SE Modul - 1

What is Program

. **Program** – it is a set of instructions

**Que:** **Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.**

**Ans**. C language code for hello word

#include<stdio.h>

Void main()

{

Printf(“hello word”);

}

**Output: hello word**

**.** structure and syntax

**.** Requires a main() function and header file.

. Uses semicolons and braces.

. C is a compiled language, so it needs to

Be copiled before running.

. python code for hello word

Print (“hello, word!”)

Output: hello word;

. very simple and clean

. No need to define a main function

. NO semicolons and headers

. interpreted language: runs directly sours code

What is Programming

**. Programming** is the process of creating a set of instruction that tell a computer how to perform a specific task. These instructions are written using a **programming language**

**Que: Programming process**

1. Problem definition
2. Planning the solution
3. Coding
4. Testing and Debugging
5. Documentation
6. Maintenance

**Que: Type of Programming Languages**

1. Procedural programming Language

Ex – C Language

2. Object Oriented Programming

Ex – C++ Language

3. Logical Programming

Ex – Prolog Language

4.Function Programming

Ex – Python Language

**Que: Differences between High-level and low-level programming languages**

1. **High-level Languages:**

**. Abstraction:** They use syntax and structures closer to natural language,

Hiding the complexities of the underlying hardware.

**. Readability:** Easier for programmers to understand, write and maintain code.

**. Portability:** Code can be executed on different platforms with minimal or no Changes.

**. Examples:** Python, java, C++, JavaScript,

**. Uses:** Developing applications, websites and other software where ease of

Development and portability are prioritized.

**. Cons:** Can be slower and less memory-efficient than low-level languages.

1. **Low – Level Languages:**

**. Abstraction:** Close to machine code, requiring a deep understanding of

Computer architecture.

**. Readability:** Harder for humans to understand due to the use of mnemonics

and binary code.

**. Portability:** Generally not portable and tightly coupled with specific hardware or operating systems.

**. Examples:** Assembly languages, machine code.

**. Uses:** Developing system software, device drivers, and performance-critical

applications where speed and hardware control are crucial.

**. Cons:** Difficult to learn, program, and maintain.

**World Wide Web: WWW**

**.** WWWis a collection of webpages and resources that are accessed through the internet using web

Web browsers .

**.** The internet works as a global network of interconnected computers that communicate using standardized protocols like TCP/IP.