Modual 1 :- Overview of IT Industry

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

➡ (1) C language.

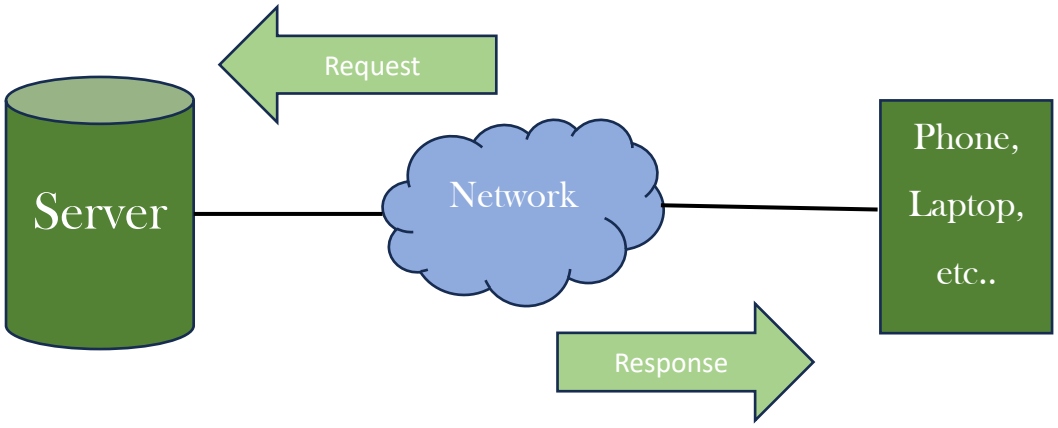
```
#include<stdio.h>
 int main()
{
   printf("Hello World");
   return 0;
}
```

(2) Python language.

```
print("Hello World")
```

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

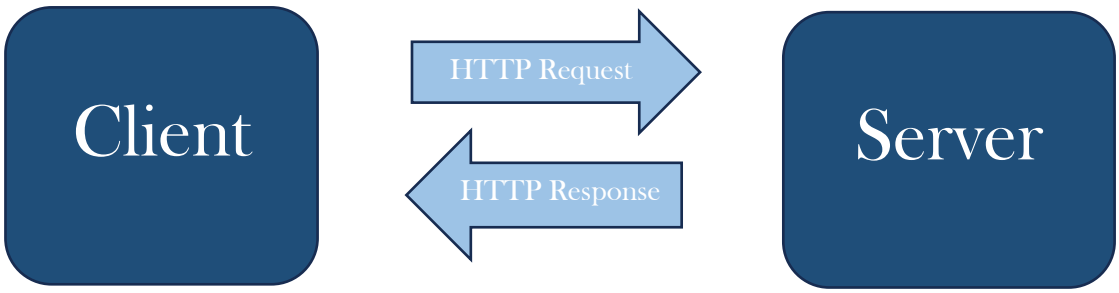
➡



1. Client send request
2. Request travels through the network
3. Server processes the request
4. Data send back to the client
5. Client receives respons.

3. Design a simple HTTP client-server communication in any language.

➡



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

➡ Here's a list of Different type of Internet connexion with a pros and cons:

1. Dial-up:- 1

Pros:

It is cheaper

It work anywhere

Cons:

It is very slower

2. Cable:-

Pros:-

It is faster

It work slower at peak time

3. Fiber Optic:-

Pros:

Super fast

More reliable

Handle heavy use/task

Cons:

It is more expensive

Not available everywhere

4. Satellite:-

Pros:

Work in remote areas

Easier to set-up

Cons:

Slower

It can be affected by bad weather

5. Fixed wireless:-

Pros:

Easier to set-up

Good for rurak areas

Cons:

Lower speed

6. Broadband:-

Pros:

Portable

Work almost anywhere

Cons:

Limited data

Weak in some areas

7. Wi-Fi Hotspot:-

Pros:

Easy to use

Good for travel

Cons :

Need a internet connection

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

6.Identify and explain three common application security vulnerabilities. Suggest possible solutions.

