**Module 1**

* **What is programe?**

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

A set of instructions that a computer follows in order to perform a particular . A computer program consists of **code that is executed on a computer to perform particular tasks**. This code is written by programmers. Programming is the process of giving machines a set of instructions that describe how a program should be carried out.

* **WHAT IS PROGRAMING?**

**Q. EXERCISE: What are the key steps involved in the programming process?**

The key steps in the programming process include analyzing the problem, designing the solution, writing the code, testing and debugging, and documenting the program. These steps ensure a well-structured and functional program is created.

* **Types of Programming Languages**

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

High-level languages are human-friendly and abstract, making them easier to learn and use, while low-level languages are closer to the machine and require detailed knowledge of the computer's architecture.

* **World Wide Web & How Internet Works**

**Q.: Research and create a diagram of how data is transmitted from a client to a server over the internet.**

a client is a device or software that requests resources or services from a server, which provides those resources or services.

**SERVER--------------INTERNENT-----------CLIENTS [MOBILE, COMPUTER, TV,]**

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

Client-server networks are computer networks that employ a dedicated computer to store data, manage/provide resources, and control user access (server). A web server takes a client request and gives a response back to the client. A web client enables user to request a resource kept over any web server

* **Network Layers on Client and Server**

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

request

←−−−−−−−− [TV]

SERVER--------------INTERNET--------[LEPTOP]

response

−−−−−−−−→ [ PC]

**THEORY EXERCISE: Explain the function of the TCP/IP model and its layers.**

The TCP/IP model, also known as the Internet protocol suite, is a conceptual framework that standardizes how data is transmitted over networks. It divides the communication process into four layers: Application, Transport, Internet, and Network Access.

* **Client and Servers.**

THEORY EXERCISE: Explain Client Server Communication Client-server communication is a model where a client (e.g., browser, mobile app) requests services from a server (e.g., web server, database), which processes the request and responds over a network using protocols like HTTP, WebSockets, or gRPC.

* **Types of Internet Connections**

**LAB EXERCISE: Research different types of internet connections (e.g., broadband, fiber, satellite) and list their pros and cons.**

Different types of internet connections include broadband, fiber, DSL, cable, satellite, and wireless options like 4G/5G. Broadband is a general term for high-speed internet, while fiber uses light signals for extremely fast and reliable connections. DSL relies on phone lines, cable uses coaxial cables, and satellite beams data from space. Wireless options, such as 4G/5G, utilize cellular networks for internet access.

**THEORY EXERCISE: How does broadband differ from fiber-optic internet?**

Broadband is a general term for any high-speed internet connection, while fiber-optic internet is a specific type of broadband that uses fiber optic cables to transmit data.

* **.Protocols**

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

* FTP-- FTP stands for File transfer protocol.
* FTP is a standard internet protocol provided by TCP/IP used for transmitting files from one host to another.
* It is mainly used for transferring web page files from their creator to the COMPUTER that acts as a server for other computer on the internet.
* It is also used for downloading files to computers from other servers.
* **.HTTP:**
  + HTTP stands for **Hypertext Transfer Protocol**.
  + It is a protocol used to access the data on the World Wide Web (www).
  + The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.

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

The main difference between HTTP and HTTPS is that HTTPS provides encryption to secure data transmission, while HTTP does not

In essence, HTTPS provides a secure layer on top of HTTP, making it the preferred protocol for websites dealing with sensitive information.

* **Student Account in Github**

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

Using GitHub for your school projects is a practical way to collaborate with others and build a portfolio that showcases real-world experience. Everyone with a GitHub account can collaborate in unlimited public and private repositories with GitHub Free.

* **.Application Software**

**THEORY EXERCISE: What is the role of application software in businesses?**

Application software plays a crucial role in modern businesses by enabling efficient task management, enhancing productivity, and streamlining operations.

* **Software Applications and Its Types**

**LAB EXERCISE: Identify and classify 5 applications you use daily as either system software or application software.**

System software, like an operating system, manages the computer's hardware and provides a platform for other software to run. Application software, on the other hand, is designed for specific tasks like word processing, browsing, or gaming, and relies on system software to function

* **Software Architecture**

**THEORY EXERCISE: What is the significance of modularity in software architecture?**

Modularity in software architecture is significant because it promotes flexibility, reusability, and maintainability. By breaking down a system into independent modules, developers can create adaptable solutions, simplify development, and improve overall system quality. This approach also enables easier updates, testing, and collaboration, ultimately leading to scalable software.

