# Module-1

## OVERVIEW OF IT INDUSTRY

# 1. Hello World in Two Languages:

- Python:

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

- C:

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

# 2. Diagram of Client-Server Data Transmission:

- A client (browser) sends a request → Router → Internet → Web Server → Server processes it → Sends back response → Client receives data. (Diagram in PPT)

# 3. HTTP Client-Server Example in Python:

- Server:
- from http.server import BaseHTTPRequestHandler, HTTPServer
- class Handler(BaseHTTPRequestHandler):
- def do_GET(self):
- self.send_response(200)
- self.end_headers()
- self.wfile.write(b"Hello from server")
- HTTPServer(("localhost", 8000), Handler).serve_forever()
- Client:
- import requests
- response = requests.get("http://localhost:8000")
- print(response.text)

# 4. Internet Connection Types:

- Broadband: Common, stable; not very fast.
- Fiber-optic: Very fast, reliable; expensive.
- Satellite: Useful in remote areas; slower and affected by weather.

# 5. Simulate HTTP and FTP with curl:

- HTTP: curl https://example.com
- FTP: curl ftp://ftp.example.com/file.txt

# 6. Application Security Vulnerabilities:

- SQL Injection → Solution: Use parameterized queries.
- Cross-Site Scripting (XSS) → Escape user input.
- Insecure Passwords → Use strong hashing and 2FA.

# 7. Classify Applications:

- Chrome: Application
- Windows OS: System
- MS Word: Application
- Antivirus: Utility
- File Explorer: System

# 8. Three-Tier Architecture:

- Presentation (UI)
- Business Logic (Rules/Processing)
- Data Access Layer (Database)

# 9. Case Study:

- Library Management System:
- UI: Book search, borrow/return
- Logic: Due dates, fines
- Data: Books, users, transaction records

# 10. Software Environments:

- Development: For writing code (e.g., VS Code)
- Testing: Bugs and performance (test servers)
- Production: Live software used by users

# 11. Upload First Code to GitHub:

- Create GitHub account
- Create new repo
- Push code using terminal:

```
git init
git add .
git commit -m "first commit"
git remote add origin <repo-url>
git push -u origin main
```

# 12. Document Git Commit Process:

- Steps: git init, git add, git commit, git push → Use GitHub to track changes.

# 13. Collaborate on GitHub:

- Create repo → Invite classmate as collaborator → Both commit code → Use branches and pull requests.

# 14. Classify Software:

- System: Windows, BIOS
- Application: Chrome, Word
- Utility: Antivirus, WinRAR

# 15. Git Practice:

- Use Git to:
- Clone: git clone
- Branch: git checkout -b feature
- Merge: git merge feature

# 16. Application Software Report:

- Helps in:
- Communication (Zoom)
- Productivity (MS Office)
- Design (Canva)
- Data Analysis (Excel)

# 17. SDLC Flowchart:

- Stages:
- Requirement
- Design
- Development
- Testing
- Deployment
- Maintenance
  (Flowchart in PPT)

# 18. Library System Requirement Spec:

- Search books
- Issue/Return books
- Track users
- Admin panel

# 18. Library System Requirement Spec:

- Search books
- Issue/Return books
- Track users
- Admin panel

# 19. Functional Analysis – Online Shopping:

- Login/Register
- Browse products
- Add to cart
- Payment gateway
- Order tracking

# 20. System Design – Food Delivery:

- UI: Menu, Order button
- Logic: Location, delivery time
- Data: Restaurant list, order history

# 21. Test Cases – Calculator:

- Add 2+2 = 4
- Subtract 5-3 = 2
- Divide by 0 = Error
- Clear function test

# 22. Maintenance Case Study:

- **Example: Zoom had to update security after "Zoom bombing" attacks during COVID. Critical patching was done fast.**

# 23. DFD for Hospital:

- **Patient → Reception → Doctor/Nurse → Pharmacy/Billing**
  **(Use Level 0 and Level 1 DFD in PPT)**

# 24. Desktop Calculator with GUI:

- Use Python Tkinter:
- from tkinter import *
- root = Tk()
- # Add buttons, display and logic
- root.mainloop()

# 25. Flowchart – Online Registration:

- Start → Enter Details → Validate → Save to DB → Confirm Registration → End