➡

1. SQL Injection:

Attackers insert malicious SQL commands into input fields to manipulate or access a database.

Solution:

Use prepared statements or parameterized queries.

Validate and sanitize user input.

2. Cross-Site Scripting :

Attackers inject malicious scripts like JavaScript into a web page that is then executed in users' browsers.

Solution:

Escape or sanitize user input before displaying it.

Avoid directly including untrusted input in HTML or scripts.

7.Identify and classify 5 applications you use daily as either system software or application software.

➡

1. Operating System:

Type: System Software

It manages hardware and software resources and provides a platform for other applications to run.

2. Web Browser:

Type: Application Software

It helps users browse the internet access websites , and use web based applications.

3. Word Processor

Type: Application Software

It allows users to create, edit, and customize text documents.

4. Antivirus Program:

Type: System Software

It protects the system from malware and secure safe operation.

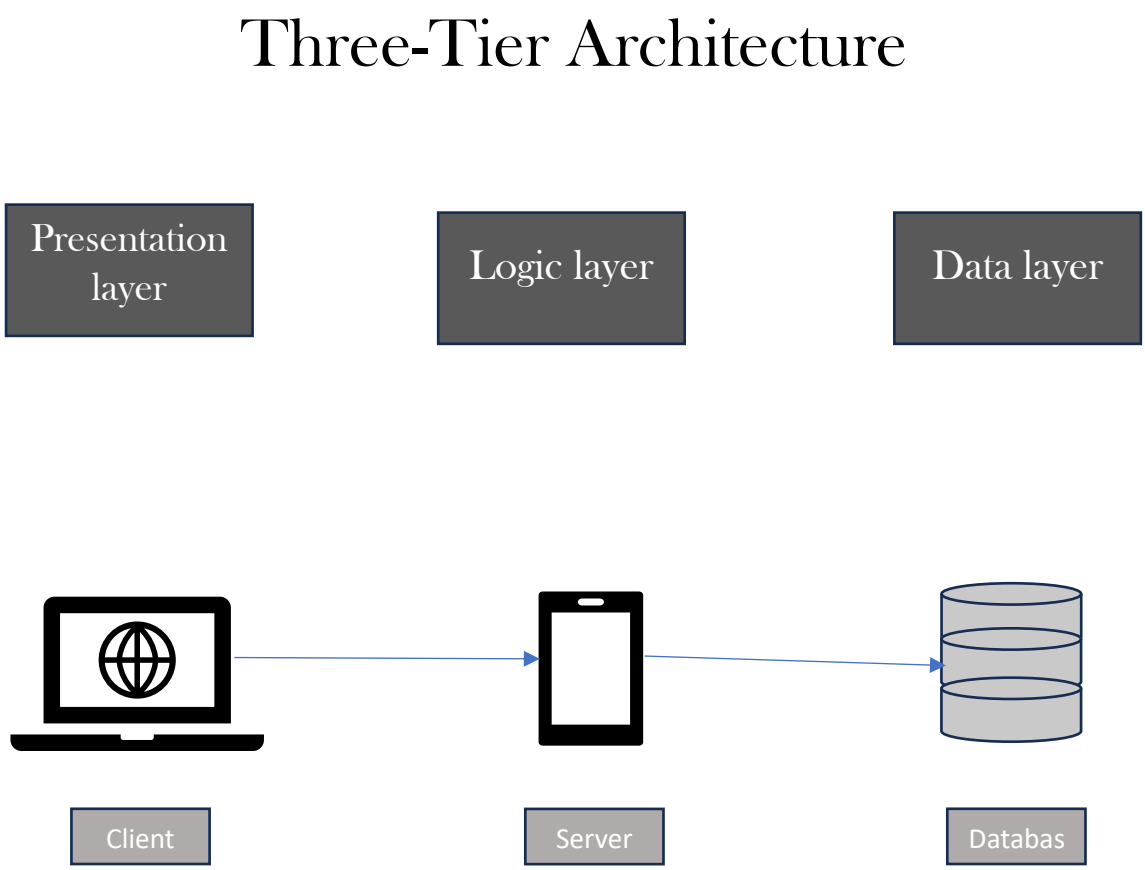
5. Messaging App:

Type: Application Software

It enables users to communicate via text, voice, or video.

