Module-1
Overview of IT Industry (practical)

1. Write a simple "Hello World" program in two different programming languages. Compare structure and syntax.

Python:
```
print("Hello, World!")
```

C:
```
#include <stdio.h>

int main()

{
   printf("Hello, World!");
}
```

Comparison:
- Python is simpler, no extra code like `#include` or `main()`.
- C needs structure, header files, semicolons.

---

2. Diagram: Data transmission from client to server over the internet

Diagram :

Client → DNS → Internet → Server → Response to Client

---

3. Design a simple HTTP client-server communication in any language.

Server (Python):

```
from http.server import BaseHTTPRequestHandler, HTTPServer
```

```
class SimpleHandler(BaseHTTPRequestHandler):
   def do_GET(self):
      self.send_response(200)
      self.end_headers()
      self.wfile.write(b"Hello from Server!")

server = HTTPServer(('localhost', 8080), SimpleHandler)
server.serve_forever()
```

Client:

```
import requests
response = requests.get('http://localhost:8080')
print(response.text)
```

---

4. Research different types of internet connections and list their pros and cons.

Type	Pros	Cons
Broadband	Fast, reliable, always connected	May slow down at peak times
Fiber	Very high speed, stable	Expensive, limited in rural areas
Satellite	Available anywhere	High latency, affected by weather

---

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

HTTP:

```
bash
curl http://example.com
```

FTP:

```
bash
curl ftp://ftp.example.com/file.txt
```

---

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

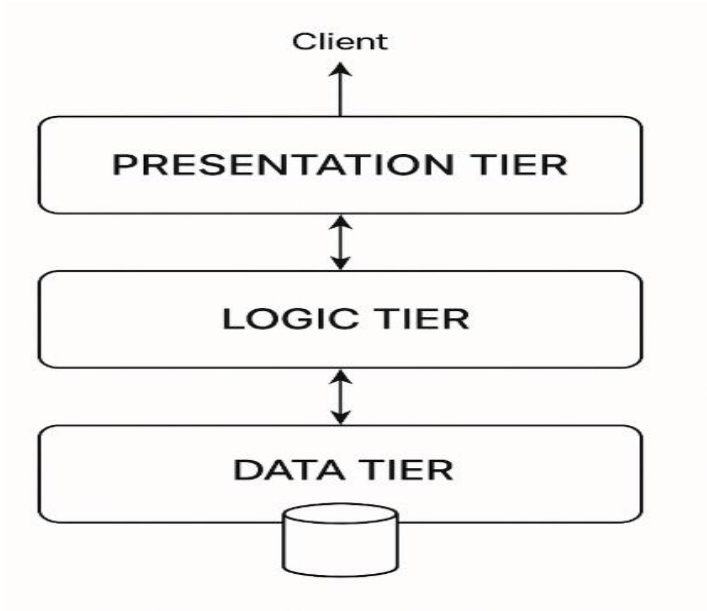
Application	Type
Windows OS	System Software
MS Word	Application
Antivirus	Utility Software
Chrome Browser	Application
File Explorer	System Software

---

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

Three Tiers:

1. Presentation Layer (Frontend)
2. Business Logic Layer (Backend)
3. Data Layer (Database)



8. Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.

Example: E-commerce System

- Presentation Layer: User views products and interacts via UI.

- Business Logic Layer: Processes cart, orders, payments.

- Data Access Layer: Connects to database storing products, users, orders.

---

9. Explore different types of software environments. Set up a basic environment in a virtual machine.

Types:
- Development: for writing and testing code.
- Testing: for checking bugs before release.
- Production: used by real users.

Setup: Install VirtualBox, create Ubuntu VM, and install Python/Node.js.

---

10. Write and upload your first source code file to Github

Steps:
1. Create file (e.g., hello.py)
2. Initialize Git:

```
git init
git add .
git commit -m 'First commit'
git remote add origin <https://github.com/Darshanaprjajapati266/Tops-assignments >
git push -u origin main
```

---

11. Create a Github repository and document how to commit and push code changes.

Steps:
- Create repo on GitHub
- Use terminal:
git init
git add .
git commit -m 'Initial commit'
git branch -M main
git remote add origin < https://github.com/Darshanaprjajapati266/Tops-assignments>
git push -u origin main

---

12. Create a student account on Github and collaborate on a small project with a classmate.

- Register: https://education.github.com
- Create repo → Settings → Collaborators → Add classmate's GitHub username

---

13. Classify software you use into system, application, and utility software.

Software	Type
Windows	System Software
Chrome	Application Software
MS Word	Application Software
WinRAR	Utility Software
Antivirus	Utility Software

---

14. Follow a GIT tutorial to practice cloning, branching, and merging repositories.

git clone <https://github.com/Darshanaprjajapati266/Tops-assignments>
git checkout -b new-feature
Make changes
git commit -am 'feature added'
git checkout main
git merge new-feature

---

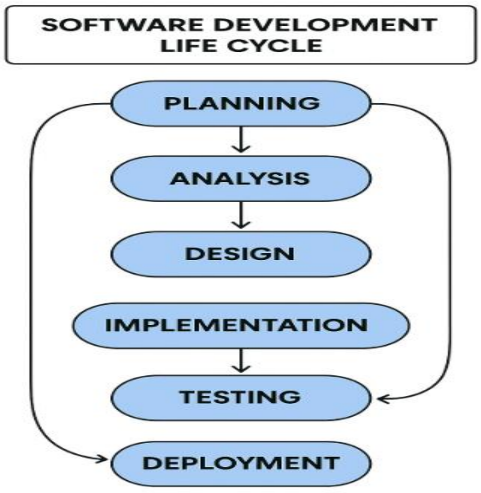
15. Write a report on the various types of application software and how they improve productivity.

