# Lab Exercises - 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.

Solution:

Language 1: Python  
Code:  
print("Hello, World!")  
  
Output:  
Hello, World!  
  
Language 2: C  
Code:  
#include <stdio.h>  
int main() {  
 printf("Hello, World!");  
 return 0;  
}  
  
Output:  
Hello, World!  
  
Comparison:  
- Python is simpler with one line of code.  
- C requires boilerplate (headers, main function, return).

## 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.

Solution:

Steps of Data Transmission:  
1. Client sends request (HTTP) to server via browser.  
2. Request passes through network layers (Application → Transport → Network → Data Link → Physical).  
3. Server receives the request, processes it, and sends back response.  
4. Response passes through same layers in reverse and reaches client.  
  
Diagram (simplified representation):  
Client → ISP → Internet Backbone → Web Server → Response back to Client

## Network Layers on Client and Server

Lab Exercise: Design a simple HTTP client-server communication in any language.

Solution:

Python Example:  
  
# Server Code  
import socket  
server = socket.socket()  
server.bind(('localhost', 8080))  
server.listen(1)  
print("Server listening...")  
conn, addr = server.accept()  
print("Connected by", addr)  
conn.send(b"Hello from Server!")  
conn.close()  
  
# Client Code  
import socket  
client = socket.socket()  
client.connect(('localhost', 8080))  
print(client.recv(1024).decode())  
client.close()  
  
Expected Output:  
Server side: Connected by ('127.0.0.1', <port>)  
Client side: Hello from Server!

## Types of Internet Connections

Lab Exercise: Research different types of internet connections and list their pros and cons.

Solution:

1. Broadband:  
 Pros: Fast, reliable, always-on connection.  
 Cons: Availability depends on location.  
  
2. Fiber:  
 Pros: Very high speed, low latency.  
 Cons: Expensive, limited availability.  
  
3. Satellite:  
 Pros: Available in remote areas.  
 Cons: High latency, weather dependent.  
  
4. Mobile Data (4G/5G):  
 Pros: Portable, flexible.  
 Cons: Data limits, network coverage issues.

## Protocols

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

Solution:

Example using curl (HTTP):  
curl http://example.com  
  
Output:  
<html> ... </html>  
  
Example using curl (FTP):  
curl ftp://speedtest.tele2.net/1MB.zip -o file.zip  
  
Output:  
File downloaded successfully.

## Application Security

Lab Exercise: Identify and explain three common application security vulnerabilities. Suggest possible solutions.

Solution:

1. SQL Injection:  
 Vulnerability: Attackers insert malicious SQL queries.  
 Solution: Use prepared statements.  
  
2. Cross-Site Scripting (XSS):  
 Vulnerability: Injection of malicious scripts in web pages.  
 Solution: Validate and sanitize inputs.  
  
3. Weak Passwords:  
 Vulnerability: Easy to guess or brute-force.  
 Solution: Enforce strong password policies, use MFA.

## Software Applications and Its Types

Lab Exercise: Identify and classify 5 applications you use daily as either system software or application software.

Solution:

1. Windows OS → System Software  
2. MS Word → Application Software  
3. Google Chrome → Application Software  
4. Antivirus (Avast) → System Utility Software  
5. Photoshop → Application Software

## Software Architecture

Lab Exercise: Design a basic three-tier software architecture diagram for a web application.

Solution:

Three-Tier Architecture:  
1. Presentation Layer (User Interface – Web browser)  
2. Business Logic Layer (Server-side scripts – Python, PHP, Java)  
3. Data Layer (Database – MySQL, MongoDB)  
  
Diagram Representation:  
[Client UI] ↔ [Application Server] ↔ [Database]

## 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.

Solution:

Case Study: Online Shopping System  
- Presentation Layer: User views products, adds to cart.  
- Business Logic Layer: Handles discounts, payment validation.  
- Data Access Layer: Fetches product details from database, updates stock.

## Software Environments

Lab Exercise: Explore different types of software environments (development, testing, production).

Solution:

1. Development Environment: Programmers write and test code.  
2. Testing Environment: QA team runs test cases.  
3. Production Environment: Final live system for end users.  
  
