**Module 1 – Overview of IT Industry**

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

Answer: In Python

print("Hello, World!");

In C

#include <stdio.h>

int main() {

printf("Hello, World!\n");

return 0;

}

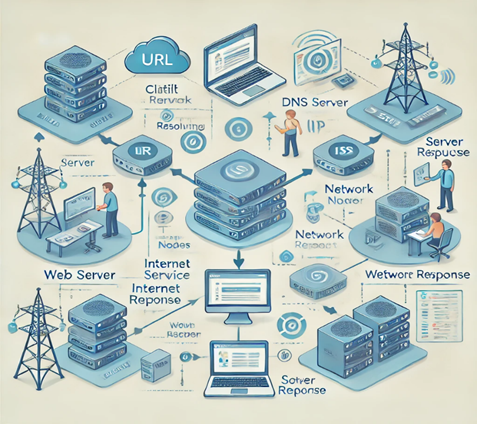
| **Feature** | **Python** | **C** |
| --- | --- | --- |
| **Simplicity** | Very concise (1 line) | Requires multiple lines, including header and main function |
| **Compilation** | Interpreted (no compilation needed) | Compiled (requires a compiler like GCC) |
| **Syntax** | No semicolons, indentation-based | Requires semicolons, curly braces for code blocks |
| **Function Requirement** | No need for a function to print | Requires main() function as an entry point |
| **Standard Library** | print() is built-in | Needs <stdio.h> for printf() |

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

Answer: **Steps in Data Transmission from Client to Server**

1. **User Request (Client-Side)**
   * The user enters a URL (e.g., www.example.com) in a web browser.
   * The browser checks its cache for the IP address; if not found, it contacts the **DNS (Domain Name System)** server.
2. **DNS Resolution**
   * The DNS server translates the domain name into an **IP address** (e.g., 192.168.1.1).
   * The browser sends the request to the obtained IP address.
3. **Data Packet Formation**
   * The request is divided into **small packets** of data.
   * Each packet is assigned a header containing:
     + Source IP Address (Client)
     + Destination IP Address (Server)
     + Sequence Number
4. **Routing Through the Internet**
   * The packets travel through multiple network devices such as:
     + **Router** (Directs traffic between networks)
     + **Switches** (Manage local network communication)
     + **ISP (Internet Service Provider)** (Provides internet access)
5. **Reaching the Web Server**
   * The request reaches the **web server** hosting the website.
   * The server processes the request and prepares a response.
6. **Server Response**
   * The server sends the response back to the client.
   * The response is divided into **packets** and follows the reverse path.
7. **Data Reassembly & Display**
   * The browser reassembles the packets and renders the web page for the user.

Diagram of Data Transmission from Client to Server



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

**Applications and their classification:**

1. **Microsoft Windows**
   * **Type**: System Software
   * **Reason**: Windows is an operating system that manages hardware resources and provides a platform for running application software.
2. **Google Chrome**
   * **Type**: Application Software
   * **Reason**: It is a web browser designed for browsing the internet and is used by end-users for specific tasks.
3. **Microsoft Word**
   * **Type**: Application Software
   * **Reason**: A word processor used for creating and editing text documents, designed specifically for users.
4. **Antivirus Software (e.g., Avast, McAfee)**
   * **Type**: System Software
   * **Reason**: Provides system-level security by protecting the computer from malware and viruses.
5. **Windows File Explorer**
   * **Type**: System Software
   * **Reason**: Helps users manage files, folders, and storage, working directly with the operating system.

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

1. **Three-Tier Architecture Diagram**:

--------------------------------------------------

| Presentation Layer |

| (User Interface, Web Browser, etc.) |

--------------------------------------------------

|

v

--------------------------------------------------

| Business Logic Layer |

| (Application Server, Web Server, APIs) |

--------------------------------------------------

|

v

--------------------------------------------------

| Data Access Layer |

| (Database Server, Data Storage) |

--------------------------------------------------

**Explanation**:

1. **Presentation Layer**: This layer is responsible for displaying the UI and interacting with the user (e.g., a web browser displaying web pages).
2. **Business Logic Layer**: It contains the core business logic and rules for processing data, which could be served via APIs or application servers.
3. **Data Access Layer**: This layer handles the storage and retrieval of data from a database. It typically involves databases like MySQL, PostgreSQL, etc.

