[ Frontend Theory Assignment ]

Module 1 – Foundation

**THEORY EXERCISE:**

**1. What is HTTP?**  
HTTP (HyperText Transfer Protocol) is a protocol used for transferring data over the web. It defines how web browsers and servers communicate.

**2. What is a Browser? How do they work?**  
A browser is software (like Chrome or Firefox) that lets you access and view websites.  
**How it works:**

* You enter a URL.
* The browser sends an HTTP request to the web server.
* The server responds with the website's files (HTML, CSS, JS).
* The browser displays the website on your screen.

**3. What is a Domain Name?**  
A domain name is the human-readable address of a website (e.g., example.com). It points to the IP address of the web server hosting the site.

**4. What is Hosting?**  
Hosting is a service that stores website files and makes them accessible on the internet.

Here are short and clear answers to your questions:

**Module 2 – Fundamentals of World Wide Web**

**THEORY EXERCISE:**

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

* **Web Designer:** Focuses on the look and feel (layout, colors, UX/UI).
* **Web Developer:** Builds the actual website using code (HTML, CSS, JavaScript, etc.).

**2. What is W3C?**  
W3C (World Wide Web Consortium) is the main international organization that develops web standards to ensure websites work properly across all browsers and devices.

**3. What is Domain?**  
A domain is the web address of a website (like google.com) that points to a server’s IP address.

**4. What is SEO?**  
SEO (Search Engine Optimization) is the process of improving a website to rank higher in search engine results and attract more visitors.

**5. What is SDLC Life Cycle?**  
SDLC (Software Development Life Cycle) is a process for planning, creating, testing, and deploying software. Common phases:  
**Requirement → Design → Development → Testing → Deployment → Maintenance**

**Module 3 – Fundamentals of IT**

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

A program is a set of instructions written in a programming language that a computer follows to perform specific tasks. It functions by being executed by the computer's processor, which reads the instructions one by one and carries them out—like doing calculations, displaying text, or interacting with hardware.

**What is Programming?**

**2. What are the key steps involved in the programming process?  
Problem Analysis** – Understand the problem and requirements.

1. **Design** – Plan the solution using algorithms or flowcharts.
2. **Coding** – Write the program using a programming language.
3. **Testing** – Run the program to find and fix errors.
4. **Debugging** – Correct the issues found during testing.
5. **Documentation** – Write explanations for how the code works.
6. **Maintenance** – Update and improve the program over time.

**Types of Programming Languages**

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

High-level languages are user-friendly, closer to human language (e.g., Python, Java), and easier to write, read, and debug.  
Low-level languages are closer to machine code (e.g., Assembly, Machine Language), faster and more efficient, but harder to understand and write.

**[World Wide Web]**  
**3. Describe the roles of the client and server in web communication**

The client sends requests to access data or services (like a web browser).  
The server receives the requests, processes them, and sends back the appropriate response (like a web page or data).

**Network Layers on Client and Server**

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

The TCP/IP model is a framework for how data is transmitted over networks. It ensures reliable communication between devices.

Its four layers are:

1. Application – Interfaces with user applications (e.g., HTTP, FTP).
2. Transport – Ensures reliable data transfer (e.g., TCP, UDP).
3. Internet – Routes data across networks (e.g., IP).
4. Network Access – Handles physical data transmission (e.g., Ethernet, Wi-Fi).

**Client and Server**

1. **Explain Client Server Communication**

**Client-server communication** is a model where the **client** (e.g., a web browser) sends a **request** to a **server**, and the server processes it and sends back a **response** (e.g., a webpage). This exchange happens over a network, usually using protocols like **HTTP**.

**Types of Internet Connections**

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

**Broadband is a general term for high-speed internet (includes DSL, cable, satellite).  
Fiber-optic internet is a type of broadband that uses light through fiber cables, offering faster speeds, higher reliability, and lower latency than other broadband types.**

**Protocols**

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

**HTTP** is the basic protocol for transferring web data but is **not secure**.  
**HTTPS** is the secure version of HTTP; it uses **encryption (SSL/TLS)** to protect data, making it safer for sensitive information like passwords and payments.

**Application Security**

**8. 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. It secures applications by preventing data theft during storage or transmission.

