- **Write a detailed introduction to C programming language.**

Answer:

C is a powerful, general-purpose, structured, and high-level programming language developed by **Dennis Ritchie at Bell Laboratories in 1972**.

It was designed to build system software like operating systems and later became highly popular for application development.

C is often called the **"mother of programming languages"** because many modern languages such as C++, Java, PHP, Python, and JavaScript are either derived from it or influenced by its concepts.

The major reasons for its popularity are:

1. **Simple and easy to learn**
2. **Portable** — programs written in C can run on different machines with little modification
3. **Efficient and fast** — supports low-level memory access
4. **Structured programming support** — breaking problems into smaller functions
5. **Rich collection of built-in operators and library functions**

C language has a variety of applications including:

Operating system development
Database systems
Compilers and interpreters
Embedded systems
General application development

Because of its speed, flexibility, reliability, and control over hardware, C remains an essential programming language for students and professionals.

Q1) What is a Programming Language? Explain its need.

Answer:

A programming language is a formal language used to write instructions that a computer can understand and execute.

We use programming languages because:

1. **Humans cannot talk to computers directly.**
   Computers only understand binary (0s and 1s).
2. **Programming languages act as a bridge.**
   They convert human instructions into machine-understandable form.
3. They help in **solving problems**, **developing software**, **controlling devices**, and **processing data**.

Examples: C, Python, Java.

Q2) Explain Compiler, Interpreter, Linker, and Loader.

Answer:

1. Compiler

A compiler translates the entire program into machine language at once.
It reports errors after translation.

Example language: C, C++

2. Interpreter

An interpreter executes the program **line-by-line**.
It stops immediately when an error occurs.

Example language: Python, JavaScript

3. Linker

Linker combines multiple object files and libraries into a single executable program.
For example, when a C program uses printf(), linker connects stdio library to the program.

4. Loader

Loader loads the executable program into the computer's main memory (RAM) for execution.

*(All these concepts are mentioned in the Unit-1 syllabus list) *

Q3) What is an Algorithm? Give an example.

Answer:

An algorithm is a step-by-step procedure used to solve a problem.

Example: Algorithm to add 2 numbers:

1. Start
2. Input A
3. Input B
4. Compute Sum = A + B
5. Display Sum
6. Stop

Algorithms help convert real-life problems into programmable steps.

Q4) What is a Flowchart? Explain commonly used symbols.

Answer:

A flowchart is the graphical representation of an algorithm using standard symbols. It visually shows the flow of logic in a program.

Common Flowchart Symbols:

Symbol	Meaning
Oval	Start/Stop
Parallelogram	Input/Output
Rectangle	Process/Operation
Diamond	Decision

Flowcharts improve understanding and debugging of logic.

Q5) Explain the structure of a C program.

Answer:

A C program generally contains:

1. **Header files** – includes built-in functions
   Example: #include<stdio.h>
2. **Main function** – starting point of execution
   void main()
3. **Variable declarations** – storing data
   int a, b;
4. **Program logic/statements**
   Example: printf("Hello");
5. **Return statement** (optional)

This structure ensures organized execution and readability.

Q6) Write and explain a simple C program.

Answer:

```
#include <stdio.h>

void main()

{

   printf("Hello World");

}
```

Explanation:

#include<stdio.h> — allows use of printf()
main() — execution begins from here
printf() — prints message
; — indicates end of statement

This is the first C program to display output.

Q7) What are comments in C? Why are they used?

Answer:

Comments are text written inside programs which the compiler ignores.

Types:

Single line → // comment
Multi-line → /* comment */

Comments improve readability and documentation of code.

Q8) What are C tokens? List different types.

Answer:

Tokens are smallest individual units of a program.

Types:

1. Keywords — if, int, return
2. Identifiers — user defined names
3. Constants — fixed values
4. Operators — + - * /
5. Strings — "Hello"
6. Symbols — , ; () {}

Tokens are building blocks of C programs.

Q9) What are data types? Explain basic data types in C.

Answer:

Data types define the type of value a variable can store.

Basic data types:

Data type	Use
int	integer
float	decimal
double	large decimal

char	character

Example: int age = 18;

Q10) What is a variable? Write rules for naming variables.

Answer:

A variable is a name given to a memory location where data is stored.

Example: float marks = 90.5;

Rules:

Must start with letter/underscore
Case sensitive
No spaces
No reserved keyword use

Q11) Explain Operators and Expressions with examples.

Answer:

Operators

Symbols used to perform operations.

Types:

Arithmetic — + - * / %
Relational — > < ==
Logical — && || !

Expressions

Combination of operators and operands.
Example: x = a + b * 2;

Q12) What is Type Conversion? What is Typecasting?

Answer:

Type Conversion (Implicit Conversion)

Automatic conversion performed by compiler.

Example:

int a = 5;

float b = a; // 5 becomes 5.0

Typecasting (Explicit Conversion)

Manual conversion done by programmer using cast operator.

Example:

float x = 3.7;

int y = (int)x; // result = 3

Typecasting controls how data changes from one type to another.