8. Design a basic three-tier software architecture diagram for a web application.

➡



9. Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.

➡

1. Presentation Layer

- This is what users see and interact with, like the website or mobile app.

  The presentation layer sends the search request to the next layer (business logic) and displays the results it gets back.

2. Business Logic Layer

- This is the brain of the system. It processes user requests and applies rules to make decisions.
- It talks to the data access layer to fetch product details, applies any rules, and sends the information back to the presentation layer.

3. Data Access Layer

- This is where all the data is stored and retrieved from. It connects to the database.
- The business logic layer asks this layer for specific data (e.g., "show all headphones in stock"), and this layer retrieves it securely.

10. Explore different types of software environments (development, testing, production). Set up a basic environment in a virtual machine.

➡

1. Development Environment

- Purpose:
  - o Developers write, debug, and test code.
  - o Frequently code changes here.

- Tools :

    o IDEs (integrated Development Environment)like Visual studio, intellij.

    o Local servers for quick testing.

2. Testing Environment

- Purpose:

    o QA (Quality assurance) teams test applications for bugs, performance, and security issues.

- Tools:

    o Automated testing tools o Load testing tools.

3. Production Environment

- Purpose:

    o The live environment where the application is available to end users. o Highly stable, with minimal downtime.

- Tools:

    o Monitoring tools o Security measures like firewalls and backups.

11. Write and upload your first source code file to Github.

➡

```
print("Hello, World!")
```

Upload to github

```
git init
git add hello_world.py
git commit -m "First commit"
git branch -M main
```

git remote add origin https://github.com/username/repo.git

git push -u origin main.

12. Create a Github repository and document how to commit and push code changes.

➡ Git push: Pushes changes from the local repository to GitHub.

Git commit -m “message”:

13.Create a student account on Github and collaborate on a small project with a classmate.

➡

1.Go to the repository you just created.

2.Click on the Settings tab in the repository.

3.Scroll down to the Collaborators section on the left sidebar.

4.Under Manage access, click Invite a collaborator.

5.Search for your classmate’s GitHub username and invite them to your repository.

14. Create a list of software you use regularly and classify them into the following categories: system, application, and utility software.

➡

1. System Software:

These manage and control computer hardware

- Operating system- Windows, macOS, Linux
- Device drivers- Printer drivers, graphics drivers

2. Application Software : These help perform specific tasks

- Web browsers- google chrome, mozilla firefox
- Productivity tools- microsoft word, excel, powerpoint
- Messaging apps- whatsApp, microsoft teams, slack
- Media players- VLC media player, spotify
- Graphics tools- photoshop, canva

3. Utility Software:

These maintain and optimize computer performance.

- Antivirus software- McAfee
- File tompression tools- Tar,7-Zip
- Backup tools- google drive backup
- Disk cleanup tools- cleaner
- Monitoring tools- task manager.

15. Follow a GIT tutorial to practice cloning, branching, and merging repositories.

➡

16. Write a report on the various types of application software and how they improve productivity.

➡

1. Word Processing Software

Ex: Microsoft Word, Google Docs

Purpose: Helps users create, edit, and format documents like letters, essays, and reports.

How It Improves Productivity:

- Includes spelling and grammar tools to reduce errors.
- Enables collaboration through sharing and real-time editing.

2. Spreadsheet Software:-

Ex: Microsoft Excel, Google Sheets

Purpose: Used for organizing data, calculations, and creating charts.

How It Improves Productivity:

- Automates calculations with formulas and functions.

3. Presentation Software

Ex: Microsoft PowerPoint, Canva

Purpose: Helps design slides for presenting information.

