

---

# Module 1 - [Overview of IT Industry]

---

## 1. What is program?

A **program** is a set of instructions written in a programming language that tells a computer how to perform a specific task. It takes input, processes it, and gives output. Programs are used to automate tasks, solve problems, and control computer behavior.

## 2. What is Programming?

**Programming** is the process of writing instructions (called code) that a computer can understand and execute to perform specific tasks. It involves creating programs using programming languages like C, Python, or Java to solve problems or automate work.

## 3. What are the key steps involved in the programming process?

Brief Steps in the Programming Process:

- **Understand the problem** – Know what the program should do.
- **Plan the solution** – Design logic using flowcharts or algorithms.
- **Write the code** – Use a programming language to create the program.
- **Test and debug** – Check for and fix errors in the code.
- **Run the program** – Execute to see if it gives the correct output.
- **Maintain** – Update or improve the program over time.

## 4. Types of Programming Languages?

- **Low-Level Languages**
  - Close to machine; hard to understand (e.g., Machine Language, Assembly).
- **High-Level Languages**
  - Easy to read and write; user-friendly (e.g., C, Java, Python).
- **Procedural Languages**
  - Follows step-by-step instructions (e.g., C, Pascal).
- **Object-Oriented Languages**
  - Based on objects and classes (e.g., Java, C++).

## 5. What are the main differences between high-level and low-level programming languages?

Feature	High-Level Language	Low-Level Language
Ease of Use	Easy to read and write	Hard to understand
Closer to	Human language	Machine hardware
Examples	Python, Java, C++	Assembly, Machine code
Portability	Platform-independent	Platform-dependent
Speed	Slower execution	Faster execution
Use	Application development	System-level programming

## 6. Describe the roles of the client and server in web communication?

- **Client**
  - The client is typically a **web browser or app** used by a user.
  - It **sends requests** to a server for data or services (like opening a website).
  - It **receives and displays** the server's response (e.g., a web page).
- **Server**
  - The server is a **powerful computer** that **stores websites, data, or services**.
  - It **listens for client requests**, processes them, and **sends back responses**.
  - Example: When you type a URL, the server sends the website content to your browser.

## 7. Explain the function of the TCP/IP model and its layers?

### • **Function of TCP/IP model:**

- Sends data from one device to another.
- Breaks data into layers for easy handling.
- Routes data to the correct destination.
- Ensures data is delivered correctly

### • **Layers of TCP/IP Model:**

1. **Application Layer**
  - Supports user services like email, web, file transfer (e.g., HTTP, FTP).
2. **Transport Layer**
  - Ensures reliable data delivery between devices (e.g., TCP, UDP).
3. **Internet Layer**
  - Handles addressing and routing of data (e.g., IP).
4. **Network Access Layer**
  - Deals with physical transmission over network (e.g., Ethernet, Wi-Fi)

## 8. Explain Client Server Communication?

- **Client-Server Communication:**
  - i. Client sends a request.
  - ii. Server receives the request.
  - iii. Server processes the request.
  - iv. Server sends back a response.
  - v. Client uses or displays the response.

## 9. Types of Internet Connections?

- **Dial-up** – Slow connection via telephone line.
- **DSL (Digital Subscriber Line)** – Faster than dial-up, uses telephone lines.
- **Cable** – Uses TV cables, high-speed.
- **Fiber-optic** – Very fast, uses light signals through fiber cables.
- **Satellite** – Internet via satellite signals; useful in remote areas.
- **Wireless (Wi-Fi)** – Connects devices wirelessly via a router.
- **Mobile Data (3G/4G/5G)** – Internet via mobile networks on phones and dongles.
- **Broadband** – General term for high-speed wired internet (DSL, cable, fiber).

## 10. How does broadband differ from fiber-optic internet?

Feature	Broadband	Fiber-Optic
Type	General category	Specific type of broadband
Speed	Varies (slower in DSL/cable)	Very high speed (up to 1 Gbps or more)
Medium	Copper cables or coaxial cables	Glass or plastic fiber-optic cables
Reliability	Moderate	Very reliable and stable
Availability	Widely available	Limited to areas with fiber infrastructure

## 11. What are the differences between HTTP and HTTPS protocols?

Feature	HTTP (Hyper Text Transfer Protocol)	HTTPS (Hyper Text Transfer Protocol Secure)
Full Form	Hyper Text Transfer Protocol	Hyper Text Transfer Protocol Secure

Feature	HTTP (Hyper Text Transfer Protocol)	HTTPS (Hyper Text Transfer Protocol Secure)
Security	Not secure	Secure (uses SSL/TLS encryption)
Data Encryption	No encryption (data is in plain text)	Yes, encrypted using SSL/TLS
Port Number	80	443
URL Prefix	http://	https://
Used For	General websites with no sensitive data	Secure websites (e.g., banking, login pages)
Performance	Slightly faster due to no encryption	Slightly slower due to encryption overhead

## 12. What is the role of encryption in securing applications?

Encryption protects sensitive data by converting it into unreadable code that can only be decoded with the correct key. It ensures:

1. **Confidentiality** – Only authorized users can access the data.
2. **Integrity** – Prevents data from being altered during transmission.
3. **Authentication** – Verifies the identity of users and systems.
4. **Data Protection** – Secures stored and transmitted data from hackers.

