

Module 1 – Overview of IT Industry

1.What is a Program?

-computer program is a set of instructions that tells a computer what to do. It takes input, processes it according to those instructions, and produces output. Like a recipe for a computer.

2.What is Programming?

-Programming is creating instructions for a computer. Key steps: define the problem, plan the solution, write the code, test for errors (debug), deploy the program, and maintain it.

3.Types of Programming Languages?

-High-Level Languages:

- **Abstraction:** They provide a high level of abstraction, meaning they hide the complex details of the computer's hardware. You don't need to worry about memory addresses or CPU registers.
- **Readability:** They use syntax that is closer to human language (often English), making them easier to read, write, and understand.
- **Portability:** Programs written in high-level languages can often be run on different types of computers with minimal or no changes. This is because they are translated into machine code by interpreters or compilers.
- **Development Time:** They generally allow for faster development because they require less code to achieve the same result as a low-level language.
- **Examples:** Python, Java, C++, JavaScript, C#

Low-Level Languages:

- **Hardware Control:** They provide direct control over the computer's hardware, allowing for very efficient and optimized code.

- **Complexity:** They are more complex and require a deeper understanding of the computer's architecture.
- **Machine-Specific:** Programs written in low-level languages are often specific to a particular type of computer or processor.
- **Development Time:** They generally require more code and more time to develop compared to high-level languages.
- **Examples:** Assembly Language, Machine Code

4. World Wide Web & How Internet Works?

-data is transmitted from a client to a server over the internet:

1. **Client Request:** The user (client) initiates a request (e.g., visiting a website) via a web browser.
2. **DNS Lookup:** The browser queries a DNS server to resolve the domain name into an IP address of the server hosting the content.
3. **Transmission over Internet:** The request is sent via routers and network protocols (TCP/IP) to the web server.
4. **Server Processing:** The web server processes the request, often retrieving data from a database if needed.
5. **Response:** The server sends the requested data back to the client (e.g., a webpage), which is then displayed in the browser.

Key Components:

- **Client:** User's device (browser).
- **DNS:** Resolves domain names.
- **Router/Network:** Routes the request.
- **Web Server:** Processes and serves content.
- **Database Server (if used):** Stores data.

This process ensures that data successfully travels from the client to the server and back.

5.What is the difference between source code and machine code?

-Source Code and Machine Code are both essential in programming, but they differ significantly:

- Source Code:
 - Written by humans in a high-level programming language (e.g., Python, Java, C++).
 - It's human-readable and contains instructions that are easy to understand and edit.
 - Needs to be compiled or interpreted into machine code for execution.
- Machine Code:
 - The low-level code that a computer's processor can directly execute.
 - Written in binary (1s and 0s), representing instructions specific to a computer's architecture.
 - It's not human-readable and is generated by a compiler or assembler from the source code.

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

-first time use

```
git config --global user.email "your-email@example.com"
```

```
git config --global user.name "Your Name"
```

new repository

1) git init

2) git add .

3) git status

4) git commit -m "first commit"

5) git branch -M main

6) git remote add origin <https://github.com/Shubham-2122/xyz.git>

7) git push -u origin main

new file

1) git add .

2) git status

3) git commit -m "file name"

4) git push -u origin main

7. How does GIT improve collaboration in a software development team?

-Git improves collaboration in a software development team by:

1. **Version Control:** Tracks changes and allows reverting to previous versions, ensuring transparency and traceability.
2. **Branching and Merging:** Developers can work on isolated branches, reducing conflicts, and easily integrate changes later.
3. **Distributed System:** Each team member has a local copy, enabling offline work and syncing changes when online.
4. **Conflict Resolution:** Git helps identify and resolve code conflicts when multiple developers edit the same file.
5. **Collaborative Workflows:** Git supports structured workflows (e.g., GitFlow) to manage feature development, bug fixes, and releases efficiently.

These features enable smooth collaboration, minimize conflicts, and improve team productivity.

8.: What are the main stages of the software development process?

-The main stages of the software development process are:

1. **Planning:** Define project goals, scope, timeline, resources, and requirements. This stage sets the foundation for the development process.
2. **Design:** Create the software architecture and detailed design, including system components, user interface, and database structures.
3. **Development:** Actual coding of the software according to the design specifications. Developers write the source code and implement functionality.
4. **Testing:** Perform various tests (unit, integration, system) to identify bugs and ensure the software meets the required standards and works correctly.
5. **Deployment:** Release the software to users by deploying it in the production environment.
6. **Maintenance:** After deployment, provide ongoing support, fix bugs, and release updates or improvements based on user feedback.