**Module:- 1**

1. **What is a HTTP?**

* HTTP (HyperText Transfer Protocol) is the communication protocol used by web browsers and servers to send and receive data over the internet.
* Example: When you open a website, your browser sends an HTTP request to the server, and the server sends back the webpage**.**

1. **What is a Browsers? How they works?**

* A browser (like Chrome, Firefox) is software that lets you view websites.
* How it works:
* You enter a URL.
* Browser sends an HTTP request to the server.
* Server sends HTML/CSS/JS files.
* Browser renders and shows the webpage.

1. **What is Domain Name?**

* A domain name is the human-friendly name of a website (like google.com).
* It points to a server's IP address where your site is hosted.

1. **What is hosting?**

* **Hosting** is a service that stores your website's files on a server so they can be accessed online.
* Without hosting, your site can't be live on the internet.

**Module:- 2**

1. **Difference between Web Designer and Web Developer.**

* Web Designer: Focuses on look & feel of the website (layout, colors, UI/UX).
* Web Developer: Builds the functionality (coding, backend, frontend).

1. **What is a W3C?**

* W3C (World Wide Web Consortium) is the organization that creates web standards to ensure websites work properly across all browsers and devices.

1. **What is Domain?**

* A Domain is the name of a website (like example.com) that maps to the server where the site is hosted.

1. **What is SEO?**

* SEO (Search Engine Optimization) is the process of improving a website to rank higher on search engines like Google.

1. **What is SDLC Life Cycle?**

* SDLC (Software Development Life Cycle) is the step-by-step process to build software:
* Requirement Gathering
* Design
* Coding
* Testing
* Deployment
* Maintenance

**Module:- 3**

1. **Explain in your own words what a program is and how it functions.**

* A program is a set of instructions written for a computer to follow. It tells the computer what to do, step by step, to complete a task. The computer reads and executes these instructions to perform functions like calculations, displaying information, or running apps.

1. **What are the key steps involved in the programming process?**

* The key steps in the programming process are:
* **Planning –** Understand the problem and define the goal.
* **Designing –** Create a blueprint or flowchart of the solution.
* **Coding –** Write the actual program using a programming language.
* **Testing –** Run the program to find and fix errors.
* **Debugging** – Correct any bugs or issues found.
* **Maintaining** – Update and improve the program over time.

1. **What are the main differences between high-level and low-level programminglanguages?**

* **High-level languages** are easier to read, write, and understand (e.g., Python, Java). They are closer to human language and are portable across systems.
* **Low-level languages** are closer to machine code (e.g., Assembly, Machine Language). They give more control over hardware but are harder to write and understand.

1. **Describe the roles of the client and server in web communication.**

* The **client** sends requests (like opening a website), and the **server** receives those requests, processes them, and sends back the needed data (like the webpage).

1. **Explain the function of the TCP/IP model and its layers.**

* The TCP/IP model is a framework for how data is sent over the internet. It has four layers:
* Application – Interfaces with user apps (e.g., web, email).
* Transport – Ensures reliable data delivery (TCP).
* Internet – Handles addressing and routing (IP).
* Network Access – Manages physical data transmission.

1. **How does broadband differ from fiber-optic internet?**

* **Broadband** is a general term for high-speed internet (like DSL or cable).
* **Fiber-optic** internet is a type of broadband that uses light through glass fibers, offering much faster speeds and better reliability than other broadband types.

1. **What are the differences between HTTP and HTTPS protocols?**

* **HTTP** is a protocol for transferring data on the web, but it's not secure.
* **HTTPS** is the secure version of HTTP—it encrypts the data to protect it from hackers.

1. **What is the role of encryption in securing application, Software Applications and Its Types?**

* Encryption protects data by converting it into unreadable code, ensuring only authorized users can access it—this secures apps from hackers.
* Software applications are programs designed for specific tasks. Types include:
* Productivity (e.g., Word, Excel)
* Media (e.g., Photoshop, VLC)
* Web (e.g., browsers)
* Mobile (e.g., WhatsApp, Instagram)
* Enterprise (e.g., ERP, CRM)

1. **What is the difference between system software and application software?**

* **System software** runs the computer and manages hardware (e.g., Windows, macOS).  
  **Application software** helps users do specific tasks (e.g., Word, Chrome).

1. **What is the significance of modularity in software architecture?**

* **Modularity** means breaking software into separate, independent parts (modules). It makes code easier to build, understand, test, and maintain.

1. **Why are layers important in software architecture?**

* **Layers** in software architecture separate different functions, making the system more organized, scalable, and easier to maintain. Each layer handles a specific task, promoting reusability and reducing complexity.

1. **Explain the importance of a development environment in software production?**

* A **development environment** provides tools and resources (like code editors, compilers, and debuggers) to help developers write, test, and debug software efficiently. It ensures consistency, productivity, and a streamlined development process.

1. **What is the difference between source code and machine code?**

* **Source code** is the human-readable code written by developers in programming languages (e.g., Python, Java).  
  **Machine code** is the low-level, binary code that the computer's CPU understands and executes.