## 13. What is the difference between system software and application software?

Feature	System Software	Application Software
Definition	Software that manages hardware and system operations	Software designed to perform specific user tasks
Purpose	To operate and control computer hardware	To help users perform tasks (e.g., writing, browsing)
Examples	Operating System (Windows, Linux), Drivers	MS Word, Chrome, Photoshop
User Interaction	Runs in the background, minimal interaction	Directly interacts with the user
Dependency	Required for the system to function	Runs on top of system software

## 14. What is the significance of modularity in software architecture?

Modularity means breaking software into smaller, independent, and manageable components or modules.

- **Key Benefits:**

1. **Easier Maintenance:**

Each module can be updated or fixed without affecting the whole system.

2. **Reusability:**

Modules can be reused in other projects, saving time and effort.

3. **Improved Debugging and Testing:**

Smaller parts are easier to test and debug individually.

4. **Parallel Development:**

Different teams can work on different modules at the same time.

5. **Scalability:**

Makes it easier to scale the software by adding new modules.

6. **Better Code Organization:**

Code becomes more readable, manageable, and structured.

## 15. Layers in Software Architecture?

1. **Presentation Layer:**

- User interface (UI)
- Handles input/output

2. **Application Layer:**

- Manages communication between UI and business logic

3. **Business Logic Layer:**

- Processes data and applies rules

4. **Data Access Layer:**

- Connects to and queries the database

5. **Database Layer:**

- Stores and manages data

## 16. Why are layers important in software architecture?

**Layers are important in software architecture because they:**

- Organize the system into manageable parts.
- Make the code easier to understand and maintain.
- Allow independent development of each layer.
- Improve scalability and flexibility.
- Enhance security by isolating functions.

## 17. Explain the importance of a development environment in software production.

A **development environment** is the setup (tools, software, and configurations) used by developers to build, test, and debug applications.

- **Efficient Development:**  
Provides necessary tools (IDEs, compilers, debuggers) to write and manage code easily.
- **Error Detection:**  
Helps identify and fix errors early through testing and debugging tools.
- **Version Control:**  
Supports collaboration and tracking of code changes (e.g., Git).
- **Standardization:**  
Ensures consistency across all developers working on the project.
- **Testing and Simulation:**  
Allows testing code in a controlled setting before deploying to real users.
- **Automation:**  
Speeds up tasks like code building, testing, and deployment using automated tools.

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

Aspect	Source Code	Machine Code
<b>Definition</b>	Human-readable instructions written in programming languages (e.g., C, Java)	Binary code understood by the computer (0s and 1s)
<b>Readable By</b>	Humans (programmers)	Machines (CPU)
<b>Format</b>	Text format (e.g., .c, .java files)	Binary format (e.g., .exe, .bin files)
<b>Requires</b>	Needs to be compiled or interpreted	Directly executed by the computer
<b>Editable</b>	Easily editable by developers	Not easily editable

19. Why is version control important in software development?

- **Tracks Changes:**  
Records every change made to the code, making it easy to review or revert.
- **Collaboration:**  
Allows multiple developers to work on the same project without overwriting each other's work.
- **Backup and Recovery:**  
Keeps a history of code, so old versions can be restored if something breaks.
- **Branching and Merging:**  
Enables testing new features or fixes without affecting the main codebase.
- **Accountability:**  
Tracks who made which change and why (with commit messages).

## 20. What are the benefits of using Github for students?

- **Free Code Hosting:**  
Store and manage your projects online.
- **Learning Git:**  
Gain real-world experience with version control (Git), a key skill for developers.
- **Portfolio Building:**  
Showcase your work to potential employers.
- **Collaboration:**  
Work on group projects easily with friends or classmates.
- **Access to GitHub Student Pack:**  
Free tools and resources (like free domains, cloud credits, etc.).
- **Open-Source Contribution:**  
Learn by contributing to real-world projects.

## 21. What are the differences between open-source and proprietary software?

Aspect	Open-Source Software	Proprietary Software
Source Code	Publicly available; can be viewed, modified, and shared.	Not available; only the developer/company can modify it.
Cost	Usually free or low-cost.	Usually paid; license fees apply.
Customization	Highly customizable by users.	Limited customization; controlled by the vendor.
Support	Community-driven support (forums, online docs).	Official support from the vendor.
Updates	Anyone can contribute improvements; frequent updates possible.	Updates released only by the vendor at their discretion.

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

### Git helps collaboration by:

- ✓ Allowing multiple people to work on the same project.
- ✓ Keeping track of all code changes.
- ✓ Letting developers work on separate branches.
- ✓ Merging updates without losing work.
- ✓ Reverting to older versions if needed.

- ✓ Working with platforms like GitHub for reviews and feedback.

