Assignment-1

Module-1-se-Overview-of-IT-Industry

1. What is a Program?

LAB EXERCISE:

=> program java 1

```
 class Main {
 public static void main(String[] args) {
    System.out.println("Hello Word");
}
}
```

Output: Hello Word

=> program c

```
 #include <stdio.h>
int main() {
   // Write C code here
   printf("Hello Word");

   return 0;
}
```

Output: Hello Word

4. The World Wide Web and Internet Mechanics

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

```
        [ User Device ]
              |
              v
   |    The Internet       | <-- Infrastructure: Cables, Routers, Protocols (TCP/IP)
              |
              v
   |  World Wide Web (WWW)  | <-- Websites, Web Pages, HTML, HTTP/HTTPS
              |
              v
      [ Web Browser ]
```

- **Internet** is the infrastructure that connects computers.
- **WWW** is a service that runs on the Internet (like email, FTP).
- A **web browser** accesses web pages using **HTTP/HTTPS**.

5. Network Layers on Client and Server

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

```
import requests

nums = [4, 2, 7, 1]

res = requests.post("http://localhost:8000", json=nums)

print("Sorted:", res.json())
```

Output: `Sorted: [1, 2, 4, 7]`

7.Types of Internet Connections

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

1. Fiber Optic Internet

Pros:

Extremely fast speeds (up to 1 Gbps or more)

Very low latency (ideal for gaming/streaming)

More reliable and consistent connection

Future-proof technology

Cons:

Limited availability in rural areas

Installation can be expensive and time-consuming

2. Cable Internet

Pros:

Fast speeds (up to 1 Gbps in some areas)

Widely available in urban and suburban areas

Bundling with TV services possible

Cons:

Shared bandwidth can slow speeds during peak times

Higher latency than fiber.

3. DSL (Digital Subscriber Line)

Pros:

More widely available than fiber or cable

Dedicated line (no slowdowns from neighbors)

Usually affordable

Cons:

Slower speeds (especially upload)

Speed decreases the farther you are from the provider's station

4. Satellite Internet

Pros:

Available almost everywhere, including remote areas

Doesn't rely on phone or cable lines

Cons:

High latency (not ideal for gaming or video calls)

Data caps are common

Weather can affect performance

5. Fixed Wireless Internet

Pros:

Good alternative for rural areas

No need for cables or phone lines

Faster than satellite in many cases

Cons:

Line of sight to tower required

Speeds can vary due to weather and interference

6. Dial-Up Internet

Pros:

Extremely cheap

Still available in some very remote areas

Cons:

Very slow speeds (~56 Kbps)

Ties up phone line

Outdated for modern usage

7. Mobile (Cellular) Internet (3G/4G/5G)

Pros:

Portable and flexible

Fast speeds with 4G/5G

Increasingly available

Cons:

Data caps and overage fees

Signal strength varies by location

Can be expensive for heavy usage

8. Network Protocols

LAB EXERCISE:

```
curl http://example.com
curl ftp://speedtest.tele2.net
```

9. Application Security

LAB EXERCISE:

SQL Injection → Employ prepared statements

Cross-Site Scripting (XSS) → Implement input validation

Cross-Site Request Forgery (CSRF) → Utilize anti-CSRF tokens

10.Software Applications and Its Types

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

Google Chrome – Application Software

Microsoft Word – Application Software

Windows 11 – System Software

Spotify – Application Software

Antivirus Program (e.g., Windows Defender) – System Software

11. Software Architecture

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

```
|    Presentation Tier     |
|  Web Browser (React) |
|  HTML/CSS/JavaScript |
           |
           v
|    Application Tier      |
|  Backend Server (Node.js) |
|  API / Business Logic    |
           |
           v
|       Data Tier        |
|  Database (PostgreSQL) |
|  Data Storage Layer      |
```

Presentation Tier handles user interaction.

Application Tier processes logic and rules.

Data Tier manages data persistence.

12. Layered Architecture Case Study

LAB EXERCISE: Presentation: HTML/CSS interface
Logic: Server-side processing (e.g., PHP, Python)
Data Access: SQL-based database interaction

13. Software Environments

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

Development Environment – Where developers write and debug code; includes IDEs, local servers, and libraries.

Testing Environment – Used to run test cases and find bugs before release; simulates real-world usage.

Production Environment – The live system users interact with; stable, secure, and performance-optimized.

Deploy an Ubuntu-based virtual machine equipped with development tools (e.g., VS Code, Python)

14. Source Code

LAB EXERCISE: Create and push a code repository to GitHub

15. GitHub Basics

LAB EXERCISE:

```
git init
git add .
git commit -m "Initial commit"
git push origin main
```

16. Student Account in Github

LAB EXERCISE: Create a student account on Github and collaborate on a small project with aclassmate.

Sign up on GitHub
Go to github.com and create an account. Then apply for a **student pack** at education.github.com.

