* **What is A Program ?**
* A Program is a set of instructions that a computer follows to perform a specific

task.

- Programs are written in Programming Languages, and are stored as codes.

- While executing, the program directs the hardware to carry out instructions.

- Every Proper Program has an output or a result.

* **Key Steps of Programming :**
* Programming Involves defining the Problem, Designing a Solution, Coding it, Testing for Errors, and Maintaining the Software after Deployment.
* **Types of Programming Languages :**
* Programming languages include : -

1. Procedural Language like C
2. Object-Oriented Language like Java
3. Functional Language like Haskell
4. Scripting Language like JavaScript
5. Low-Level Languages like Assembly

* **Difference between High-Level and Low-Level Programming Languages :**

|  |  |
| --- | --- |
| High-Level | Low-Level |
| * Easier to Learn and Use. | * More Complex and Harder to Learn. |
| * Highly Portable. | * Less Portable. |
| * Examples :   Python, C++, Java, JavaScript | * Examples :   Assembly Language and Machine Code |

* **World Wide Web & how internet Works :**
* The World Wide Web(WWW) is a system of interlinked Web Pages and Resources accessed via the Internet.
* The Internet is a global network where data travels through routers using protocols like TCP/IP, Connecting Devices instantly across the world.
* **TCP/IP :**

- The Full Form of TCP/IP is, (Transmission Control Protocol/Internet Protocol).

- It is a conceptual framework that defines how data is transmitted across

computer networks, most notably the internet.

- Its primary function is to provide a standardized set of rules & procedures

that allow diverse devices & networks to communicate seamlessly.

* **Client-Server Communication :**
* Client Sends a request > Server Processes it > Server Replies to Client
* It Uses HTTP For Data Exchange.
* It Enables Web & API Interactions.
* **Types of Internet Connections :**

**-> Types of connections availible :**

* DSL,Cable, Fiber, Satellite, and Mobile Broadband, etc,
* 1) Cable :-
* The Cable Internet offers High-Speed, Wide Availibility, etc.
* Some of their cons are Asymmetrical Speeds, Shared Bandwidth, etc.
* 2) DSL :-
* The DSL is Affordable and offers dedicated connection, wide availibility,etc.
* Cons are Slower & Asymmetrical Speeds, and it's dependant on Phone Line
* Quality.
* 3) Mobile Broadband :-
* It offers Quick Setup, Portability, etc.
* It's cons are Higher cost, Variable Performance.
* **Broadband & Fiber-Optic Internet Difference :**
* Broadband is a general term for high-speed internet. Fiber-Optic internet is a specific, high-performance type of broadband. It uses light signals for much faster, more reliable connections than other broadband technologies that rely on copper wires.
* Therefore, fiber is a type of broadband, but not all broadband is fiber.
* **Role of Encryption :**
* Encryption plays a fundamental & critical role in securing applications by transforming readable data into an unreadable format.
* It ensures that even if unauthorized individuals gain access to data, it remains incomprehensible & useless without correct decryption key.
* It’s key benefits are Data Confidentiality, Data Integrity, Regulatory Compliance, etc.
* **Difference between System & Application Software :**
* System Software manages a computer’s hardware and provides a platform for other programs to run. It is the essential foundation, including OS like Windows and MacOS.
* Application Software is designed for specific user tasks such as web Browse or Word Processing, and it relies on system software to function.
* **Significance of modularity in Software Architecture :**
* Modularity in Software Architecture is the practice of breaking a system into smaller, independent components called modules. This simplifies maintainance, allows for concurrent development, and makes code reusable and easier to understand. It’s significant because it helps manage complexity, improves efficiency, and makes software more flexible and scalable.
* **Why are layers important in Software Architecture ?**
* Layers organize software into distinct parts with specific jobs, simplifying development, maintainance, and scalability by isolating concerns. This structured approach ensures a more manageable system.
* **Development Environment in Software Production :**
* A development environment in software production is a workspace equipped with a specific set of tools, configurations and processes that developers use to write, test, debug, and refine code.
* It’s essentially a controlled ‘sandbox’ where software can be built and iterated upon without affecting live systems or end users.
* **Difference between Source code & Machine Code :**
* **Source Code :**
* Written in Languages like Python, C++, etc. It uses a syntax that’s easy for people to understand.
* It can’t be run directly & must be converted into Machine Code by a Compiler or Interpreter.
* **Machine Code :**
* Composed of Binary, it’s the only Language a Computer’s Processor can Execute directly.
* It’s unique to a specific Processor Arcihtecture, so code for one type of CPU won’t work on another.
* **Benefits of using GitHub for Students :**
* GitHub allows Students to Showcase their Coding Skills & Projects to Potential Employers. A well-maintained GitHub Profile with a variety of Projects Demonstrates a student’s real world experience, collaboration skills, & commitment to their craft.
* Beyond building a Portfolio, another major benefit of using GitHub is learning to collaborate and contribute to open-source projects.
* **Difference between Open-Source & Proprietary Software :**

1. **Source Code:**

* The source code is publicly available, allowing anyone to view, modify, and distribute it. This fosters a collaborative environment where a community of developers can work together to improve the software.