1. **Why is version control important in software development?**

* **Version control** tracks changes in code over time, allowing developers to collaborate, manage different versions, and easily revert to previous states, ensuring better organization and reducing errors.

1. **What are the benefits of using Github for students?**

* **Collaboration** – Easily work with others on projects.
* **Version control** – Track and manage code changes.
* **Portfolio** – Showcase projects for future employers.
* **Learning** – Access open-source projects and contribute.

1. **What are the differences between open-source and proprietary software?**

* **Open-source**: Code is **public**, can be **used, modified, and shared** freely (e.g., Linux).
* **Proprietary**: Code is **closed**, owned by a **company**, and **can't be modified** (e.g., Windows).

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

* Tracks **changes** in code.
* Enables **branching** for separate features.
* Supports **merging** work from multiple developers.
* Keeps a **history** of all changes.
* Helps avoid **conflicts** and makes teamwork smooth.

1. **What is the role of application software in businesses?**

* **Automates tasks** (e.g., accounting, HR).
* **Improves productivity** and efficiency.
* **Manages data** and operations.
* **Enhances communication** (e.g., email, collaboration tools).
* Supports **decision-making** with reports and analytics.

1. **What are the main stages of the software development process?**

* **Planning** – Define goals and requirements.
* **Analysis** – Understand user needs.
* **Design** – Plan system architecture and UI.
* **Development** – Write the actual code.
* **Testing** – Find and fix bugs.
* **Deployment** – Release to users.
* **Maintenance** – Update and improve over time.

1. **Why is the requirement analysis phase critical in software development?**

* Ensures **clear understanding** of what users need.
* Helps avoid **costly changes** later.
* Guides **design and development**.
* Reduces **errors and confusion**.
* Sets the **foundation** for project success.

1. **What is the role of software analysis in the development process?**

* **Identifies user needs** and system requirements.
* **Defines functional** and **non-functional requirements**.
* **Clarifies scope** of the project.
* Helps design a **feasible solution**.
* Reduces risks by detecting potential issues early.

1. **What are the key elements of system design?**

* **Architecture Design –** Structure of the system (e.g., client-server).
* **Database Design** – Organizing data and relationships.
* **Interface Design** – User interaction and UI layout.
* **Component Design** – Detailing system modules and their functions.
* **Security Design** – Ensuring data protection and access control.
* **Performance Design** – Optimizing efficiency and scalability.

1. **Why is software testing important?**

* **Ensures quality** by identifying bugs.
* **Improves user experience** by ensuring reliability.
* **Reduces costs** by catching issues early.
* **Confirms requirements** are met.
* **Enhances security** by detecting vulnerabilities.

1. **What types of software maintenance are there?**

* **Corrective** – Fixing bugs or issues.
* **Adaptive** – Updating software to work with new environments or technologies.
* **Perfective** – Enhancing features or performance.
* **Preventive** – Making changes to prevent future problems.

1. **What are the key differences between web and desktop applications?**

* **Web Applications: Run in a browser, require internet access, and are platform-independent.**
* **Desktop Applications: Run on a specific operating system, do not require internet for basic functions, and are often more resource-intensive.**

1. **What are the advantages of using web applications over desktop applications?**

* **Cross-platform** compatibility (works on any device with a browser).
* **No installation** required, accessed via the internet.
* **Easier updates** and maintenance (centralized).
* **Remote accessibility** from anywhere.
* **Lower hardware requirements** for users.

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

* **UI (User Interface)**: Ensures a **visually appealing** and **easy-to-navigate** layout.
* **UX (User Experience)**: Focuses on providing a **smooth, intuitive** experience that meets user needs.
* Together, they **enhance usability**, **increase user satisfaction**, and **boost engagement**.

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

* **Native Apps**: Built for a specific platform (iOS or Android), offer **better performance** and **access to device features**.
* **Hybrid Apps**: Developed using web technologies (HTML, CSS, JavaScript), run on multiple platforms, but may have **slower performance** and limited **device integration**.

1. **What is the significance of DFDs in system analysis?**

* **Visualize data flow** through a system.
* **Clarify system processes** and data interactions.
* Help identify **data sources, processes**, and **endpoints**.
* **Simplify communication** between developers, analysts, and stakeholders.
* Assist in **requirements gathering** and **problem-solving**.

1. **What are the pros and cons of desktop applications compared to webapplications?**

* **Desktop Applications:**
* **Pros:**
* Faster performance (no reliance on internet).
* Access to system resources (hardware, files).
* Works offline.
* **Cons:**
* Platform-dependent (needs specific OS).
* Requires installation and regular updates.
* Limited access from other devices.
* **Web Applications:**
* **Pros:**
* Cross-platform (works on any device with a browser).
* No installation required.
* Easier updates and maintenance.
* **Cons:**
* Requires internet connection.
* Slower performance (due to browser and server dependency).
* Limited access to local resources.

1. **How do flowcharts help in programming and system design?**

* **Visualize processes** and decision logic clearly.
* Help **plan and structure** algorithms.
* **Simplify debugging** by identifying problem areas.
* Enhance **communication** among team members and stakeholders.
* Serve as **documentation** for system design.