# SOFTWARE ENGINEERING ASSIGNMENT

Overview OF IT Industry

1. **What is program ?**

* Program is a set of Instructions. It is written in a programming language. By a program a computer can do any specific task or solve a problems.

1. **Write a simple hello world program with two different programming languages.**

* C language “hello world” program.

#include <stdio.h>

int main(){

printf(“hello world”);

}

* Python language “hello world” program.

print(“hello, world!”)

1. **Explain in your own words what a program is and how it functions.**

* Program is a set of instructions written by humans in different languages. Program gives directions to do an specific work, and instruct what to do.

1. **What is programming ?**

* Programming is a process of writing code in different languages. That a computer can understand and can do specific tasks by following the code.

1. **What are the key steps involved in the programming process?**

* There are 7 key steps involved in the programming process.

1. Problem definition
2. Planning And design
3. Coding
4. Testing And debugging
5. Deployment
6. Maintenance and Updates
7. Documentation
8. **Types of programming language**.

* There are 4 types of programming language

1. Procedural
2. Object Oriented
3. Logical
4. Functional
5. **What are the main difference between Low level and high level programming languages?**

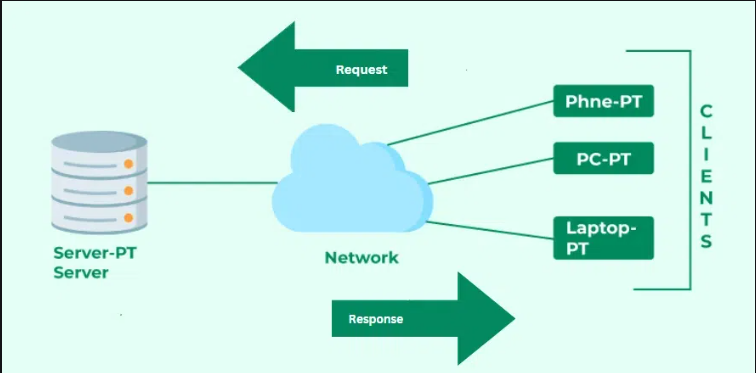
* The main difference between low level and high level languages are the abstraction, ease of use, performance,

And hardware control.

**World Wide Web & How Internet Works**

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

* When a user (client) sends a request (like opening a website), the following steps occur:
* Application Layer (e.g., Browser)
* The client types a URL (like https://www.example.com) in a browser.
* The browser creates an HTTP request.
* DNS Resolution

****

1. **Describe the roles of the client and server in web communication.**

* Client

Role: The client initiate the communication by sending request to a server.

* Server (Provider-Side)

Role: The server responds to client requests by delivering the required resources or services.

1. **Explain the function of the TCP/IP model and its layers.**

* The **TCP/IP model** (Transmission Control Protocol/Internet Protocol) is a **communication framework** used to send and receive data over the internet.  
  It breaks the process of data transmission into **4 layers**, each with a specific role.

1. Explain Client Server Communication.

* Client server communication is a model where two devices communicate.
* The client requests a service or resource.
* The server provides that resource or service.
* This model is used every where on the internet , for browsing websites, using apps, or accessing database.

Types of Internet Connection

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

| **Type** | **Description** |
| --- | --- |
| **Broadband (DSL)** | Uses telephone lines to provide high-speed internet. |
| **Fiber-Optic** | Transmits data as light through glass fastest available. |
| **Cable** | Uses TV coaxial cables for internet; common in urban areas. |
| **Satellite** | Uses satellites in space to deliver internet; good for remote areas. |
| **Mobile (4G/5G)** | Provides internet through cellular networks using SIM cards. |
| **Dial-Up** | Older technology using telephone lines; very slow speeds. |

1. How does broad band different from fiber optic internet ?

* The difference between broad band and fiber optic internet,

**\*Broadband (DSL)** is like a **bicycle on a rough road**—reliable for basic use but not very fast.

**\*Fiber-Optic** is like a **bullet train**—super fast, smooth, and efficient.

Protocols

1. What are the differences between HTTP and HTTPS protocols?

* **HTTP**: Sends data **without protection** (like sending a postcard).
* **HTTPS**: Sends data **safely using encryption** (like sending a sealed envelope).

**Application Security**

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

**(1)SQL Injection.**

**What it is:**

An attacker injects **malicious SQL code** into a form or input field (like a login box) to access or damage the database.

