1.Introduction to computers and computer programs:

Computers are electronic devices that process and store data. Computer programs, also known as software, are sets of instructions that direct the computer to perform specific tasks.

2.Need for a programming language:

Programming languages are necessary to communicate with computers. They provide a structured way for humans to give instructions to computers, enabling them to execute tasks and solve problems.

3. Classification of programming languages

1. Machine Level Languages: Low-level languages directly correspond to machine code. Examples include assembly languages.
2. Low-Level Languages: These are closer to machine code and include assembly languages. They provide more control over hardware but are less user-friendly.
3. High-Level Languages: More abstract and user-friendly, allowing programmers to write code more efficiently. Examples include C++, Java, and Python.

4. Why use high-level programming languages:

High-level programming languages offer abstraction, making it easier for programmers to write and understand code. They provide built-in functions, improved readability, and portability across different platforms, reducing development time and errors.

5. Assembler, Interpreter, and Compiler:

1. Assembler: Translates assembly language code into machine code. It is specific to a particular architecture.
2. Interpreter: Executes code line by line. It translates and runs the source code directly. Examples include Python and JavaScript.
3. Compiler: Translates the entire source code into machine code before execution. The resulting executable file can be run independently of the compiler.

6. Phases of a C++ program:

1. Edit/Write: The programmer writes the source code using a text editor or an integrated development environment (IDE).
2. Preprocess: The preprocessor handles directives like includes and macros.
3. Compile: The compiler translates the source code into object code.
4. Link: The linker combines object code with libraries to create an executable file.
5. Load/Execute: The operating system loads the executable into memory, and the program is executed.