SECTION 5

INTRODUCTION TO PROGRAMMING

Fundamentals of Programming

Algorithm

An algorithm is a step-by-step procedure or set of rules for solving a problem or performing a task, especially by a computer.

The word “algorithm” comes from the name of the Persian scholar, astronomer, geographer, and mathematician Muḥammad ibn Mūsā al-Khwārizmī (circa 780–850 CE). He wrote a book on mathematics and Hindu-Arabic numerals, which was translated into Latin in the 12th century.

Pseudo-Code

Pseudo-code is a way of writing algorithms using a mix of natural language and programming concepts. It is not written in any specific programming language but follows a structured format that resembles actual code.

Key Features of Pseudo-code:

- Easy to Read: It focuses on logic rather than syntax.
- Language-Independent: It can be translated into any programming language.
- Uses Simple Keywords: Common keywords include IF, ELSE, WHILE, FOR, PRINT, etc.
- Helps in Planning: It is often used to outline an algorithm before writing actual code.

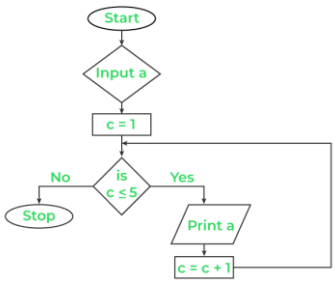
Example:

```
Begin
    Ask the age
    If the age is 18 or older
        Show a message saying: "You are eligible to vote!"
    Else
        Show a message saying: "Sorry, you must be at least 18 to vote."
End
```

Flowchart

A flowchart is a visual representation of a process or algorithm using different symbols to show the flow of steps. It helps in understanding, designing, and debugging a process before writing actual code.

Example:



Compiled and Interpreted Languages

What is compilation?

Compilation is the process of converting source code, written in a high-level programming language (like C, C++, or Java), into machine code or an intermediate code that a computer can execute directly. This process is done by a program called "compiler".

Programs are written in high-level languages (such as C, Java, or Python) because they are easier for humans to understand and write. Computers, however, understand only machine code (binary code), which consists of zeros and ones. A CPU executes instructions written in machine code.

The compiler takes the human-readable source code and translates it into a form that the computer can understand—either directly into machine code or into intermediate code (like bytecode in Java). This translation is essential because the high-level code itself cannot be executed by the CPU.

Compiled Languages

Compiled languages such as C, C++, Rust, Go, and Swift are programming languages that are translated into machine code by a compiler before they can be executed directly by the computer. The resulting machine code or executable file can run directly on a computer without needing further translation.

Advantages:

- Faster execution: The machine code runs directly on the CPU.
- Error detection: Many errors are caught at compile time.
- No interpreter needed: The program runs independently of the source code.

Disadvantages:

- Slow development: Requires a separate compilation step.
- Platform dependency: Compiled code is usually specific to the operating system (OS) and architecture.

Interpreted Languages

Interpreted languages such as Python, JavaScript, Ruby, and PHP are programming languages where the source code is executed line by line by an interpreter rather than being compiled into machine code first. The interpreter translates the code into machine instructions on the fly during execution.

Advantages:

- Faster development: Immediate feedback, no need for a compile step.
- Portability: The same code can run on different platforms as long as the interpreter is available.

Disadvantages:

- Slower execution: Interpreting code line by line is generally slower than running compiled code.

Hybrid Languages

Hybrid Languages such as Java and C# are languages that combine elements of both compiled and interpreted approaches. They may first be compiled into an intermediate form, which is then interpreted or executed by a virtual machine (VM).

Hybrid languages aim to combine the performance benefits of compilation with the flexibility of interpretation.