Introduction to Programming

Subject: C Language

Date: 20/12/2025

1. What is programming ?

⇒ Programming is the process of writing a program to give instructions to a computer so that it can perform a specific task or work.

2. What is Low-Level Language ?

⇒ Low level language is a programming language that is close to hardware and easy for the computer to understand but difficult for humans.

Two types of Low Level Language :-

1.) Machine Language
- Written in 0s and 1s
- Directly understood by CPU.
- Example :- 00011010 , 1110100l

2.) Assembly Language
- Uses mnemonics insted of 0s and 1s.
- Needs an assembler.
- Example :-

```
MOV A, B
ADD A, 05
```

3. What is High Level Language ?

⇒ High Level language is a programming language that is close to human

Subject: Date: / /

language and easy to write, read and understand.

Example :- C, C++, Java, Python, Javascript etc

4. What is Translator ?

⇒ A translator is system software that converts a program written in one language (high-level or assembly) into machine language (binary) so that the computer can understand and execute it.

5. Why Translator Is Needed ?

⇒ • Computer understands only machine language (0s and 1s).
• Programs are written in high-level or assembly language.
• Translator acts as a bridge between human and Computer.

Three types of Translators :-

1.) Compiler
• Converts the entire program at once.
• Produces an executable file.
• Errors are shown after Compilation.

Example :- • C Compiler
• C++ Compiler

Subject: ____________ Date: __/__/__

2.) Interpreter
- Converts and executes line by line
- No separate executable file.
- Stops at the first error.

Example :— • Python
• Javascript

3.) Assembler
- Converts assembly language into machine language.
- Uses mnemonics

Example :— • Assembly

6. Difference Between Compiler and Interpreter.

⇒

Feature	Compiler	Interpreter
Translation	Whole program	Line by Line
Speed	Faster execution	Slower execution
Executable file	Yes	No
Error reporting	After full scan	one by one

Subject: ______ Date: __/__/____

7. What is Features of C Language?

⇒ 1) Simple and Easy to Learn.
- Uses simple keywords and syntax.
- Easy to understand compared to low-level languages.

2) Structured Programming Language.
- Programs are divided into functions.
- Improves readability and debugging.
- Supports top-down approach.

3) Middle-Level Language.
- Supports low-level features (pointers, memory access).
- Supports high-level features (loops, functions)
- Used for both system and application programming.

4) Fast Execution
- Compiled language
- Direct interaction with hardware
- Very efficient and fast.

5) Portable (Machine Independent)
- Same C program can run of different machines
- Only Compiler changes; Code remains same.

6) Rich Library Support.
- Large number of built-in functions.
- Example : stdio.h, math.h, string.h.

Subject: Date: / /

7) Pointer Support.
- Allows direct memory manipulation.
- Used in arrays, structures, dynamic memory.

8) Dynamic Memory Allocation
- Memory can be allocated at runtime.
- Function: malloc(), calloc(), free()

9) Extensible
- New functions can be added by user.
- Easy to expand programs.

10) Low-Level Access
- Supports bitwise operations
- Used in OS, compilers, embedded systems.

8. How Processor and RAM work Together (Step-by-step).

⇒ 1.) Programming loading
- You open a program (C program, browser, etc)
- Program is stored on hard disk.
- OS loads the program into RAM.

NOTE:- CPU cannot directly work with hard disk — it needs RAM.

2) Fetch (Processor ⟷ RAM)
- CPU fetches instructions from RAM.
- Instruction address is stored in Program Counter (PC).

Example:- c = a + b ;

Subject: ______ Date: __/__/____

3.) Decode (Inside CPU)
- Control Unit (CU) decode instruction
- Decides:
  - what operation ?
  - Which data from RAM?
  - Which register/ALU to use ?

4.) Execute (Inside CPU)
- ALU Performs the operation
  - Read a and b from RAM
  - Add them

5.) Store result (CPU→RAM)
- Result is stored back in RAM
  C = 15;

6.) Repeat cycle
- This process repeats millions/billions of times per second.
  This is Called the Fetch - Decode - Execute cycle.