**PHP Full Stack Assignment Set**

**Module 1 – Overview of IT Industry**

**Q-1 THEORY EXERCISE: Explain in your own words what a program is and how it functions.**

**Ans-**

A **program** is a set of instructions written in a specific language that a computer can understand and execute.

**How it functions:**

**1. Writing the Program:** A programmer writes the instructions using a programming language like Python, Java, or C++.

**2. Compiling or Interpreting:** Before the computer can follow the instructions, they often need to be translated into machine code (binary code the computer understands).

**3. Executing the Program:** Once translated, the computer's processor carries out the instructions in sequence—reading data, performing operations, and producing output.

**4. Output/Result:** The final result might be something the user sees on the screen, a file that's saved, oran action like sending a message.

**Q-2 THEORY EXERCISE: What are the key steps involved in the programming process?**

**Ans-**

1. **Problem Definition** – Understand what needs to be solved and gather requirements.

2. **Planning & Design** – Plan the solution, choose tools, and design algorithms or structure.

3. **Coding (Implementation)** – Write the actual program using a programming language.

4. **Testing & Debugging** – Check the program for errors and fix any bugs.

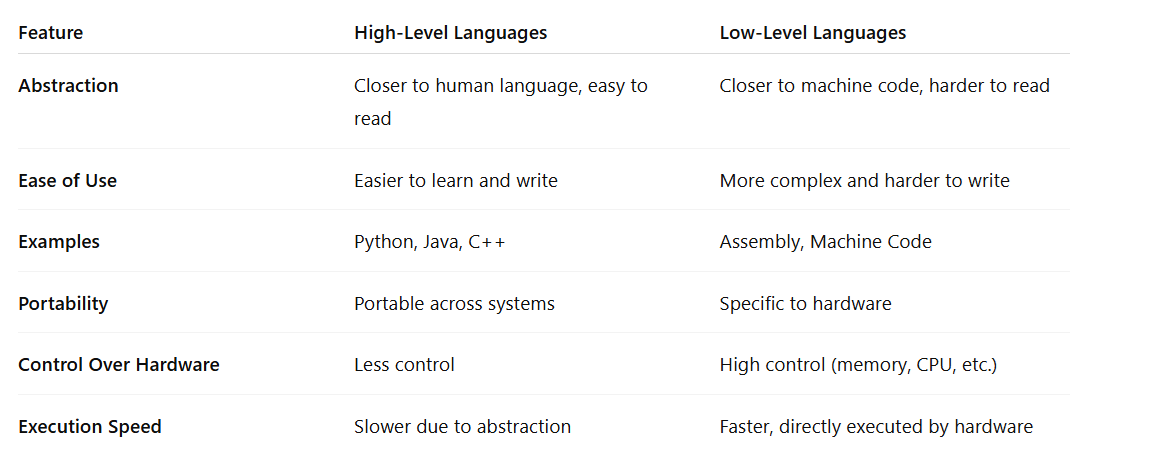
5. **Documentation** – Record how the program works and how to use it.

6. **Deployment** – Release the program for use.

7. **Maintenance** – Update the program to fix issues or add new features.

**Q-3 THEORY EXERCISE: What are the main differences between high-level and low-level programming languages?**

**Ans-**



**Q-4THEORY EXERCISE: Describe the roles of the client and server in web communication.**

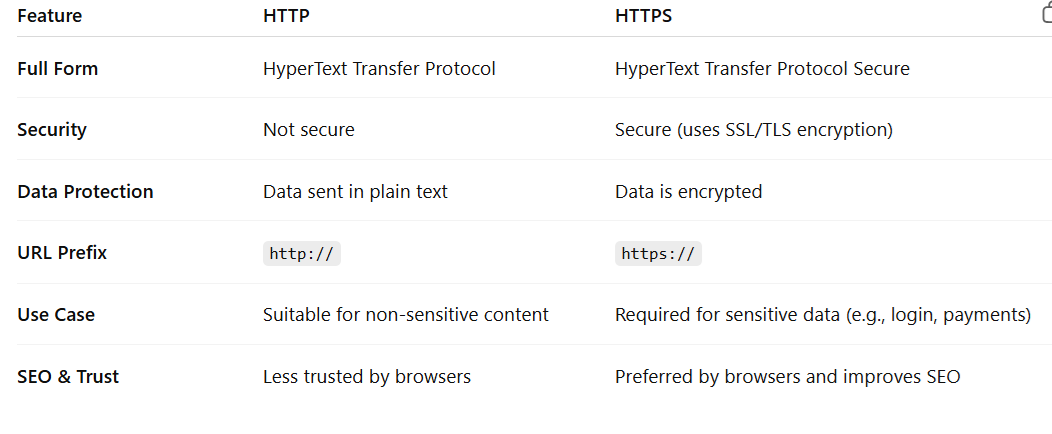
**Ans-**

 **Client**:  
The client is usually a web browser or app that **requests** data or services from a server. It sends a request (like opening a webpage) to the server.

 **Server**:  
The server is a powerful computer that **receives** requests, processes them, and **sends back** the appropriate response (like a webpage, image, or data).

**Q-5THEORY EXERCISE: What are the differences between HTTP and HTTPS protocols?**

**Ans-**

****

**Q-6 LAB EXERCISE: Identify and explain three common application security vulnerabilities.Suggestpossible solutions.**

**Ans-**

1. **SQL Injection** – Injecting malicious SQL via user input; **solution**: use prepared statements and input validation.
2. **Cross-Site Scripting (XSS)** – Injecting malicious scripts into web pages; **solution**: escape output and validate input.
3. **Broken Authentication** – Weak login mechanisms allow unauthorized access; **solution**: implement MFA and secure session handling.