Create a New Repository
Click the **"+" > New repository**, name it (e.g., `my-project`), and add a README.

Invite a Classmate
Go to **Settings > Collaborators**, and add your classmate by their GitHub username.

Clone and Work Together
Use Git or GitHub Desktop to clone the project, make changes, and push them.

Submit the Project
Both students contribute code, and the final version stays in the shared repository.

17. Types of Software

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

System Software: Windows 11, macOS, Android OS, BIOS.

Application Software: MS Word, Chrome, Zoom, Spotify, WhatsApp.

Utility Software: Windows Defender, CCleaner, WinRAR, Disk Cleanup.

System software runs the hardware and supports apps.

Utility software maintains and optimizes system performance.

18. Git Fundamentals

LAB EXERCISE: Follow a GIT tutorial to practice cloning, branching, and merging repositories.

Git installed: Download Git

GitHub account (or use any public Git repo)

1. Clone a Repository

```
git clone https://github.com/YOUR-USERNAME/YOUR-REPO.git
cd YOUR-REPO
```

Replace the URL with your actual GitHub repo.

2. Create and Switch to a New Branch

```
git checkout -b feature-branch
```

feature-branch is your new branch name.

3. Make Changes

Edit a file or create a new one:

```
echo "Hello from feature branch" >> hello.txt
```

Stage and commit your changes:

```
git add hello.txt
git commit -m "Add hello.txt on feature-branch"
```

4. Switch Back and Merge

```
git checkout main
git merge feature-branch
```

19. Application Software in Business

LAB EXERCISE: Software Purpose MS Word Text Processing Excel Data Analysis Chrome Internet Browsing Zoom Video Conferencing Photoshop Multimedia Editing

20. Software Development Process

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

```
1. Requirement
   Gathering &
    Analysis
        |
        v
 2. System Design
        |
        v
 3. Implementation
       (Coding)
        |
        v
 4. Testing
        |
        v
 5. Deployment
        |
        v
6. Maintenance
```

21. Requirements Engineering

LAB EXERCISE: Specification Document for Library Management:
Functionalities: Book Addition, Search, Issue, Return
User Roles: Admin, Student

22. Software Analysis

LAB EXERCISE: Perform a functional analysis for an online shopping system.

Users can register, log in, and manage their profiles.

Customers browse products, add to cart, and make payments.

The system allows order tracking, cancellations, and reviews.

Admins manage users, products, orders, and promotions.

Features include search filters, notifications, and ratings

23. System Design

LAB EXERCISE:

Diagram:

```
       [User]
          |
          v
     [Frontend App]
          |
          v
   [Order Management]
      /       |       \
     v        v        v
```

```
[Admin] [Delivery] [Restaurant]
          |
          v
     [Delivery Tracker]
```

24. Software Testing

LAB EXERCISE: Develop test cases for a simple calculator program.

Test Case 1 – Addition:
Input: `2 + 3`
Expected Output: `5`

Test Case 2 – Subtraction:
Input: `10 - 4`
Expected Output: `6`

Test Case 3 – Multiplication:
Input: `7 * 5`
Expected Output: `35`

Test Case 4 – Division:
Input: `20 / 4`
Expected Output: `5`

Test Case 5 – Division by Zero:
Input: `8 / 0`
Expected Output: `Error: Division by zero`

25.Maintenance

LAB EXERCISE: Document a real-world case where a software application required critical maintenance.

In 2017, Amazon S3 experienced a major outage due to a mistyped maintenance command.
The error caused critical systems to go offline, disrupting major websites and services.
AWS resolved the issue and implemented safeguards to prevent similar incidents.

30. Data Flow Diagrams (DFD)

LAB EXERCISE: Create a DFD for a hospital management system.

DFD:

```
[Patient]
   |
   v
[Reception] ---> [Doctor]
                    |
                    v
                [Pharmacy]
```

31. Desktop Applications

LAB EXERCISE: Build a simple desktop calculator application using a GUI library.

```
import tkinter as tk

root = tk.Tk(); e = tk.Entry(root); e.pack(); l = tk.Label(root); l.pack()

tk.Button(root, text="Sort", command=lambda: l.config(text=','.join(map(str, sorted(map(int, e.get().split(','))))))).pack()

root.mainloop()
```

32. Flow charts

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

```
[Start]

  |

  v

[Display Registration Form]

  |

  v

[User Enters Details]

  |

  v

[Validate Input Data]----->(Invalid?)

      |                 |

     No                  Yes

      |                 |

      v                  v

[Check If User Already Exists] <---[Show Error Message and Reload Form]

      |

      v

   (Exists?)

    /   \

  Yes   No

  /       \

 v         v

[Show Error]  [Create User Account]

  |            |
  v         v
[Reload Form]   [Send Confirmation Email]
           |
           v
        [Show Success Message]
           |
           v
         [End]
```