programs need not to be short and simple. Complex programs can be better understood if additional information is provided. Documentation is a method of providing useful information about a program. It consists of written descriptions, specification and development of the program.

Question 2:

Explain programme maintenance.

Answer:

Changes in requirements force to modify existing programs. Program maintenance is the

modification of a program to take care of changing requirements or any errors found after the program is put to use.

Question 3:

What do you mean by running and debugging of programme.

Answer:

Running and Debugging Programs : Once a program has been written, it has to be tested for its correctness. During this process, we may find errors. In computer terminology, errors are known as bugs, and the process of correcting the errors is known as debugging.

Question 4:

Define syntax error.

Answer:

The grammatical rules of a language are referred to as syntax of that language. If we do not strictly follow the syntax of programming language at the time of writing a program, syntax errors are bound to occur.

Long Answer Type Questions (4 marks each)

Question 1:

Write the factors on which documentation depend.

Answer:

1. A program must have meaningful constant and variable names.
2. A program should include specification of the problem.
3. The initial lines of comment should contain programmer's name, date, brief description of the problem, and the purpose of the program.
4. Other comments include the input by form and type, output by form and type, data structures used, main function and other functions used, and assumptions.
5. The purpose of every function and their inter-relationships must be described with comments.
6. Comments are to be inserted in the program wherever necessary for clarity. The purpose of such comments is not to tell how or why something is done, but what is being done.

Question 2:

Explain program maintenance with reference to organization.

Answer:

Program maintenance is an important duty of programmers. It involves all steps from problem definition to program preparation. In certain organizations, programmers can do nothing but maintain the programs.

At the initial stage of computerization, an organization's programming effort mostly goes into the development of new applications. As the number of installed programs grows, the

programming effort shifts from program development to program maintenance, in fact, in many organizations, more than half of the programming effort is on program maintenance. An estimate shows that the cost of program maintenance is twice larger than that of initial development.

Question 3:

Explain run time error with example.

Answer:

Run-Time Errors : A program's syntax may be correct- It may be compiled and linked successfully. But it may not be executed successfully because of run-time errors. Such errors are found at the time of executing the program.

Let us consider the following program segment

```
n = 9;
while (n > 1)
{
    n-1
    m = m / (n -1);
}
```

When the value of n becomes 2, the expression $n > 1$ is true.

Hence, the statement $n - 1$ decreases the value of n by 1. When the value of n is 1, the divisor is 0 which generates a run-time error. Run-time errors include finding square root of a negative number, stack overflow, and null pointer assignment.

Question 4:

Explain logical error.

Answer:

A logic error is a defect in the program that causes it to produce an incorrect result, but one that is not so blatant as to be detectable as a runtime error. (A logic error in one part of the program might eventually trigger a runtime error in some other part of the program, but those are separate errors.) An example would be a function that is supposed to return the larger of its two arguments but in fact returns the smaller:

```
def larger(m, n):
    if (m > n):
        return n
    return m
```

Question 5:

Write all steps of program solving methodology. [CBSE Text Book]

Answer:

There are following seven steps of program solving methodology:

1. **Problem Definition:** Computer programs are written to solve problems posed by humankind. Prior to writing a program, one has to understand a description of the problem to solve. This description may be very precise or vague, but nevertheless, it is necessary /present. Once programmer is sure of what the problem entails, he must write down a list of specifications. Specifications are precise definitions of what the program must do.
2. **Problem Analysis:** In this step, the problem has to be fragmented into smaller and manageable parts. The original problem has to be analyzed and divided into a number of sub-problems as these sub-problems are easier to solve and their solutions would become the components of the final problem. Each sub-problem is divided into further smaller ones and this fragmentation has to be continued to achieve simple solutions. The use of modular programming is to get proper solution.
3. **Designing the Problem :** Designing the problem can be expressed in the form of:
 - (i) Algorithm
 - (ii) Flowchart

(i) Algorithm: An algorithm is a set of instructions that describe a method for solving a problem. It is normally given in mixture of computer code and English Language. This is often called “Pseudocode”.

(ii) Flowchart : The algorithm is represented in the form of a diagram with action boxes linked by lines showing the order in which they are executed. This is known as ‘the flow of control’. It is the diagrammatic representation of an algorithm.
4. **Coding :** The process of translating the algorithm into syntax of a given language is known as ‘coding’. Since algorithm cannot be executed directly by the computer, it has to be translated into a programming language.
5. **Program Testing and Debugging :** Program testing means running the program, executing all its instructions and testing the logic by entering sample data in order to check the output. Debugging is the process of finding and correcting the errors in the program code.
6. **Documentation :** The documentation includes the problem definition, design documents, a description of the test perform, a history of the program development and its different versions and a user’s manual. Such a manual is designed for a native user and illustrates the preparation of input data, running the program and obtaining and interpreting the results.
7. **Program Maintenance :** It is not directly part of the original implementation process, but needs special emphasis. All activities that occur after a program operation are part of the program maintenance. Many large programs have long life span that often exceed the lifetime of the hardware they run on. Maintenance is an important part of the life cycle of a program.

Question 6:

What is run-time error, logical error and syntax error?

Answer:

Syntax error : The errors which are traced by the compiler during compilation, due to wrong grammar for the language used in the program, are called syntax errors.

For example, `cin<<a;` // instead of extraction operator insertion operator is used.

Run time Error : The errors encountered during execution of the program, due to unexpected input or output are called run-time error.

For example – `a=n/0;` // division by zero

Logical Error : These errors are encountered when the program does not give the desired output, due to wrong logic of the program.

For example : `remainder = a+b` // instead of using % operator + operator is used.