POSSIBLE SOLUTIONS :-

 **Use Prepared Statements** or **Parameterized Queries**

 **Validate and sanitize user input**

 Avoid building SQL queries with raw user data

**(2)Cross-Site Scripting (XSS).**

**What it is:**

An attacker inserts **malicious JavaScript code** into a website, which runs in other users' browsers. This can steal cookies or redirect.

POSSIBLE SOLUTIONS:-

 Escape HTML special characters in user input

 Use Content Security Policy (CSP)

 Sanitize inputs before displaying them on the page

**(3)Broken Authentication**

**What it is:**

Weak or faulty login systems allow attackers to break into user accounts, especially if passwords or tokens are not protected properly.

**POSSIBLE SOLUTIONS:-**

* Use strong password policies
* Implement Multi-Factor Authentication (MFA)
* Secure session tokens (never store them in URLs or share them insecurely)
* Limit login attempts and monitor for brute force attacks

1. **What is the role of encryption in securing applications?**

| * **Role** | **Simple Explanation** |
| --- | --- |

|  |  |
| --- | --- |
| 1. Protects user data | -keeps things like passwords and messages private. |

|  |  |
| --- | --- |
| 2. Secures communication | - Makes sure no one can spy when data moves online. |

|  |  |
| --- | --- |
| 3. Stops hackers from reading data | - Even if stolen, the data looks like random nonsense. |
|  |  |

**SOFTWARE APPLICATIONS AND ITS TYPES**

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

* Here are 5 applications you might use daily, classified as **system software** or **application software**:

| **Application** | **Type** | **Classification** |
| --- | --- | --- |
| Windows OS / macOS / Linux | Operating System | System Software |
| Google Chrome / Safari | Web Browser | Application Software |
| WhatsApp / Telegram | Messaging App | Application Software |
| MS Word / Google Docs | Word Processing Tool | Application Software |
| Antivirus Software (e.g., Quick Heal) | Security Tool | System Software |

1. **What is the difference between system software and application software?**

**System Software:**

* Dition: Software that controls and manages the basic operations of a computer.
* Purpose: Helps the computer hardware and software to work together.

**Application Software:**

* Definition: Software that allows users to perform specific tasks.
* Purpose: Helps users to do activities like writing, browsing, designing, etc.

