



ASSIGNMENT-01

Programming Fundamental Theory



ABDUL MOIZ CHISHTI

SEC-A

SE-20F-022

Question 1:

What are the different levels of programming? List some advantages and disadvantages of each type

Answer:

In today's world there are 3 types of programming languages which are as followed:

- **Machine Language**
- **Assembly Language**
- **High-Level Language**

Machine Language:

The machine language is the fundamental language of the computer's processor and also called as Low level Language. It was the first type of programming language to be developed. It consists of combination of 0's and 1's that represent high and low electrical voltage. All programs are converted into machine language before they can be executed.

Advantages:

Machine Language makes fast and efficient use of the computer. It requires no translator to translate the code. It is directly understood by the computer.

Disadvantages:

All operation codes have to be remembered. All memory addresses have to be remembered. It is very difficult to amend or to find an error in a program written in the machine language.

Assembly Language:

An assembly language is a low level language that is similar to the machine language. It uses symbolic operation code to represent the machine operation code instead of 0's and 1's. It was developed to overcome some of the many inconveniences of machine languages. Mostly assembly language is used to help in compiler orientations.

Advantages:

Assembly language is easier to use and understand as compared to machine language. It is easy to locate and correct errors. It can be easily modified.

Disadvantages:

Like the machine language, it is also machine dependent. Since it is machine dependent, the programmer also needs to understand the hardware.

High-Level Languages:

High level languages are basically the programming languages which are easier to learn. It uses English like statements. The purpose of developing high-level languages was to enable people to write programs easily, in their own native language environment(English). Each instruction in the high-level language is translated into many machine language instructions that the computer understand.

Advantages:

High-level languages are user-friendly languages. They are similar to English and uses English vocabulary and well-known symbols. They are easier to learn and maintain. The language is independent of the machine on which it is used I.e. programs developed in a high-level language can be run on any computer text.

Disadvantages:

A high-level language has to be translated into the machine language by a translator, which takes up time. The object code generated by a translator might be inefficient compared to an equivalent assembly language program.

Question 2:

Write a note on evolution of programming languages; list some programming languages of each generation.

Answer:

Evolution of Programming Languages:

Programming languages have been developed over the year in a phased manner. Each phase of developed has made the programming language more user-friendly, easier to use and more powerful. Each phase of improved made in the development of the programming languages can be referred to as a generation. The programming language in terms of their performance reliability and robustness can be grouped into five different generations which are as followed:

- 1. First Generation Languages (1GL)**
- 2. Second Generation Languages (2GL)**
- 3. Third Generation Languages (3GL)**
- 4. Fourth Generation Languages (4GL)**
- 5. Fifth Generation Languages (5GL)**

First Generation Language (Machine Language):

The first generation programming language is also called low-level programming language because they were used to program the computer system at a very low level of abstraction.

Second Generation Language (Assembly Language):

The second generation programming language also belongs to the category of low-level- programming language. In the assembly language, symbolic names are used to represent the opcode and the operand part of the instruction.

Third Generation Languages (High-level Languages):

The third generation programming languages were designed to overcome the various limitations of the first and second generation programming languages.

Fourth Generation Language (Very High-level Language):

The languages of this generation were considered as very high-level programming languages required a lot of time and effort that affected the productivity of a programmer.

Fifth Generation Languages (Artificial Intelligence Language):

The programming languages of this generation mainly focus on constraint programming. The major fields in which the fifth generation programming language are employed are Artificial Intelligence and Artificial Neural Networks.

Question 3:

What are the different paradigms of programming languages? List some advantages and disadvantages of Procedural Programming Languages and Object-Oriented Programming Languages.

Answer:

Programming paradigms:

Programming paradigms are a way to classify programming languages based on their features. Some common programming paradigms include:

- Imperative
- Procedural
- Object-oriented
- Declarative
- Functional
- Logic
- Mathematical

Procedural programming:

Procedural programming is a programming, derived from structured programming, based on the concept of the procedure call.

Advantages:

Procedural programming is excellent for general-purpose programming. The coded simplicity along with ease of implementation of compilers and interpreters. A large variety of books and online course material available on tested algorithms, making it easier to learn along the way. We can access the same code without repeating it at various points on the software program.

Disadvantages:

The code re-usability feature is not present in the procedural programming. We have to write the same programming code to many times. We can not perform encapsulation, inheritance etc in the procedural programming.

Object-Oriented Programming:

Object-oriented programming is a programming paradigm based on the concept of “objects”, which can contain data in the form of fields and code in the form of procedures.

Advantages:

In the Object-oriented programming we can eliminate code and extend the use of existing classes through inheritance. It can be reused and is very easy to understand. It models the real world well. The principle

of data hiding helps the programmer to build secure programs. In the object-oriented programming software complexity can be easily managed.

Disadvantages:

The thought process involved in object oriented programming may not be natural for some people, and it can take time to get used to it. Object-oriented programs typically involve more lines of code than procedural programs. Object-oriented programs are typically slower than the procedure-based programs, as they typically require more instructions to be executed.

Question 4:

Differentiate between a compiler and an interpreter.

Answer:

Interpreter	Compiler
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code
Interpreter usually takes less amount of time to analyze the source code. However the overall execution time is comparatively slower than Compiler.	Compilers usually take a large amount of time to analyze the source code. However, the overall execution time is comparatively faster than the interpreters.
No intermediate object code is generated, hence are memory efficient.	Generates intermediate object code which further requires linking, hence requires more memory.
Programming languages like JavaScript, Python, Ruby use interpreters.	Programming languages like C, C++, java use compilers.

Question 5:

What are the rules required to be followed while naming a variable in Python?

Answer:

Variable Names:

A variable can have a short name (like x and y) or a more descriptive name (age, surname, total_volume).

Rules For Naming a Variable in Python:

Following are the rules which are required to be followed while naming a variable in python:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)