1. **Proprietary Software :**

* The source code is closed & kept confidential. It is the intellectual property of the company or the individual who created it, and only they can make changes to it.
* **Role of Application Software in Business :**
* Application Software streamlines business operations by automating tasks to enhance productivity and provides valuable insights from centralized data to support informed decision-making. This helps companies gain a competitive Advantage.
* **Flowchart representing the software development life cycle :**

Requirements Analysis

Coding

Deployment

Maintainance

Deployment

Testing

Design

Planning

* **Main stages of Software Development Process :**
* The Software Development Process typically involves several key stages : **Plannning, Requirements Analysis,** and **Design** to establish project’s foundation.
* This is followed by **Implementation (Coding), Testing** and **Deployment,** where the Software is built, validated and released. Finally, the **Maintenance** stage ensures the software remains functional and updated over time.
* **Requirement Specification for Simple Library Management System :**

1. ***Functional Requirements :-***

* **Member Management :** Add, Edit, Delete and Search for members.
* **Book Management :** Add, Edit or Remove Books; Track their State.
* **Transaction Management :** Issue books to members, record returns, and calculate overdue fines.
* **Search & Reporting :** Search for books by title/author and generate a report of books on loan

1. ***Non-Functional Requirements :-***

* **Performance :** System Actions must be fast.
* **Usability :** The Interface should be easy and intuitive for librarians.
* **Security :** A password is required for login.
* **Reliability :** The system needs a data backup mechanism.
* **Role of Software analysis in Development Process :**
* Software analysis is a critical upfront phase in development where the team gathers documents, and refines project requirements. Its Primary role is to create a detailed **Software Requirements Specification (SRS),** which acts as a blueprint for the Project.
* By doing this, it helps to migrate risks, prevent misunderstandings, and ensure the final software meets user needs, ultimately saving time & resources during coding & testing phases.
* **What are the Key Elements of System Design :**

1. **Scalability :** Ensures the system can handle increasing load, using techniques like load balancing and caching.
2. **Reliability :** The System must be available and functional, using mechanisms like redundancy & failover.
3. **Data Management :** Involves selecting the right database (SQL vs. NoSQL) & managing data distribution & replication.
4. **Security :** This includes protecting the system with authentication, authorization, an ddata encryption.
5. **Communication :** Defines how different components talk to each other, often using APIs and message queues.

* **Test cases for a simple calculator program :**

1. **Addition :**

* Input 5+3, Expected Output : 8
* Input 10+(-4), Expected Output : 6

1. **Subtraction :**

* Input 8-2, Expected Output : 6
* Input 2-8, Expected Output : -6

1. **Multiplication :**

* Input 6\*7, Expected Output : 42
* Input 5\*0, Expected Output : 0

1. **Division :**

* Input 10/2, Expected Output : 5
* Input 15/4, Expected Output : 3.75
* **Why Software Testing is Important ?**
* Software testing is important because it ensures the **quality and reliability** of a product by finding and fixing bugs early in development. It reduces costs by preventing expensive fixes after launch and enhances **security** by identifying vulnerabilities. Ultimately, a well-tested product leads to higher **customer satisfaction** and builds a good reputation for the software and its creators.
* **What Types of Software Maintenance are there ?**
* There are 4 main types of Software Maintenance :-

1. Corrective Maintenance
2. Adaptive Maintenance
3. Perfective Maintenance
4. Preventive Maintenance

* **Key Differences between web and desktop applications :**

1. **Web Applications**

* **Accessibility & Updates :** Accessible from any device with an internet connection & a web browser. Updates are automatic & managed centrally, so all users have the latest version instantly.
* **Platform Independence :** They are not tied to a specific operating system. The same web app can be used on Windows, macOS, or Linux, as long as a compatible browser is available.

1. **Desktop Applications**

* **Performance & Offline Use :** They run locally on a computer's hardware, offering faster performance and the ability to work without an internet connection.
* **Installation & Updates :** Require a one-time download and installation on each device. Updates must be downloaded and installed by the user, which can lead to version fragmentation.
* **What are the Advantages of using web applications over the desktop applications ?**

1. **Universal Accessibility:** Web apps can be used on any device with an internet browser, eliminating the need for installation. This makes them easily accessible to a broader user base regardless of their operating system.
2. **Simplified Maintenance:** Updates are deployed centrally on the server, so all users get the latest version automatically. This reduces the burden of manual updates on the user and simplifies maintenance for developers.

* **Role of UI/UX Design in Application Development :**
* UI/UX design is crucial for application development as it ensures the software is **user-friendly, intuitive, and effective**. UX design focuses on the overall user journey and logical flow, while UI design concentrates on visual elements like colors and layout.
* Together, they are essential for **high user satisfaction, improved retention, and achieving business goals**, as a good experience directly impacts an app's success.
* **Difference between Native & Hybrid Mobile Apps :**

1. **Native Mobile Apps**

* **Performance:** Faster performance and full access to device hardware (e.g., camera, GPS).
* **Cost:** Require a separate codebase for each platform (iOS and Android), making development more expensive and time-consuming.

1. **Hybrid Mobile Apps**

* **Performance:** Can be slightly slower and have limited access to some device-specific features compared to native apps.
* **Cost:** Use a single codebase built with web technologies, making them cheaper and faster to develop.
* **What is the significance of DFDs in System Analysis ?**
* DFDs are a key tool in system analysis because they provide a visual, easy-to-understand map of how data flows through a system.
* They are crucial for improving communication among stakeholders, identifying bottlenecks or inefficiencies, and ensuring a clear, shared understanding of the system’s requirements before the development process begins.

* **What are the pros & cons of desktop applications compared to web applications ?**
* **Pros :**
* Offer superior Performance & can function offline as they run directly on the OS, leveraging a device’s full hardware potential.
* Whereas Web applications require a stable internet connection to function & are limited by browser’s capabilities & network speed, which can impact performance.
* **Cons :**
* Require a separate development for each platform & must be manually installed & updated on each device, limiting accessibility.
* While Web Applications are highly accessible from any device with a browser & internet connection, with updates automatically handled on the server side.
* **How do Flowcharts help in programming & system design ?**
* Flowcharts are essential tools in programming and system design because they provide a visual, step-by-step representation of a process or algorithm.
* They help designers to :
* Plan & Logically Organize Code
* Debug & Troubleshoot
* Communicate & Document
* Design & Analyze Systems