Example: Using VirtualBox to create separate environments.

## Source Code

Lab Exercise: Write and upload your first source code file to Github.

Solution:

Example: Python file (hello.py)  
print("Hello GitHub")  
  
Steps to upload:  
1. git init  
2. git add hello.py  
3. git commit -m "First commit"  
4. git remote add origin <repo\_url>  
5. git push -u origin main

## Github and Introductions

Lab Exercise: Create a Github repository and document how to commit and push code changes.

Solution:

Steps:  
1. Create repository on GitHub.  
2. git init  
3. git add .  
4. git commit -m "Initial commit"  
5. git branch -M main  
6. git remote add origin <repo\_url>  
7. git push -u origin main

## Student Account in Github

Lab Exercise: Create a student account on Github and collaborate on a small project with a classmate.

Solution:

Steps:  
1. Sign up at github.com/education.  
2. Verify student email ID.  
3. Create repository and invite collaborator.  
4. Use git clone, push, pull for collaboration.

## Types of Software

Lab Exercise: Create a list of software you use regularly and classify them.

Solution:

System Software: Windows 11  
Application Software: MS Word, Chrome, WhatsApp  
Utility Software: Antivirus, WinRAR

## GIT and GITHUB Training

Lab Exercise: Follow a GIT tutorial to practice cloning, branching, and merging repositories.

Solution:

Steps:  
1. git clone <repo\_url>  
2. git checkout -b new-feature  
3. Make changes and commit.  
4. git merge new-feature into main.

## Application Software

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

Solution:

Types:  
1. Word Processing Software → MS Word  
2. Spreadsheet Software → Excel  
3. Database Software → MySQL  
4. Web Browser → Chrome  
5. Communication Tools → Zoom  
  
Productivity: They save time, automate tasks, improve collaboration.

## Software Development Process

Lab Exercise: Create a flowchart representing the Software Development Life Cycle (SDLC).

Solution:

Stages:  
1. Requirement Analysis  
2. System Design  
3. Implementation  
4. Testing  
5. Deployment  
6. Maintenance  
  
Flowchart: (Linear representation)  
Requirement → Design → Implementation → Testing → Deployment → Maintenance

## Software Requirement

Lab Exercise: Write a requirement specification for a simple library management system.

Solution:

Requirements:  
- Add/Delete books  
- Issue/Return books  
- Maintain student records  
- Generate reports

## Software Analysis

Lab Exercise: Perform a functional analysis for an online shopping system.

Solution:

Functional Requirements:  
- Browse products  
- Add to cart  
- Checkout and payment  
- Track orders

## System Design

Lab Exercise: Design a basic system architecture for a food delivery app.

Solution:

Architecture:  
- Client: Mobile app for ordering food  
- Server: Processes orders, payments  
- Database: Stores menu, user details, orders

## Software Testing

Lab Exercise: Develop test cases for a simple calculator program.

Solution:

Test Cases:  
1. Input: 2 + 3 → Expected Output: 5  
2. Input: 10 - 4 → Expected Output: 6  
3. Input: 6 \* 3 → Expected Output: 18  
4. Input: 8 / 2 → Expected Output: 4

## Maintenance

Lab Exercise: Document a real-world case where a software application required critical maintenance.

Solution:

Example: WhatsApp outage (2021)  
- Cause: Server configuration error  
- Solution: Engineers updated server settings  
- Importance: Ensured billions of users could reconnect

## DFD (Data Flow Diagram)

Lab Exercise: Create a DFD for a hospital management system.

Solution:

DFD:  
- Patient → Registration → Database  
- Doctor → Diagnosis → Database  
- Admin → Billing → Database

## Desktop Application

Lab Exercise: Build a simple desktop calculator application using a GUI library.

Solution:

Python Tkinter Example:  
from tkinter import \*  
root = Tk()  
root.title("Calculator")  
entry = Entry(root)  
entry.pack()  
root.mainloop()  
  
Expected Output: A simple GUI window with an input box.

## Flow Chart

Lab Exercise: Draw a flowchart representing the logic of a basic online registration system.

Solution:

Flowchart Steps:  
Start → Enter User Details → Validate Data → Store in Database → Confirmation Message → End