How It Improves Productivity:

- Provides templates for quick slide creation
- Allows multimedia integration, such as videos and images

4. Email and Communication Software

Examples: Outlook, Gmail, Zoom, Slack

Purpose: Facilitates communication through emails, video calls, and instant messaging

How It Improves Productivity:

- Speeds up communication with quick messaging and email
- Organizes the messages with filter like search and sorting

5. Database Software

Ex: Microsoft Access, MySQL

Purpose: Manages and organizes large amounts of data efficiently.

How It Improves Productivity:

- Stores data systematically for easier access
- Reduces manual work by automation of data

6. Web Browsers

Ex: Google Chrome, Mozilla Firefox

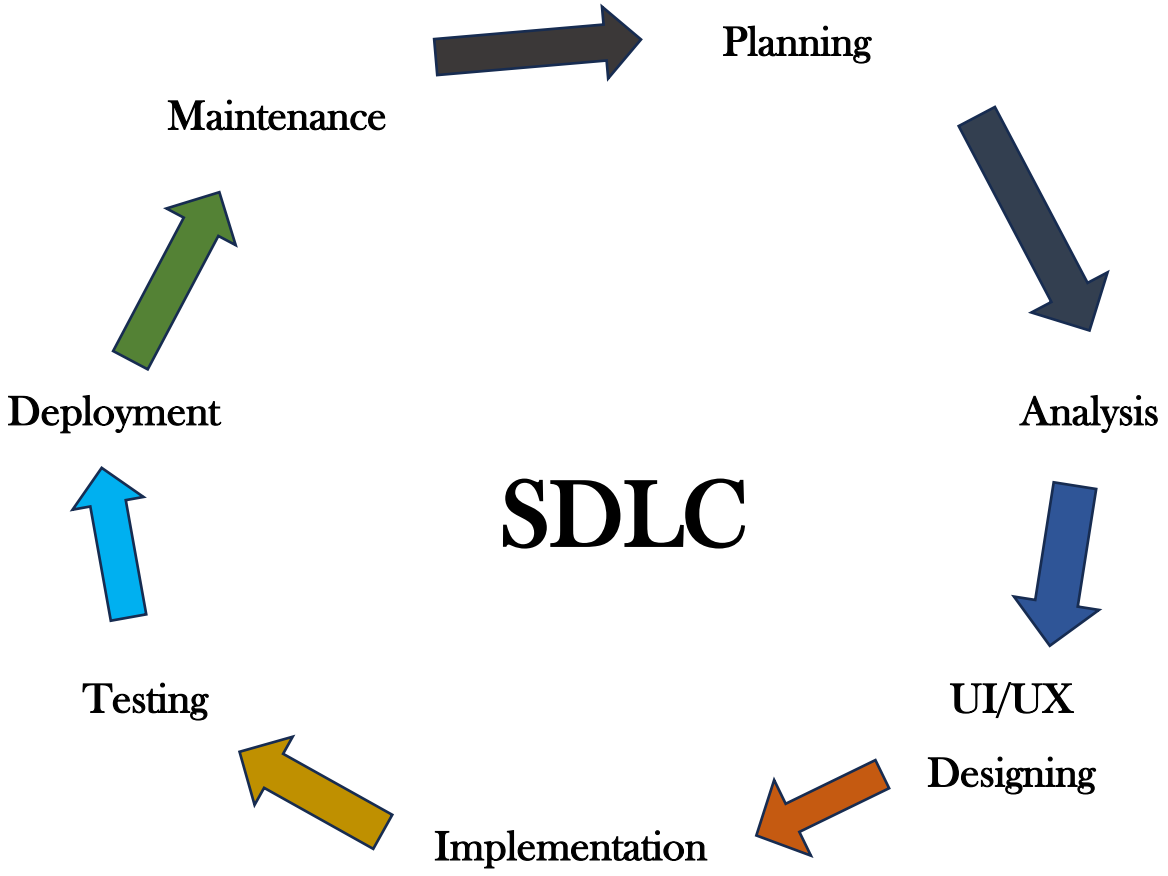
Purpose: Provides access to the internet and online tools.

How It Improves Productivity:

- Enables quick research and access to online resource.
- Supports cloud-based applications for remote work.

