Assignment-1

a. What is Programming Language? Describe the classification of Programming Languages.

A programming language is a set of instructions that allows humans to communicate with computers. Using programming languages, we write programs that perform specific tasks. C, Java, Python etc. are examples of programming languages.

Classification of Programming Languages

Based on the level of interaction with hardware, programming languages are classified into:

1. Machine Language

- The lowest-level language.
- Written in binary (0s and 1s).
- Directly understood by the computer.
- Very difficult for humans.

2. Assembly Language

- Uses symbolic codes called mnemonics (ADD, SUB).
- Easier than machine language but still hardware dependent.
- Requires an assembler for translation.

3. High-Level Language

- Easy to understand, close to human language.
- Examples: C, Java, Python.
- Requires compiler/interpreter.

b. Explain different input and output functions used in C.

1. printf() — Output Function

- Used to print text, variables, numbers etc.
- Syntax:
- printf("Message");
- Comes from <stdio.h>

2. scanf() — Input Function

- Used to take input from the user.

- Syntax:
- scanf("%d", &num);
- Reads data from keyboard.

3. getch() — Input (Character) Function

- Reads a single character without pressing Enter.
- Comes from <conio.h>

4. puts() and gets() (optional)

- Used for strings.
- gets() is unsafe so avoided in modern compilers.

c. C program to find maximum and minimum among 10 numbers

```
#include <stdio.h>

int main() {

    int arr[10], i, max, min;

    for(i = 0; i < 10; i++) {

        printf("Enter number %d: ", i+1);

        scanf("%d", &arr[i]);

    }

    max = min = arr[0];

    for(i = 1; i < 10; i++) {

        if(arr[i] > max)

            max = arr[i];

        if(arr[i] < min)

            min = arr[i];

    }

    printf("Maximum = %d\n", max);

    printf("Minimum = %d\n", min);
    return 0;

}
```

d. Explain the process of Compilation in detail.

Steps of Compilation

1. Source Code (.c file)
    - Code is written and saved with .c extension.

2. Preprocessing
    - Handles #include, #define, macros.
    - Inserts header files into program.

3. Compilation
    - Compiler checks errors.
    - Converts source code into object file (.obj).

4. Linking
    - Object file is linked with library functions (like printf, scanf).
    - Produces executable file (.exe).

5. Loading
    - Loader loads the program (.exe) into RAM.
    - Gives starting address to CPU.

6. Execution
    - CPU executes instructions one by one using Program Counter, Instruction Register etc.

e. Discuss various Special Control Statements used in C with example.

1. break
    - Terminates loop or switch immediately.
      Example:

```
if(i == 4)

    break;
```

2. continue

- Skips one iteration and moves to next.
  Example:

```
if(i == 4)

    continue;
```

3. goto

- Jumps to a labeled statement.
  (Not recommended but part of C.)

4. exit()

- Terminates entire program instantly.

f. What is a Pre-test Loop? Describe pre-test loops in C with example.

Pre-test Loop

A loop where the condition is checked before executing the loop body.
If condition is false at start → loop body will NOT execute even once.

Two Pre-test Loops in C

1. for Loop

```
for (initialization; condition; update) {

   // body

}
```

Example:

```
for(int i=0; i<5; i++)

    printf("Hello");
```

2. while Loop

```
while(condition) {

   // body

}
```

Example:

```
int i=0;

while(i<5){

   printf("Hi");

   i++;

}
```

g. What is an Operator? Explain Relational and Logical Operators with examples.

Operator

Relational Operators

Used to compare two values.

Operator	Meaning
>	Greater than
<	Less than
>=	Greater or equal
<=	Less or equal
==	Equal to
!=	Not equal

Example:

```
if(a > b)
```

Logical Operators

Operator	Meaning
&&	Logical AND
!	Logical NOT

Example:

```
if(a > 0 && b > 0)
```

h. Discuss Nesting of Structure with example.

Nesting of Structures

When one structure contains another structure inside it.

Example

```
struct Address {
   int house_no;
   char city[20];
};
struct Student {
   char name[20];
   int age;
   struct Address addr; // nested structure
};
```

i. What is Data Type? Explain different data types in C with example.

Data Type

A data type defines the type of data a variable can store (integer, float, char etc.)

Types of Data Types in C

1. Primitive Data Types

✔ int — stores integers
✔ float — stores decimal numbers (single precision)
✔ double — stores decimal numbers (double precision)
✔ char — stores a single character

Examples:

```
int age = 20;
float rate = 4.5;
char grade = 'A';
```

2. Non-Primitive Data Types

- Arrays
- Structures
- Strings, pointers (indirectly given in PDF)

Example:

```
int marks[5] = {10, 20, 30, 40, 50};
```