* **Layers in Software Architecture**

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

Layers are crucial in software architecture because they enable modularity, improve maintainability, and facilitate testing and scalability.

* **Source Code**

**Source Code: THEORY EXERCISE: What is the difference between source code and machine code?**

* + **Human-readable:** Written in languages like Python, Java, C++, etc., using words and symbols familiar to programmers.
  + **Designed for programmers:** Makes it easier for humans to understand, write, and modify code.
  + **Not directly executable:** Requires compilation or interpretation to be converted into machine code.
  + **Example:** int main() { printf("Hello, world!"); }.

Machine Code:

* + **Computer-executable:** A sequence of binary digits (0s and 1s) that a computer's processor can directly understand and execute.
  + **Machine-specific:** Different processors have different machine code languages.
  + **Difficult to read and modify:** Not designed for human comprehension or editing.
  + **Example:** A sequence of binary instructions like 10110000 01100001 (this is a simplified example and machine code is typically much more complex).
* **Github and Introductions**

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

Version control is crucial in software development because it helps manage changes to code, making it easier for teams to collaborate, track progress, and recover from mistakes.

* **Student Account in Github**

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

Using GitHub for your school projects is a practical way to collaborate with others and build a portfolio that showcases real-world experience. Everyone with a GitHub account can collaborate in unlimited public and private repositories with GitHub Free.

* **Types of Software**

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

Open-Source Software:

* **Source Code Availability:** The source code is freely available for anyone to view, use, and modify.
* **Licensing:** Uses open-source licenses (e.g., MIT, Apache, GPL) that grant users specific rights.
* **Development Model:** Typically developed collaboratively, with contributions from a community of developers.
* **Security:** Benefits from community-driven security improvements and faster vulnerability patching.
* **Examples:** Linux, Firefox, VLC media player.

Proprietary Software:

* **Source Code Availability:** Source code is not publicly available and is kept confidential by the software owner.
* **Licensing:** Users typically require a license to use the software, often with restrictions on usage.
* **Development Model:** Developed and maintained by a private team of developers.
* **Security:** Relies on the vendor for security updates and patches.
* **Examples:** Windows, Microsoft Office, Adobe Photoshop.
* **System Design**

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

Key elements of system design include architecture, components, interfaces, data, and the interaction between these elements. These elements work together to define how a system will function, how its parts are organized, and how it will meet specified requirements.

* **.Maintenance**

**THEORY EXERCISE: What types of software maintenance are there?**

A real-world case where a software application was essential is a hospital's Electronic Health Records (EHR) system. This system manages patient information, medical history, appointments, and billing, requiring a robust and secure application to ensure efficient and accurate healthcare delivery.

* **Software Testing**

**THEORY EXERCISE: Why is software testing important?**

1. Step 1: Understand the requirements. ...
2. Step 2: Define the test case objective. ...
3. Step 3: Identify test scenarios. ...
4. Step 4: Write a descriptive test case title. ...
5. Step 5: Set preconditions (if any) ...
6. Step 6: List the test steps. ...
7. Step 7: Define the test data.

* **Maintenance**
* **THEORY EXERCISE: What types of software maintenance are there?**

Software maintenance can be broadly categorized into four types: corrective, adaptive, perfective, and preventive.

* **.Development**

**THEORY EXERCISE: What are the key differences between web and desktop applications?**

Web and desktop applications differ primarily in how they are accessed and deployed, their **reliance on internet connectivity, and their performance characteristics.**

* **Web Application**

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

Web applications offer several advantages over desktop applications, primarily due to their accessibility, ease of maintenance, and cost-effectiveness.

* **Designing**

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

UI/UX design plays a crucial role in application development by ensuring a user-friendly and engaging experience. It focuses on both the visual interface (UI) and the overall user journey (UX), ultimately impacting user satisfaction, app performance, and business success.

* **Mobile Application**

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

Native mobile apps are built specifically for one operating system (like iOS or Android) using its native programming language (like Swift for iOS or Java/Kotlin for Android), offering optimal performance and access to device features. Hybrid apps, on the other hand, are built with web technologies (HTML, CSS, JavaScript) and wrapped in a native container, allowing them to run on multiple platforms.

* **. Desktop Application**

**THEORY EXERCISE: What are the pros and cons of desktop applications compared to web applications?**

Desktop applications, while offering advantages like performance and offline functionality, also have drawbacks such as platform dependency and update management.

* **Flow Chart**

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

Flowcharts are valuable tools in programming and system design because they offer a visual representation of a process, making it easier to understand, analyze, and debug code or system workflows.