**1. GitHub Account & Repository**

* Go to [github.com](https://github.com/) → Sign up → New → Create Repository.

**2. SDLC Phases**

* **Planning → Analysis → Design → Implementation → Testing → Deployment → Maintenance**

**3. "Hello World" in 2 Languages**

**Python:**

print("Hello World")

**C:**

#include <stdio.h>

int main() {

printf("Hello World");

return 0;

}

**Comparison:** Python is simpler; C requires setup and semicolons.

**4. Client to Server Data Flow Diagram**

**Client → DNS → ISP → Internet → Server → Response**  
*(Use draw.io or diagrams.net to visualize)*

**5. HTTP Client-Server (Python Flask)**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def hello():

return "Hello, Client"

app.run()

**6. Internet Connections: Pros & Cons**

| **Type** | **Pros** | **Cons** |
| --- | --- | --- |
| Fiber | Fast, reliable | Costly |
| Broadband | Available, stable | Slower than fiber |
| Satellite | Remote coverage | High latency |

**7. Simulate HTTP/FTP with CLI**

curl http://example.com # HTTP

ftp ftp.example.com # FTP

**8. App Security Vulnerabilities**

* **SQL Injection** → Use parameterized queries
* **XSS** → Sanitize input
* **Broken Auth** → Strong password & sessions

**9. 5 Applications**

| **App** | **Type** |
| --- | --- |
| Chrome | Application |
| Notepad | Application |
| Windows OS | System |
| File Explorer | System |
| CCleaner | Utility |

**10. Three-Tier Architecture**

**Presentation (UI) → Business Logic → Data Access (DB)**

**11. Case Study: Library System**

* **UI**: Search books
* **Logic**: Borrow rules
* **DB Layer**: Book/member data

**12. Software Environments**

* **Dev**: Coding
* **Test**: QA
* **Prod**: Live  
  *Setup: Use VirtualBox + Ubuntu*

**13. Upload Code to GitHub**

git add .

git commit -m "First commit"

git push origin main

**14. Git Commit & Push**

1. git add file
2. git commit -m "msg"
3. git push

**15. GitHub Student Project**

* Apply for [Student Pack](https://education.github.com/pack)
* Add classmate as collaborator

**16. Software Classification**

| **Software** | **Category** |
| --- | --- |
| Windows | System |
| MS Word | Application |
| VLC | Application |
| CCleaner | Utility |
| Antivirus | Utility |

**17. Git Tutorial Commands**

git clone <repo>

git checkout -b new-branch

git merge new-branch

**18. App Software Report**

* Word processors, browsers, spreadsheets
* Boost productivity via automation & accuracy

**19. SDLC Flowchart**

Start → Planning → Analysis → Design → Dev → Test → Deploy → Maintain → End

**20. Library System Specs**

* Users: Admin, Student
* Features: Borrow/return/search
* DB: Books, Users, Logs

**21. Online Shopping Functional Analysis**

* Register/Login
* Browse Products
* Add to Cart
* Checkout & Payment

**22. Food Delivery App Architecture**

* **Client App → Server (API) → Database → Delivery Module**

**23. Calculator Test Cases**

| **Operation** | **Input** | **Output** |
| --- | --- | --- |
| Add | 2 + 2 | 4 |
| Divide | 4 / 0 | Error |

**24. Real-World Maintenance**

**WhatsApp crash bug** → Fixed via app update → Released patch

**25. Hospital System DFD**

Patient → Reception → Doctor → Lab/Pharmacy → DB

**26. Simple Calculator (Python GUI)**

Use **Tkinter** to create GUI with buttons and input fields.

**27. Registration Flowchar**

Start → Fill Form → Validate → Submit → Save → Confirmation → End