**Software application types:**

1. **System Software** – Runs the computer (e.g., OS).
2. **Application Software** – Performs user tasks (e.g., MS Word, browsers).
3. **Utility Software** – Maintains system health (e.g., antivirus, backup tools).
4. **Programming Software** – Helps write code (e.g., compilers, IDEs).

**Software Architecture**   
9. **What is the significance of modularity in software architecture?**

Modularity in software architecture is significant because it breaks a system into separate, independent components, making the software easier to develop, test, maintain, and scale. It improves code reuse, enhances readability, and allows teams to work on different modules simultaneously.

**Layers in Software Architecture**

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

Layers in software architecture are important because they separate concerns, organize code by functionality, and improve maintainability, scalability, and testability. Each layer handles a specific role, making the system easier to manage and modify.

**Software Environments**

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

A development environment is crucial in software production because it provides the tools, configurations, and infrastructure needed for coding, testing, and debugging. It ensures consistency, reduces errors, and speeds up development by simulating the production environment**.**

**Source Code**   
**12. What is the difference between source code and machine code?**

Source code is human-readable code written by programmers in high-level languages (like Python or Java), while machine code is the binary, low-level code that a computer's processor can directly execute.

**GitHub**   
**13. Why is version control important in software development?**

Version control is important in software development because it helps track changes, collaborate with others, manage code history, and easily revert to previous versions if something breaks.

**Student Account in Github**

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

GitHub benefits students by offering:

* Free access to version control and collaboration tools
* Portfolio building for job opportunities
* GitHub Student Developer Pack with free tools and resources
* Easy collaboration on group projects
* Learning real-world development workflows

**Types of Software**

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

* **Access to Code**:
  + *Open-source*: Source code is available to the public.
  + *Proprietary*: Source code is closed and owned by the company.
* **Cost**:
  + *Open-source*: Usually free.
  + *Proprietary*: Often requires payment or licensing fees.
* **Customization**:
  + *Open-source*: Can be modified and redistributed.
  + *Proprietary*: Cannot be legally modified or shared.
* **Support**:
  + *Open-source*: Community-based support.
  + *Proprietary*: Official, often paid support.
* **Security**:
  + *Open-source*: Transparent, but depends on community vigilance.
  + *Proprietary*: Hidden code, relies on vendor for security.

**GIT and GITHUB Training Git Training:** **GitHub Training:**

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

* Version Control: Tracks changes, so team members can work on the same project without overwriting each other’s code.
* Branching: Allows developers to work on features or fixes in isolated environments.
* Merge & Conflict Resolution: Combines changes and manages conflicts efficiently.
* History Tracking: Keeps a detailed log of who changed what and why.
* Remote Repositories: Enables collaboration from anywhere via platforms like GitHub or GitLab.

**Application Software**

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

* Automates Tasks: Speeds up processes like accounting, HR, and inventory management.
* Improves Productivity: Helps employees complete work faster and more accurately.
* Data Management: Stores, processes, and analyzes business data efficiently.
* Communication: Facilitates internal and external communication (e.g., email, chat apps).
* Customer Service: Enhances support through CRM systems and helpdesk tools.

**Software Development Process**

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

1. Requirement Analysis – Understand what the software must do.
2. Design – Plan the system architecture and UI.
3. Implementation (Coding) – Write the actual code.
4. Testing – Find and fix bugs or errors.
5. Deployment – Release the software to users.
6. Maintenance – Update and fix the software post-release.

**Software Requirement**

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

* Defines Project Goals: Ensures everyone understands what the software must do.
* Prevents Miscommunication: Aligns stakeholders, developers, and users.
* Reduces Cost & Rework: Fixing issues early is cheaper than post-development.
* Guides Design & Development: Acts as the foundation for the next stages.
* Improves Quality: Leads to a product that meets user needs and expectations.

**Software Analysis**

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

* Understand Requirements: Translates user needs into technical specs.
* Identify Problems: Detects gaps, conflicts, or feasibility issues early.
* Plan Architecture: Guides system structure and functionality design.
* Support Decision-Making: Helps choose tools, technologies, and strategies.
* Ensure Alignment: Keeps development aligned with business goals

**System Design**

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

1. Architecture Design – Overall system structure and components.
2. Data Design – How data is stored, organized, and accessed.
3. Interface Design – User and system interaction points.
4. Component Design – Internal logic and functionality of modules.
5. Security Design – Measures to protect data and system integrity.
6. Performance Design – Ensures speed, scalability, and efficiency**.**

**Software Testing**

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

Software testing is important because it helps identify and fix bugs, ensures the software works as intended, improves quality, and increases user satisfaction. It also reduces the risk of failures and saves time and cost in the long run.