### **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: E-Commerce Application**

1. **Presentation Layer**:
   * **Functionality**: This layer provides the user interface where customers browse products, add items to the cart, view orders, and complete the checkout process. Technologies used could include HTML, CSS, JavaScript, and frameworks like React or Angular.
2. **Business Logic Layer**:
   * **Functionality**: This layer processes business rules such as calculating product prices, handling discounts, verifying user authentication, and processing orders. The backend might be powered by technologies like Node.js, Django, or Ruby on Rails.
3. **Data Access Layer**:
   * **Functionality**: Manages all interactions with the database, such as storing user profiles, order information, inventory management, and transaction records. This layer could involve relational databases (MySQL, PostgreSQL) or NoSQL databases (MongoDB).

**Application Software**

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

**Report on Application Software Types and Productivity Improvement**Application software refers to programs designed to perform specific tasks for users, and they can be broadly classified into the following categories:

**1. Productivity Software:** Includes word processors (Microsoft Word), spreadsheet software (Microsoft Excel), and presentation software (Microsoft PowerPoint). These tools help employees create documents, analyze data, and present information effectively.

**2. Enterprise Software:** Applications like ERP (Enterprise Resource Planning) systems, CRM (Customer Relationship Management) systems, and HRM (Human Resource Management) systems are designed to manage business processes across an organization.

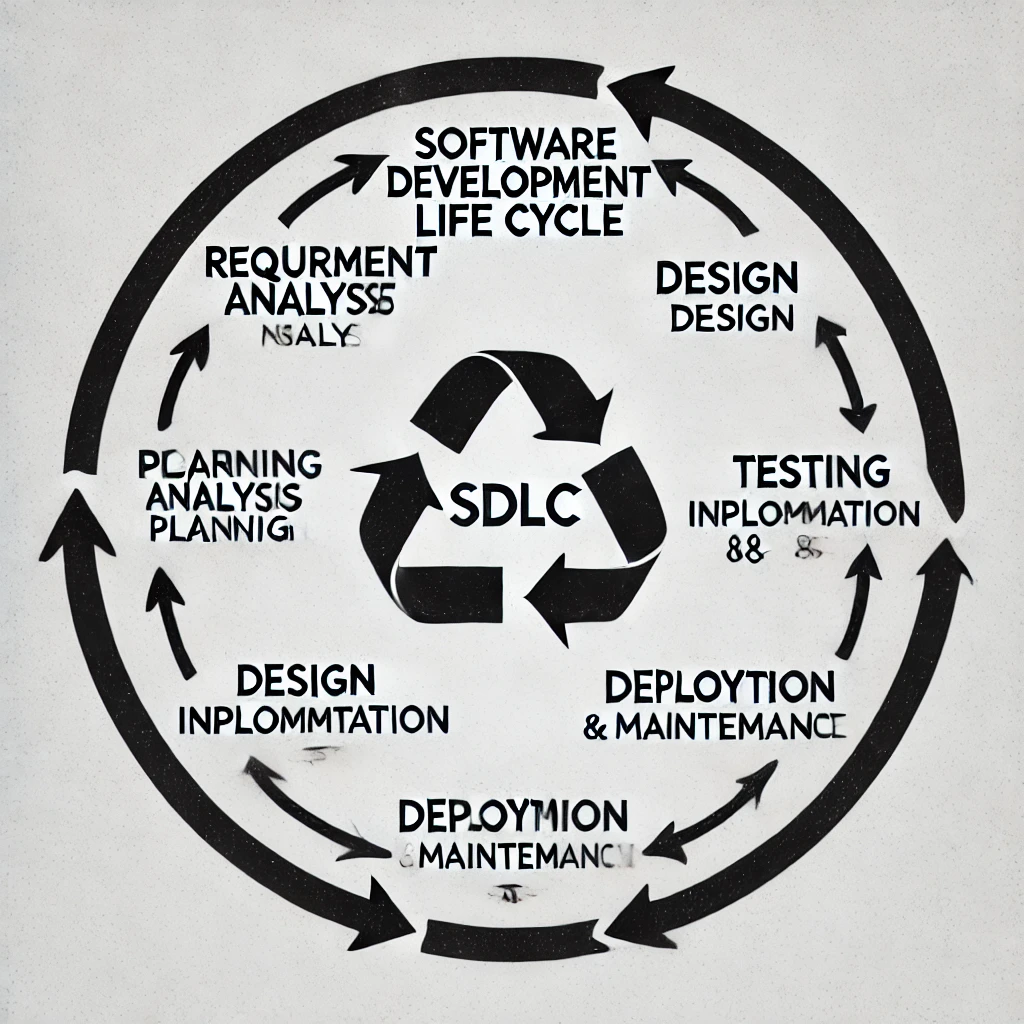
**3. Communication Software:** Includes email clients (Microsoft Outlook), messaging apps (Slack, Teams), and video conferencing tools (Zoom, Google Meet). They improve communication among team members and with clients, thus boosting overall productivity.

