What is a Program?

LAB EXERCISE: Write a simple "Hello World" program in two different programming languages of

your choice. Compare the structure and syntax.

Ans. Here are simple **"Hello World"** programs in **Python** and **Java**, followed by a comparison of their structure and syntax:

**1. Python:**

print("Hello, World!")

**2. Java:**

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**Comparison of Structure and Syntax:**

| **Aspect** | **Python** | **Java** |
| --- | --- | --- |
| **Syntax Simplicity** | Very simple and concise. No need for extra declarations or boilerplate code. | Requires more boilerplate code (e.g., class definition, main method). |
| **Code Structure** | Single line of code is enough for the program. | Requires a class definition and a main method. |
| **Declaration of Entry Point** | No explicit entry point, code runs from top to bottom. | main method serves as the entry point. |
| **Brackets/Parentheses** | No curly brackets, just indentation for code blocks. | Curly brackets {} are required to define code blocks. |
| **Statement Terminator** | No semicolon required at the end of statements. | Statements must end with a semicolon (;). |
| **Type System** | Dynamically typed (no need to declare data types). | Statically typed (data types must be declared). |

**Summary:**

* **Python** is much more concise, with minimal syntax required, making it easier for beginners and faster to write simple programs.
* **Java** is more structured, requiring explicit declarations and a specific class-based structure. While this provides more control and clarity, it also makes the syntax heavier for simple tasks.

World Wide Web & How Internet Works

LAB EXERCISE: Research and create a diagram of how data is transmitted from a client to a server

over the internet.

Network Layers on Client and Server

LAB EXERCISE: Design a simple HTTP client-server communication in any language.

Types of Internet Connections

LAB EXERCISE: Research different types of internet connections (e.g., broadband, fiber,

satellite) and list their pros and cons.

Protocols

LAB EXERCISE: Simulate HTTP and FTP requests using command line tools (e.g., curl).

Application Security

LAB EXERCISE: Identify and explain three common application security vulnerabilities.

Suggest possible solutions.

Software Applications and Its Types

LAB EXERCISE: Identify and classify 5 applications you use daily as either system software

or application software.

Software Architecture

LAB EXERCISE: Design a basic three-tier software architecture diagram for a web application.

Layers in Software Architecture

LAB EXERCISE: Create a case study on the functionality of the presentation, business logic, and

data access layers of a given software system.

Software Environments

LAB EXERCISE: Explore different types of software environments (development, testing,

production). Set up a basic environment in a virtual machine.

Source Code

LAB EXERCISE: Write and upload your first source code file to Github.

Github and Introductions

LAB EXERCISE: Create a Github repository and document how to commit and push code changes.

Student Account in Github

LAB EXERCISE: Create a student account on Github and collaborate on a small project with

a classmate.

Types of Software

LAB EXERCISE: Create a list of software you use regularly and classify them into the

following categories: system, application, and utility software.

GIT and GITHUB Training

LAB EXERCISE: Follow a GIT tutorial to practice cloning, branching, and merging repositories.

Application Software

LAB EXERCISE: Write a report on the various types of application software and how they

improve productivity.

Software Development Process

LAB EXERCISE: Create a flowchart representing the Software Development Life Cycle (SDLC).

THEORY EXERCISE: What are the main stages of the software development process?

Software Requirement

LAB EXERCISE: Write a requirement specification for a simple library management system.

Software Analysis

LAB EXERCISE: Perform a functional analysis for an online shopping system.

System Design

LAB EXERCISE: Design a basic system architecture for a food delivery app.

Software Testing

LAB EXERCISE: Develop test cases for a simple calculator program.

Maintenance

LAB EXERCISE: Document a real-world case where a software application required

critical maintenance.

30. DFD (Data Flow Diagram)

LAB EXERCISE: Create a DFD for a hospital management system.

31. Desktop Application

LAB EXERCISE: Build a simple desktop calculator application using a GUI library.

32. Flow Chart

LAB EXERCISE: Draw a flowchart representing the logic of a basic online registration system.