**Maintenance**

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

1. Corrective – Fixing bugs and errors.
2. Adaptive – Updating software for new environments or technologies.
3. Perfective – Enhancing performance or adding new features.
4. Preventive – Improving future maintainability and preventing issues.

**Development**

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

1. Installation: Web apps run in a browser; desktop apps need to be installed.
2. Accessibility: Web apps can be accessed from anywhere; desktop apps are limited to the device installed.
3. Updates: Web apps update centrally; desktop apps require manual or automatic local updates.
4. Performance: Desktop apps usually offer better performance and offline access.
5. Security: Web apps rely on internet security; desktop apps depend on local system security.

**Web Application**

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

1. Accessible anywhere with internet access.
2. No installation needed.
3. Easier to update and maintain.
4. Cross-platform compatibility.
5. Lower hardware requirements on user devices.

**Designing**

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

UI/UX design plays a crucial role in application development by ensuring the app is user-friendly, visually appealing, and intuitive, which enhances user satisfaction and engagement.

**Mobile Application**

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

Native apps are built specifically for one platform (iOS or Android) using platform-specific languages, offering better performance and access to device features. Hybrid apps use web technologies wrapped in a native container, work across multiple platforms but may have slower performance and limited access to device features.

**DFD (Data Flow Diagram)**

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

DFDs (Data Flow Diagrams) are significant in system analysis because they visually represent how data moves through a system, helping to understand processes, identify bottlenecks, and improve system design**.**

**Desktop Application**

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

* Pros:
  + Faster performance
  + Can work offline
  + Better access to hardware resources
* Cons:
  + Platform-dependent (needs different versions for each OS)
  + Requires installation and updates

Web Applications:

* Pros:
  + Accessible from any device with a browser
  + Easier to update and maintain
  + Cross-platform compatibility
* Cons:
  + Dependent on internet connection
  + Generally slower performance
  + Limited access to device hardware

**Flow Chart**

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

**ANS** - Flowcharts help by visually mapping out the logic and sequence of steps in programming and system design, making it easier to understand, communicate, and debug processes.

**Module 2 – Frontend - HTML**

**HTML Basics**

**Theory Assignment**

**Question 1: Define HTML. What is the purpose of HTML in web development?**

**ANS -** HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It defines elements like headings, paragraphs, links, images, and more.

**Question 2: Explain the basic structure of an HTML document. Identify the mandatory tagsand their purposes.**

**ANS -** A basic HTML document includes:

* <!DOCTYPE html> – Defines the document type
* <html> – Root element
* <head> – Contains metadata (title, links, etc.)
* <title> – Sets the page title
* <body> – Contains the visible content of the web page

**Question 3: What is the difference between block-level elements and inline elements in HTML? Provide examples of each.**

* Block-level elements start on a new line and take full width.  
  *Examples:* <div>, <p>, <h1>
* Inline elements do not start on a new line and only take as much width as needed.  
  *Examples:* <span>, <a>, <strong>

**• Question 4: Discuss the role of semantic HTML. Why is it important for accessibility andSEO? Provide examples of semantic elements.**

**ANS -** Semantic HTML uses meaningful tags that describe content (e.g., <header>, <article>, <nav>).  
It improves accessibility for screen readers and helps search engines understand content structure, boosting SEO.

**HTML Forms**

**Theory Assignment**

**Question 1: What are HTML forms used for? Describe the purpose of the input, textarea, select, and button elements.**

**ANS** - HTML forms collect user data.

* <input>: Takes user input (text, checkbox, etc.)
* <textarea>: Multi-line text input
* <select>: Dropdown list for options
* <button>: Submits or triggers actions in the form

**Question 2: Explain the difference between the GET and POST methods in form submission. When should each be used?**

**ANS -** GET: Sends data in the URL; used for simple, non-sensitive queries (e.g., search).

POST: Sends data in the body; used for sensitive or large data (e.g., login, registration).

**Question 3: What is the purpose of the label element in a form, and how does it improve accessibility?**

**ANS -** The <label> links text to a form input, helping screen readers identify fields and improving usability for all users.