Types:
- Word Processor: Write documents
- Spreadsheet: Manage data
- Browsers: Surf the internet
- Email Clients: Communication

They help in tasks, saving time, and organizing work.

---

16. Create a flowchart representing the Software Development Life Cycle (SDLC).



---

17. Write a requirement specification for a simple library management system.

Requirements:
- Add/Delete Books
- Register Users
- Issue/Return Books
- Search by Title/Author
- Track due dates

---

18. Perform a functional analysis for an online shopping system.

Functions:
- Browse products
- Add to cart
- Place order
- Payment
- Admin adds/removes items
- User login/signup

---

19. Design a basic system architecture for a food delivery app.

Layers:
- Frontend: User Interface (App/Web)
- Backend: APIs and Logic
- Database: Orders, Users, Restaurants

20. Develop test cases for a simple calculator program.

Test Case	Input	Expected Output
Addition	5 + 3	8
Subtraction	10 - 4	6
Multiplication	3 * 4	12
Division	8 / 2	4
Division by Zero	5 / 0	Error

---

20. Document a real-world case where a software application required critical maintenance.

Zoom (2020): Faced security issues like 'Zoom-bombing'. The company updated the software quickly with patches to fix the issues.

---

21. Create a DFD for a hospital management system.

DFD:-

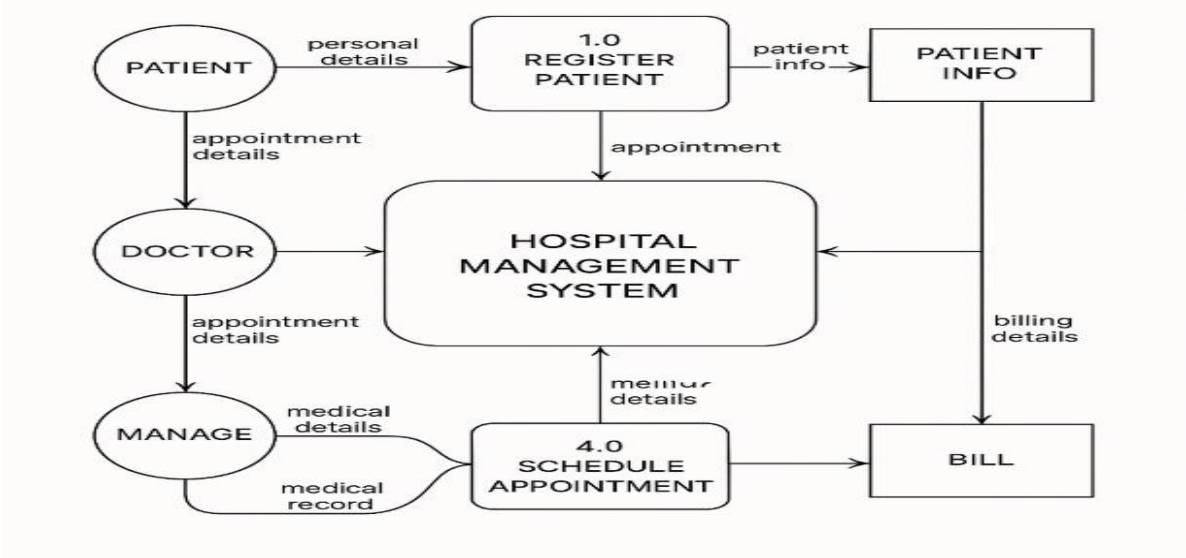
Patient →

Reception →

Doctor →

Lab/Pharmacy →

Billing



---

23. Build a simple desktop calculator application using a GUI library.

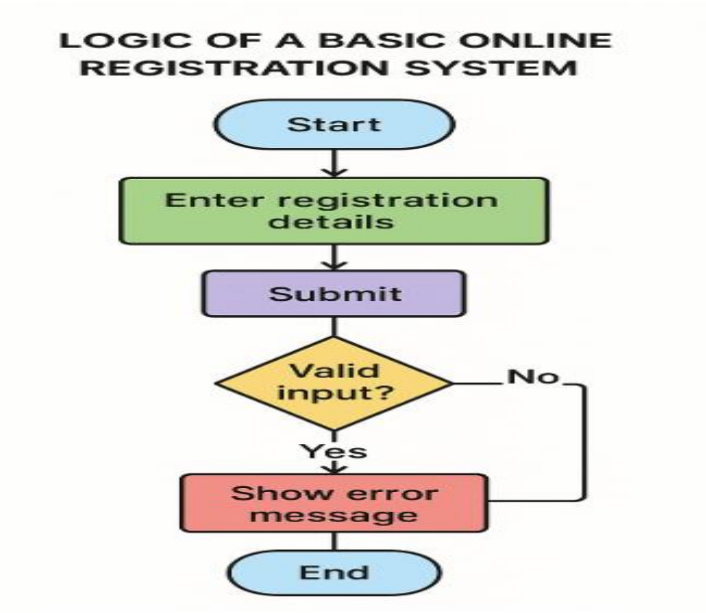
Python with Tkinter:

```
import tkinter as tk

def calculate():
   result.set(eval(entry.get())

root = tk.Tk()
entry = tk.Entry(root)
entry.pack()
result = tk.StringVar()
tk.Button(root, text="=", command=calculate).pack()
tk.Label(root, textvariable=result).pack()
root.mainloop()
```

---

24. Draw a flowchart representing the logic of a basic online registration system.

- Input user details
- Validate info
- Store in DB
- Confirm registration


LOGIC OF A BASIC ONLINE REGISTRATION SYSTEM
Start
Enter registration details
Submit
Valid input?
No
Yes
Show error message
End