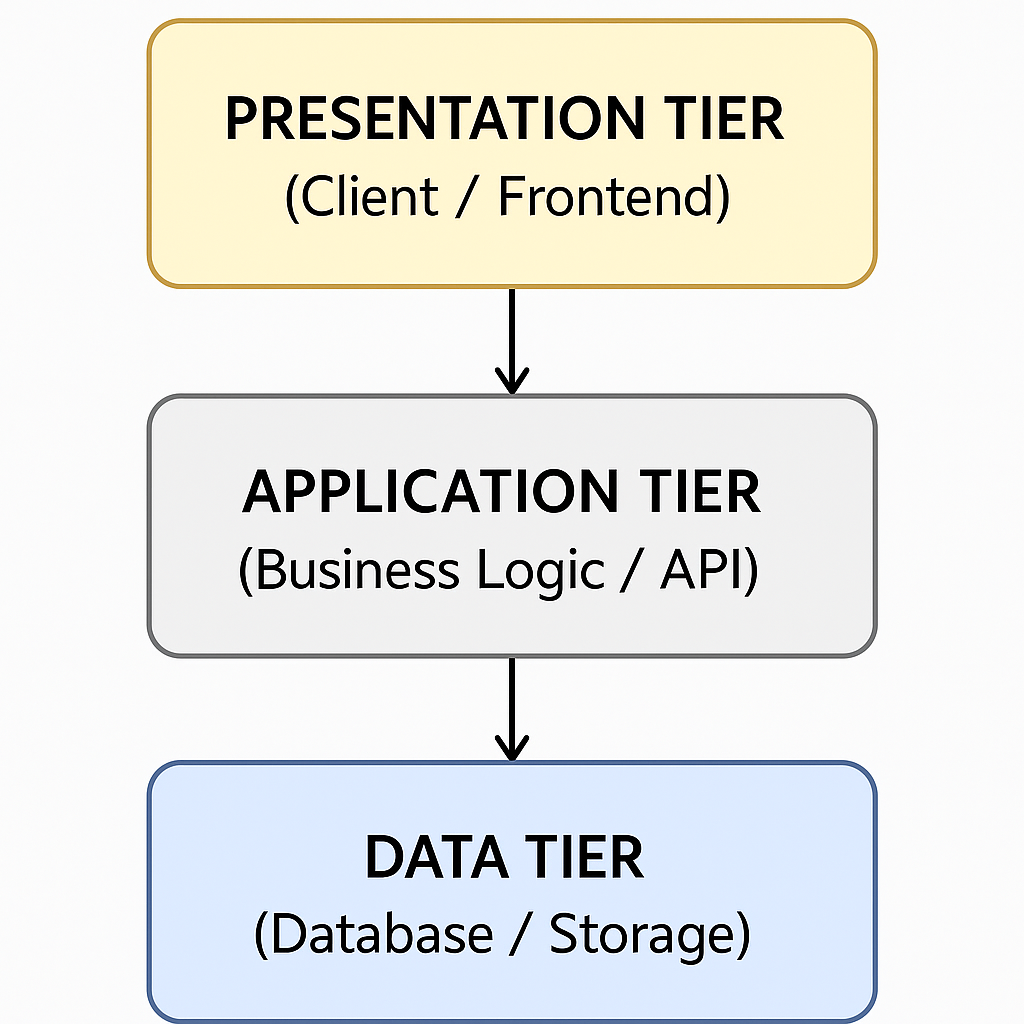
**Software Architecture**

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

**-> Tier Descriptions**

1. Presentation Tier (Frontend)
   * What the user sees and interacts with.
   * Technologies: HTML, CSS, JavaScript, React, Angular, Vue.js, etc.
   * Examples: Web browser, mobile app UI.
2. Application Tier (Backend)
   * Handles business logic, rules, data processing, and communication.
   * Technologies: Node.js, Java (Spring), Python (Django/Flask), .NET, etc.
   * Examples: REST APIs, authentication logic, data validation.
3. Data Tier (Database)
   * Stores and retrieves application data.
   * Technologies: MySQL, PostgreSQL, MongoDB, Oracle, etc.
   * Examples: User data, product listings, transaction history.

DIAGRAM



* 1. **What is the significance of modularity in software architecture?**
* Modularity in software architecture is a foundational design principle that involves dividing a system into smaller, self-contained units called modules. Each module encapsulates a specific part of the system’s functionality and interacts with other modules through well-defined interfaces.

**LAYERS IN SOFTWARE ARCHITECTURE**

* 1. **Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.**
* The application is built using a three-tier architecture:

-Presentation Layer

The presentation layer is the user interface of the system. It interacts directly with users and displays information processed by the business logic layer.

Functions:

* Displays book catalog.
* Allows users to search, filter, and sort books
* Provides login, registration, and profile management UI
* Displays cart contents and checkout forms
* Shows confirmation messages and errors

-Business Logic Layer

This layer contains the application's core functionality. It processes inputs, applies rules, and makes decisions before passing data to or from the data access layer.

-Functions :

* Validates user credentials during login
* Manages cart operations (add/remove/update items)
* Calculates discounts, taxes, and shipping
* Processes orders and payment transactions
* Applies business rules (e.g., age restrictions for certain books)

-Data Access Layer

The data access layer interacts directly with the database. It performs CRUD (Create, Read, Update, Delete) operations on data.

Functions:

* Retrieves book details from the database
* Stores user information and order history
* Updates stock levels after a purchase
* Handles database transactions and rollbacks
  1. **Why are layers important in software architecture?**
* Layers in software Architecture are essential for building structured, maintainable, and scalable applications.
* 1. Separation of corners

-Each layer handles specific responsibility.

Presentation Layer: User interface and user experience.

Business Logic Layer: Core rules and processing.

Data Access Layer: Communication with databases.

* 2. Improved maintainability.
* 3.better scalability
* 4. Code Reusability
* 5. Easier Testing
* 6. Team Collaboration
* 7. Enhanced Security
* 8. Technology Flexibility

**SOFTWARE ENVIRONMENTS**

* 1. **Explain the importance of a development environment in software production.**
* A development environment is the setup (tools, software, and hardware) where developers write, test, and debug code before deploying it. It plays a crucial role in the software development life cycle (SDLC).
* A proper development environment ensures safer, faster, and more organized software production. It reduces bugs, increases productivity, and allows teams to build high-quality software reliably.