17. Create a flowchart representing the Software Development Life Cycle (SDLC).

➡



18. Write a requirement specification for a simple library management system.

➡

1. Purpose:

The Library Management System is designed to help libraries manage books, users, and transactions like issuing and returning books.

2. Features :

1. Book Management

o Add new books to the library

o Update book details by title, author

o Remove unavailable book

2. User Management

o Register new users

o Update user details like name, contact, address ,E-mail

3. Book Transactions

o Issue books to users

o Track due dates and late fees

4. Search and Reports

o Search books by title, author

o Check user borrowing data

3. System Requirements :

1. Hardware Requirements

o A computer with basic specifications.

2. Software Requirements

o Operating System windows, macOS, or linux.

o Programming Language: Python or Java

o Database: MySQL , excel

4. User Roles:

1. Librarian

o Can add, update, and delete books and user records

o Can return books or view report

2. Member/User

o It can search for books

o Can view borrowing history.

19. Perform a functional analysis for an online shopping system.

➡

1. User Management :

• User registration and login

• Profile management

• Password recovery and change

2. Product Management:

• Add, update, and remove products it can only done by admin

• Organize products into categories and subcategories

• Display product details like name, price, description, images, rating

3. Product Search and Navigation :

• Search bar for keyword-based product

• Filters for price, category, brand, ratings , price (low to high, high to low),

• Show related products

4. Shopping Cart :

• Add products to the cart

• Add quantity or remove items from the cart.

• View the total price and item list

5. Order Management :

• Place orders from the shopping cart

• Track order status like pending, shipped, delivered, or canceled

• Cancel or return orders based on policie

6. Payment Processing :

• Support multiple payment methods: credit/debit cards, online wallets, net banking, cash on delivery.

• Ensure secure transactions using encryption

• Provide payment confirmations via email or SMS.

7. Reviews and Ratings:

• Allow users to reviews and rate the order

8. Notifications:

• Send notifications for order confirmation, shipping, and delivery.

• Promotional message for sales, discounts, or new arrivals.

• Ensure compliance with security and privacy regulations.

9. Security Features :

• Protect user data with encryption and authentication.

20. Design a basic system architecture for a food delivery app.

➡

1. Key Components :-

Frontend :

• Customer App: Allows customers to browse restaurants, order food, and track deliveries

• Restaurant App: Helps restaurants manage menus, accept orders, and update order status

• Delivery App: Enables delivery agents to give orders, navigation ,and update delivery progress

Backend :

• Acts as a bridge between customers, restaurants, and delivery agents.
Database :

• Stores all app data, such as user profiles, menus, orders, payment details, and delivery history.

APIs (Application Programming Interfaces):

• Connect the frontend apps to the backend.

• Handle requests like fetching menus, placing orders, or updating order status.

2. How It Works:

Customer Interaction:

• The customer browses restaurants and selects food

• The app sends the selected items and delivery address to the backend using APIs.

Restaurant Notification:

• The backend forwards the order to the restaurant via the App

• The restaurant confirms and prepares the food.

Delivery Assignment :

• The backend finds an available delivery agent through the Delivery Agent App.

• The agent accepts the delivery job and picks up the food from the restaurant.

Delivery Tracking :

• The backend uses geolocation services to track the agent's location

• The customer can see real-time updates about the delivery status.

Payment and Feedback:

• The customer pays through the app using the payment gateway.

• After delivery, the customer can give feedback or rate the service.

21. Develop test cases for a simple calculator program.

➡

1. Addition
    - ♣ Input: 2 + 3
    - ♣ Outpu: 5
2. Subtraction
    - ♣ Input: 10 - 4
    - ♣ Output: 6
3. Multiplication
    - ♣ Input: 5 * 3
    - ♣ Output: 15
4. Division
    - ♣ Input: 8 / 2
    - ♣ Output: 4

5. Division by Zero

- ♣ Input: 5 / 0
- ♣ Output: Error

22. Document a real-world case where a software application required critical maintenance.

➡

In 2017 Microsoft face critical issue with a windows 10 creator update, which cause system crashes, performance issue and driver compatibility problems for users.