Module 1 – Overview of IT Industry

**- *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.*

🡪 In Python Programming language we can write “Hello World” as:

SYNTAX-

print (“Hello World”)

In C Programming language we can write “Hello World” as:

SYNTAX-

#include <sdtio.h>

Int main () {

printf (“Hello World”);

return 0;

}

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

🡪 Program is refer to set of instructions or statements that tells us how to complete any task in step-by-step order. The function of program is to give instructions to perform a specific task.

-***What is Programming?***

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

🡪 The key steps involved in the programming process are:

1. Understanding the problem
2. Planning solution
3. Writing the program
4. Testing and Debugging
5. Execution of program
6. Maintenance of program

***-Types of Programming Languages***

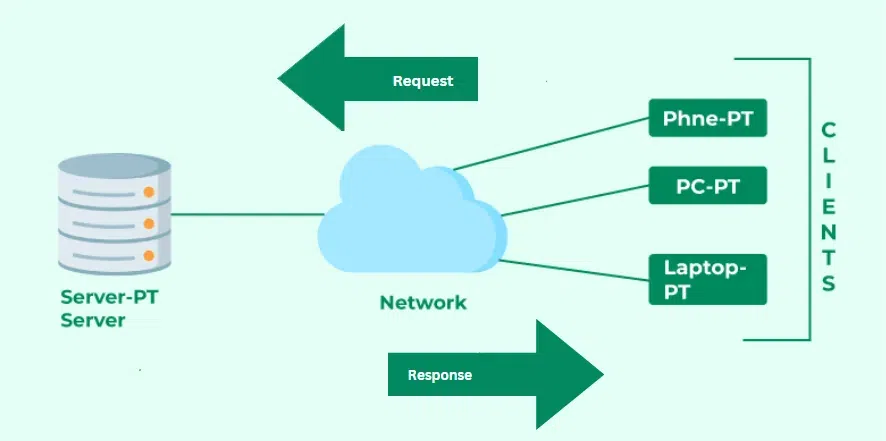
***THEORY EXERCISE****: What are the main differences between high-level and low-level programming languages?*

🡪 The main key difference between high-level and low-level programming language is that high-level languages are understandable by humans and is very convenient, while low-level languages are known as machine language so they are not friendly with humans so they are hard to understand.

***-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.*

🡪 This is the diagram for Client-Server data transmission which shows the process of data transfer when requested by the client and response of that data from server.



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

🡪 Roles of Client and Server in web communication are:

Client- The client is web browser or application that sends a request to the server asking for any data or services (e.g. Webpage).

Server- The server is a system that stores the data, processes them according to client requests and delivers the requested data or services to the client. It responses by sending back the data or services to the requests made by the client (e.g. HTML file, image, etc.).

***-Network Layers on Client and Server***

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

🡪 Same as “*of how data is transmitted from a client to a server over the internet*”.

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

🡪 The TCP/IP model (Transmission Control Protocol/Internet Protocol) is used for network communication. It defines how data is sent, received, and routed over the Internet.

TCP/IP model has 4 layers and their function are:

1. Application Layer: Let apps use the network.
2. Transport Layer: Make sure data is sent correctly.
3. Internet Layer: Decide the route for the data.
4. Network Access Layer: Send data over actual cables or Wi-Fi.

***-Client and Servers***

***THEORY EXERCISE****: Explain Client Server Communication*

🡪 Client-Server communication is the process of data exchange between a client and a server over some network, mainly over the internet. The client is the device that requests a service or data, and the server is the system that stores the data provides that request from the client and send back a response.

***-Types of Internet Connections***

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

🡪 The different types of internet connections are DSL, cable, fiber-optic, wireless, broadband and satellite.

Here are the pros and cons of each internet connections:

1. DSL:-

Pros- Uses existing phone infrastructure so can be widely available; relatively cheap; decent for basic tasks (browsing, email).

Cons- Speed drops if you’re far from the phone exchange; upload speeds often much slower; not good for very high-bandwidth needs

1. Cable:-

Pros- Much faster than DSL generally; widely available in urban/suburban areas; good for streaming, gaming

Cons- Shared bandwidth means slowdowns during peak usage; latency isn’t as low as fiber; performance can vary.

1. Fiber-optic:-

Pros- Very high speeds, often symmetrical upload & download; low latency; reliable; good for future growth and heavy data use.

Cons- Expensive to deploy especially in remote/rural areas; not everywhere is connected yet; installation cost can be higher.

1. Wireless:-

Pros- Faster installation (no need to dig long cables); can serve places where wired infrastructure is poor; good speeds in ideal conditions.

Cons- Requires line‐of‐sight to the wireless tower; weather or obstacles can affect signal; speed can fluctuate.

1. Broadband:-

Pros- Very flexible, you can use it on-the-go; newer tech (5G) offers high speeds, good for many modern uses.

Cons- Data caps/plans may be restrictive; network congestion (many users sharing towers) can slow you; coverage isn’t uniform (especially 5G) so performance varies.

1. Satellite:-

Pros- Can reach very remote or rural areas; doesn’t need many ground cables; good fall back or only option in some places.