**Q-7THEORY EXERCISE: What is the role of encryption in securing applications?**

**Ans-**

Encryption protects sensitive data by converting it into unreadable form, ensuring confidentiality during storage and transmission.

**Q-8LAB EXERCISE: Identify and classify 5 applications you use daily as either system software orapplication software.**

**Ans-**

 **Google Chrome** – Application software; used for browsing the internet.

 **Microsoft Word** – Application software; used for creating and editing documents.

 **Windows 10** – System software; manages hardware and provides a platform for applications.

 **Spotify** – Application software; used for streaming music.

 **File Explorer (Windows)** – System software; used for managing files and folders.

**Q-9THEORY EXERCISE: What is the difference between system software and application software?**

**Ans-**

**System software** runs and manages the computer hardware, while **application software** helps users perform specific tasks.

**Q-10THEORY EXERCISE: Whatis the significance of modularity in software architecture?**

**Ans-**

Modularity allows complex software to be divided into smaller, independent modules, making development, testing, and maintenance easier.

**Q-11THEORY EXERCISE: Why are layers important in software architecture?**

**Ans-**

Layers are important in software architecture because they organize code by roles, making it easier to build, understand, and maintain.

**Q-12THEORY EXERCISE: Explain the importance of a development environment in software production.**

**Ans-**

A development environment provides tools and setup that help programmers write, test, and debug code efficiently.

**Q-13THEORY EXERCISE: What is the difference between source code and machine code?**

**Ans-**

**Source code** is the set of instructions written by programmers in a high-level, human-readable language, while **machine code** is the low-level binary code that a computer's processor can directly execute.

**Q-14THEORY EXERCISE: Why is version control important in software development?**

**Ans-**

Version control is a system that tracks changes to code, allowing multiple developers to collaborate, manage revisions, and revert to previous versions if needed. It enhances teamwork, safeguards against data loss, and simplifies the process of integrating new features.

**Q-15THEORY EXERCISE: What are the benefits of using Github for students?**

**Ans-**

GitHub offers students a platform to practice version control, collaborate on coding projects, access open-source resources, and build a portfolio that can impress future employers.

**Q-16THEORY EXERCISE: What are the differences between open-source and proprietary software?**

**Ans-**

**Open-source software** has freely available source code for anyone to use or modify, while **proprietary software** has restricted access and is owned by a company.

**Q-17THEORY EXERCISE: Why is the requirement analysis phase critical in software development?**

**Ans-**

The requirement analysis phase is critical because it helps gather and clearly define what the users need, ensuring the software is built correctly and reducing costly errors or changes later in the development process.

**Q-18THEORY EXERCISE: What is the role ofsoftware analysis in the development process?**

**Ans-**

Software analysis identifies user needs, defines system requirements, and ensures a clear plan for building the right software.

**Q-19THEORY EXERCISE: Why is software testing important?**

**Ans-**

Software testing is important because it ensures the software functions correctly, meets user requirements, and helps identify and fix bugs before release, improving quality and reliability.

**Q-20THEORY EXERCISE: What types ofsoftware maintenance are there?**

**Ans-**

There are **four main types of software maintenance**:

1. **Corrective Maintenance** – Fixing bugs or errors found after the software is released.
2. **Adaptive Maintenance** – Updating the software to work in a new environment (e.g., new OS or hardware).
3. **Perfective Maintenance** – Improving performance or adding new features based on user feedback.
4. **Preventive Maintenance** – Making changes to prevent future problems and improve long-term stability.

**Q-21THEORY EXERCISE: What are the advantages of using web applications over desktop applications?**

**Ans-**

Advantages of web applications over desktop applications:

* Accessible from any device with internet
* No installation required
* Automatic updates for all users
* Works on different operating systems
* Easier for developers to maintain and update

**Q-22THEORY EXERCISE: What role does UI/UX design play in application development?**

**Ans-**

UI/UX design plays a crucial role in application development by focusing on how the app looks (User Interface) and how users interact with it (User Experience).

**Q-23THEORY EXERCISE: What are the differences between native and hybrid mobile apps?**

**Ans-**

**Native apps** are developed for a specific platform (like iOS or Android) using platform-specific languages (like Swift or Kotlin), offering better performance and access to device features. **Hybrid apps** are built using web technologies (like HTML, CSS, JavaScript) and run inside a native shell, making them cross-platform but sometimes with reduced performance and limited native functionality.

**Q-24THEORY EXERCISE: What is the significance of DFDs in system analysis?**

**Ans-**

Data Flow Diagrams (DFDs) are significant in system analysis as they visually represent how data moves through a system, helping analysts understand processes, data inputs/outputs, and data storage clearly and effectively.

**Q-25THEORY EXERCISE: What are the pros and cons of desktop applications compared to webapplications?**

**Ans-**

**Pros of Desktop Applications:**

* Faster performance and better use of system resources
* Can work offline without internet access
* More control over hardware and system integration

**Cons of Desktop Applications:**

* Limited accessibility (only on the installed device)
* More complex updates and maintenance
* Platform-dependent (Windows, macOS, etc.)

**Pros of Web Applications:**

* Accessible from any device with internet
* Easier updates and maintenance
* Platform-independent

**Cons of Web Applications:**

* Requires internet connectivity
* May have slower performance
* Limited access to device hardware

**Q-26THEORY EXERCISE: How do flowcharts help in programming and system design?**

**Ans-**

Flowcharts help in programming and system design by visually representing the logical flow of processes, making it easier to understand, plan, debug, and communicate the structure and sequence of operations.