**SOURCE CODE**

* 1. :**What is the difference between source code and machine code?**

| * Aspect | Source Code | Machine Code |
| --- | --- | --- |
| Definition | Human-readable code written by programmers | Binary code (0s and 1s) that a computer can execute |
| Written In | High-level languages (C, C++, Java, Python, etc.) | Low-level binary instructions for the CPU |
| Readability | Easy for humans to read and understand | Only understandable by computers |
| Editable By | Programmers | Not normally edited directly by humans |
| Execution | Needs to be compiled or interpreted to run | Runs directly on the computer hardware |
| Stored As | Text files with extensions like .c | Executable files like .exe, .bin, .class |

**TYPES OF SOFTWARE**

* 1. **What are the differences between open-source and proprietary software.**

| * **Aspect** | * **Open-Source Software** | * **Proprietary Software** |
| --- | --- | --- |
| Source Code Access | Source code is freely available to view, modify, and share | Source code is closed and protected |
| License | Distributed under open licenses (e.g., MIT, GPL) | Controlled by a company with restricted licenses |
| Cost | Usually free of cost | Often paid or subscription-based |
| Customization | Can be modified by anyone | Cannot be modified without vendor permission |
| Support | Community-based support (forums, GitHub, etc.) | Official vendor support (customer service, updates) |
| Examples | Linux, LibreOffice, Firefox, VLC | Windows, Microsoft Office, Adobe Photoshop |
| Security | Transparent code; bugs can be fixed by anyone | Hidden code; users must trust the vendor for fixes |
| Updates | Community or developer-driven | Vendor controls all updates and releases |

**APPLICATION SOFTWARE**

* 1. **Write a report on the various types of application software and how they improve productivity.**
* Application software is a type of computer program designed to help users perform specific tasks. These applications play a crucial role in both personal and professional environments, helping users become more efficient and productive.

Application software benefits:

-Speeds up writing and editing tasks

-Offers spell check, auto-correct, and templates

-Enables collaboration through real-time editing (e.g., Google Docs)

-Automates complex calculations using formulas

-Supports data visualization for better decision-making

-Useful for managing budgets, attendance, and reports

-Helps deliver information clearly and visually

-Saves time with ready-made templates

-Improves communication in meetings and classrooms

-Centralizes data storage and access

-Reduces data duplication and errors

-Speeds up record management and reporting

-Enables remote work and instant communication

-Reduces need for in-person meetings

-Increases teamwork efficiency

- Speeds up content creation

* 1. **What is the role of application software in businesses?**
* Application software plays a **vital role** in helping businesses operate efficiently, communicate effectively, and achieve their goals. It allows companies to automate tasks, manage data, and improve productivity.

1. Increases Productivity
2. Enhances Communication
3. Data Management and Analysis
4. Supports Business Operations
5. Improves Customer Experience
6. Enables Remote Work
7. Project and Time Management

**SOFTWARE DEVELOPMENT PROCESS**

**Designing**

* 1. **:What role does UI/UX design play in application development?**
* UI (User Interface) focuses on how the app looks,  
  while UX (User Experience) focuses on how the app works.  
  Together, they ensure the app is not just functional but also enjoyable and efficient to use.

**Mobile Application**

* 1. **What are the differences between native and hybrid mobile apps?**
* **Native Apps**

Definition:-Built specifically for one platform using platform-specific programming languages.

iOS:- Uses Swift or Objective-C

Android:- Uses Kotlin or Java

**DFD(Data Flow Diagram)**

* 1. **What is the significance of DFDs in system analysis?**
* **Data Flow Diagrams (DFDs)** are a key tool in **system analysis**. They help analysts and developers understand how data moves through a system. Here's the significance of DFDs in system analysis.
* 1. Visual Representation of the System
* 2. Helps in Understanding System Requirements
* 3. Simplifies Complex Systems
* 4. Basis for System Design
* 5.Improves Communication

**Desktop Application**

* 1. **What are the pros and cons of desktop applications compared to web applications?**

**Pros:**

-Accessible anywhere:

-Runs in a browser; no installation needed.

-Platform-independent:

-Works on any OS (Windows, Mac, Linux, mobile) with a browser.

-Easy to update:

-Developers can deploy updates instantly for all users.

-No local storage needed:

-Saves space; often uses cloud-based data.

**Cons:**

-Internet required:

-Cannot work without an active connection (unless it's a PWA).

-Performance depends on connection:

-Slower on low-speed networks.

-Security concerns:

-More exposed to web-based attacks.

-Limited hardware access:

-Restricted access to devices like printers, webcams (unless permissions are given).

**Flow Chart**

* 1. **How do flowcharts help in programming and system design?**
* **Flowcharts are powerful tools in programming and system design because they visually map out the logic and flow of a system or program.**

-1. Clear Understanding of Logic

-2. Aids in Problem Solving

-3. Supports System Design

-4. Easy Debugging and Testing

-5. Improves Communication

-6. Helps in Maintenance and Updates