Cons- High latency (delay) because signals travel long distances; more expensive; weather can interfere; often data caps; slower upload speeds.

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

*🡪* Broadband differs from Fiber-optic as each carries different transmission medium, signal type, speed, reliability, cost and availability.

In Broadband we have:

1. Copper wires / Coaxial cables for transmission.
2. Signal type is electrical.
3. Speed is moderate.
4. Reliability affected by distance.
5. Cost is low.
6. Availability is wide.

In Fiber-optic we have:

1. Glass or plastic fiber cables for transmission.
2. Signal type is Light.
3. Speed is very high.
4. Reliability is very stable & consistent.
5. Cost is high.
6. Availability is limited.

***-Protocols***

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

🡪

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

🡪 The difference between HTTP and HTTPS protocols are:

1. The HTTPS protocol is more secure then HTTP as it provides a layer of authentication and it is then only be accessed by the verified user.
2. The performance is better in HTTP is higher as it does have layer like the HTTPS protocol.

***-Application Security***

***LAB EXERCISE****: Identify and explain three common application security vulnerabilities. Suggest possible solutions.*

🡪

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

🡪 The role of encryption in securing applications is by protecting **sensitive information** from unauthorized access that only authorized user can only read it.

***-Software Applications and Its Types***

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

🡪 Google Chrome, YouTube, Spotify, Microsoft Word, Windows Defender.

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

🡪 The difference between system software and application software is that System software runs the computer and manages hardware, while application software helps users perform specific tasks.

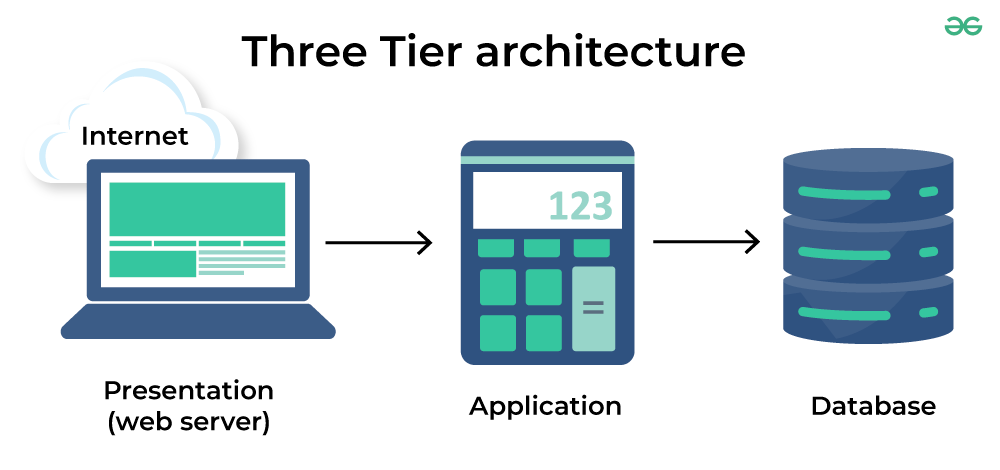
***-Software Architecture***

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

🡪 The three-tier software architecture for a web application involves:

1. Presentation Tier
2. Application Tier
3. Database Tier

Below is the diagram for the three-tier software architecture.



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

🡪 The significance of modularity is that it makes software easier to understand, develop, test, and maintain. Since each module works independently, developers can make changes or fix issues in one part without affecting the whole system. Modularity helps build cleaner, more organized, and flexible software.

***-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.*

🡪 Case Study: Fitness Tracker Application

1. Presentation Layer-

This layer is responsible for the user experience and interaction. It is what the user sees and interacts with on their screen.

* Displays dashboards, charts, and fitness progress.
* Allows users to log workouts, set goals, or track steps.
* Shows feedback messages like “Workout Added” or “Goal Achieved.”
* Sends requests to the client layer.

1. Application Layer-

The client layer acts as a bridge between the user interface and the business logic.

* Manages user requests and sends them to the business logic layer.
* Handles responses from the backend and delivers results back to the UI.
* Maintains session information such as login tokens and temporary data.

1. Business Layer-

This layer contains the core logic and processing rules of the app.

* Validates user data and ensures proper input.
* Calculates workout statistics like calories burned or steps achieved.
* Applies fitness rules (e.g., “Drink 2 litres of water daily”).
* Interacts with the data access layer for storing or retrieving data.

1. Persistence Layer-

The data access layer acts as a middleman between the business logic and the database. It manages all operations related to data storage and retrieval.

* Executes queries such as insert, update, delete, and fetch.
* Ensures data consistency and error handling.
* Converts raw database data into usable formats for the business layer.

1. Database Layer-

This is the lowest layer that stores all application data permanently and securely.

* Stores user profiles, workout logs, step counts, and nutrition data.
* Supports backup and recovery for data safety.
* Handles indexing and query optimization for faster data access.

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

🡪 Layers are important in software architecture because they organize the system into separate parts, each with a specific role. This makes the software easier to develop, understand, and maintain.

***-Software Environments***

***LAB EXERCISE****: Explore different types of software.*

🡪