## 23. What is the role of application software in businesses?

### **Role of application software in business:**

- Automates work.
- Saves time and increases productivity.
- Helps in decision-making.
- Improves communication.
- Manages resources.
- Improves customer service.

## 24. What are the main stages of the software development process?

### **Main stages of the software development process:**

- Planning** – Define goals, requirements, and resources.
- Analysis** – Study and document detailed requirements.
- Design** – Create system architecture and interface designs.
- Development** – Write and build the actual code.
- Testing** – Check for bugs and ensure quality.
- Deployment** – Release the software to users.
- Maintenance** – Fix issues and update the software.

## 25. Why is the requirement analysis phase critical in software development?

### **Requirement analysis** is critical because it:

- Defines exactly what the software should do.
- Prevents misunderstandings between developers and clients.
- Helps plan time, cost, and resources accurately.
- Reduces the risk of errors and rework later.
- Ensures the final product meets user needs.

## 26. What is the role of software analysis in the development process?

### **Role of software analysis in development:**

- Understands user needs and expectations.
- Defines clear and detailed requirements.
- Identifies possible problems early.
- Guides design and development decisions.
- Ensures the final product matches user needs.

## 27. What are the key elements of system design?



### Key elements of system design:

1. **Architecture design** – Structure of the system and its components.
2. **User interface design** – Layout and interaction for users.
3. **Database design** – Organization of data storage and retrieval.
4. **Data flow design** – How data moves within the system.
5. **Security design** – Measures to protect data and operations.
6. **Integration design** – How different modules or systems work together.

### 28. Why is software testing important?

#### Software testing is important because it:

- Finds and fixes errors before release.
- Ensures the software works as expected.
- Improves quality and reliability.
- Increases user satisfaction.
- Saves time and cost by preventing future issues.

### 29. What types of software maintenance are there?

#### Types of software maintenance:

1. **Corrective** – Fixing bugs and errors.
2. **Adaptive** – Updating software to work in new environments.
3. **Perfective** – Adding new features or improving performance.
4. **Preventive** – Making changes to avoid future problems.

### 30. What are the key differences between web and desktop applications?

Aspect	Web Application	Desktop Application
Installation	Runs in browser, no install needed	Must be installed on computer
Internet	Needs internet to work	Works without internet (mostly)
Updates	Updates automatically	Updates manually
Data Storage	Stored online (cloud)	Stored on device
Device Support	Works on any device/OS	Made for specific OS

### 31. What are the advantages of using web applications over desktop applications?

#### Advantages of web applications over desktop applications:

1. No installation needed – runs directly in a browser.
2. Accessible from anywhere with the internet.
3. Works on any device and operating system.
4. Updates automatically without user action.
5. Data is stored online, so it's easy to share and back up.
6. Easier collaboration between users.

32. What role does UI/UX design play in application development?

**Role of UI/UX design in application development:**

- **Enhances usability** – Makes the app easy to use and navigate.
- **Improves user satisfaction** – Creates a pleasant and engaging experience.
- **Boosts accessibility** – Ensures the app works for all users, including those with disabilities.
- **Increases efficiency** – Helps users complete tasks quickly.
- **Supports brand identity** – Maintains a consistent look and feel.
- **Encourages user retention** – Keeps users coming back to the app.

33. What are the differences between native and hybrid mobile apps?

Aspect	Native Apps	Hybrid Apps
Platform	Made for one platform only	Works on multiple platforms
Speed	Faster and smoother	Slower than native
Features	Full device feature access	Limited feature access
Cost	More expensive	Cheaper
Updates	Harder to update	Easier to update

34. What is the significance of DFDs in system analysis?

**Significance of DFDs (Data Flow Diagrams) in system analysis:**

1. Shows how data moves through the system.
2. Helps understand processes and data storage.
3. Makes complex systems easier to visualize.
4. Identifies inputs, outputs, and data processing steps.
5. Improves communication between developers and clients.

35. What are the pros and cons of desktop applications compared to web applications?

**Pros of Desktop Applications:**

- Usually faster and use local resources.
- Can work without an internet connection.
- Handle heavy tasks efficiently.
- Full access to device features.

#### **Cons of Desktop Applications:**

- Limited to the device where installed.
- Harder to sync data across devices.
- Require manual updates.
- OS-specific, not cross-platform.

#### **Pros of Web Applications:**

- Accessible from anywhere with internet.
- No installation required.
- Auto-updates instantly.
- Works on any device or operating system.

#### **Cons of Web Applications:**

- Slower if internet connection is poor.
- Usually need internet to work.
- May not handle heavy tasks well.
- Limited access to device features.

### 36. How do flowcharts help in programming and system design?

#### **Flowcharts help in programming and system design by:**

1. **Visualizing the process** – Show each step in a clear, graphical way.
2. **Simplifying complex logic** – Break down complicated processes into easy-to-understand parts.
3. **Improving communication** – Help developers, designers, and clients understand the system easily.
4. **Identifying problems early** – Make it easier to spot errors or inefficiencies before coding starts.
5. **Guiding development** – Serve as a blueprint for writing the program or building the system.