**4. Design and Development Software:** Graphic design tools (Adobe Photoshop, Illustrator), video editing software (Adobe Premiere Pro), and IDEs (Integrated Development Environments like Visual Studio) enable developers and designers to create, edit, and refine digital content more efficiently.

**How it Improves Productivity:**Application software automates routine tasks, enhances collaboration, simplifies complex processes, and provides better data management, ultimately increasing efficiency and reducing time spent on manual work.

**Software Development Process**

**LAB EXERCISE:** Create a flowchart representing the Software Development Life Cycle (SDLC).

****

**Software Requirement**

**Q: Write a requirement specification for a simple library management system.**

**A:**

A **Library Management System (LMS)** helps manage books, members, and transactions efficiently.

**Functional Requirements:**

**1. User Management** – Add, edit, and delete users (students, staff).

**2. Book Management** – Add, search, issue, and return books.

**3. Transaction Management** – Track issued/returned books.

**4. Reports** – Generate due date and fine reports.

**Non-Functional Requirements:**

**· Security** – Only authorized users can access the system.

**· Usability** – Easy-to-use interface.

**· Performance** – Quick book searches and updates.

**Software Analysis**

**Q: Perform a functional analysis for an online shopping system.**

**A:**

An **Online Shopping System** allows users to browse and purchase products.

**Functional Components:**

**1. User Registration/Login** – Secure user authentication.

**2. Product Catalog** – View product listings and details.

**3. Shopping Cart** – Add/remove products.

**4. Order Management** – Place, track, and cancel orders.

**5. Payment Processing** – Secure online payments.

**6. Reviews & Ratings** – Allow users to rate products.

**System Design**

**Q: Design a basic system architecture for a food delivery app.**

**A:**

A **Food Delivery App** consists of:

**1. User Interface (Frontend)** – Mobile app/web interface for customers.

**2. Backend Server** – Manages orders, payments, and restaurant data.

**3. Database** – Stores user, order, and restaurant details.

**4. Delivery Module** – Assigns delivery agents and tracks orders.

**5. Payment Gateway** – Processes payments securely.

**Software Testing**

**Q: Develop test cases for a simple calculator program.**

**A:**

| **Test Case** | **Input** | **Expected Output** |
| --- | --- | --- |
| Addition | 5 + 3 | 8 |
| Subtraction | 9 - 4 | 5 |
| Multiplication | 6 × 2 | 12 |
| Division | 8 ÷ 2 | 4 |
| Divide by Zero | 7 ÷ 0 | Error |

**Maintenance**

**Q: Document a real-world case where a software application required critical maintenance.**

**A:**

A well-known **maintenance case** is the **Windows 10 Updates Issue (2018)** where a system update accidentally deleted user files. Microsoft had to roll back the update and release a fix.

**Key Lessons:**

**· Thorough testing is necessary before release.**

**· User feedback helps identify critical issues.**

**· Quick response reduces damage.**

**Data Flow Diagram (DFD)**

**Q: Create a DFD for a hospital management system.**

**+---------------------+**

**| Patient System |**

**+---------------------+**

**│**

**▼**

**+-----------------------------+**

**| Appointment System |**

**+-----------------------------+**

**│**

**▼**

**+---------------------+**

**| Doctor System |**

**+---------------------+**

**│**

**▼**

**+---------------------+**

**| Billing System |**

**+---------------------+**

**Desktop Application**

**Q: Build a simple desktop calculator application using a GUI library.**

**A:**A simple calculator can be built using **Python (Tkinter)** or **Java (Swing)** with buttons for basic operations like addition, subtraction, multiplication, and division.

**Flow Chart**

**LAB EXERCISE:** Draw a flowchart representing the logic of a basic online registration system.

1. A flowchart representing the logic of a basic online registration system :

**+----------------------+**

**| Start |**

**+----------------------+**

**↓**

**+----------------------+**

**| Enter User Details |**

**+----------------------+**

**↓**

**+----------------------+**

**| Validate Input |**

**+----------------------+**

**↓**

**+----------------------+**

**| Is Email Unique? |**

**+----------------------+**

**/ \**

**Yes No**

**↓ ↓**

**+----------------------+**

**| Save to Database |**

**+----------------------+**

**↓**

**+----------------------+**

**| Registration Success |**

**+----------------------+**

**↓**

**+----------------------+**

**| End |**

**+----------------------+**