**Module 1 :- Theory Assignment**

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

**Ans.** A program is a set of instructions a computer follows to perform a

Specific Task

2. What are the key steps involved in the programming process?  
 Types of Programming Languages

**Ans.** The programming process involves several key steps, from understanding the problem to documenting the solution. Types of programming languages include compiled, interpreted, procedural, object-oriented, and scripting languages

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

**Ans.**

High-level and low-level programming languages differ primarily in their abstraction level and ease of understanding. High-level languages are more human-readable and portable, while low-level languages offer more direct control over hardware but are harder to understand and less portable

4. World Wide Web & How Internet Works

**Ans.**

The Internet is a global network of interconnected computers, while the World Wide Web (WWW) is a system for accessing and sharing information on the Internet. Think of the Internet as the physical road network, and the WWW as the vehicles (websites) traveling on it.

5. Describe the roles of the client and server in web communication.  
Network Layers on Client and Server

**Ans.** In web communication, the client initiates requests for resources or services, while the server provides these resources and processes the client's requests. The network facilitates communication between them, using protocols to ensure reliable and secure data transfer

6. Describe the roles of the client and server in web communication.  
 Network Layers on Client and Server

**Ans .**  In web communication, the client initiates requests for resources or services, while the server provides these resources and processes the client's requests. The network facilitates communication between them, using protocols to ensure reliable and secure data transfer.

7. Explain the function of the TCP/IP model and its layers.  
 Client and Servers

**Ans.**  The TCP/IP model is a fundamental framework for computer networking. It stands for Transmission Control Protocol/Internet Protocol, which are the core protocols of the Internet. This model defines how data is transmitted over networks, ensuring reliable communication between devices. It consists of four layers: the Link Layer, the Internet Layer, the Transport Layer, and the Application Layer. Each layer has specific functions that help manage different aspects of network communication, making it essential for understanding and working with modern networks.

8. Explain Client Server Communication Types of Internet Connections

**Ans.** Client-server communication over internet connections primarily involves a client making a request to a server, which then responds. This model is widely used for various services like web browsing, email, and file sharing. There are different ways this communication can be structured, including request-response, streaming, and messaging

9. How does broadband differ from fiber-optic internet?  
 Protocols

**Ans.** Broadband refers to a general category of high-speed internet connections, while fiber-optic internet is a specific type of broadband that uses light to transmit data over glass or plastic fibers.

10. What are the differences between HTTP and HTTPS protocols?

**Ans.**

| **HTTP** | **HTTPS** |
| --- | --- |
| HTTP stands for HyperText Transfer Protocol. | HTTPS for HyperText Transfer Protocol Secure. |
| In HTTP, URL begins with “http://”. | In HTTPs, URL starts with “https://”. |
| HTTP uses port number 80 for communication. | HTTPs uses 443 port number for communication. |
| HTTP is considered to be unsecure. | HTTPs is considered as secure. |
| HTTP works at Application Layer. | HTTPS works at Transport Layer. |
| In HTTP, Encryption is absent. | Encryption is present in HTTPS. |
| HTTP does not require any certificates. | HTTPS needs SSL Certificates. |
| HTTP does not improve search ranking | HTTPS helps to improve search ranking |
| HTTP faster than HTTPS | HTTPS slower than HTTP |
| HTTP does not use data hashtags to secure data. | While HTTPS will have the data before sending it and return it to its original state on the receiver side. |
| In HTTP Data is transfer in plaintext. | In HTTPS Data transfer in ciphertext. |
| HTTP Should be avoided. | HTTPS Should be preferred. |
| Search engines do not favour the insecure website. | Improved reputation of the website in search engine. |
| HTTP Does not require SSL/TLS or Certificates | HTTPS Requires SSL/TLS implementation with Certificates. |
| In HTTP Users ar  worried about their data. | In HTTPS Users are  confident about the security of their data. |

10. What is the role of encryption in securing applications  
 Software Applications and Its Types

**Ans.**  Encryption plays a crucial role in securing software applications by safeguarding data confidentiality, integrity, and authentication. It converts data into an unreadable format, ensuring only authorized parties with the decryption key can access it. This protection is vital for both data at rest (stored on a device) and data in transit (being transmitted over a network). Encryption helps prevent unauthorized access, interception, and alteration of sensitive information, enhancing the overall security of applications and their users

11. What is the difference between system software and application software?  
 Software Architecture

**Ans.**

System software provides the foundation for a computer's hardware and application software to function, while application software is designed for specific tasks performed by users. System software manages hardware and resources, whereas application software allows users to perform tasks like word processing or web browsing. Software architecture, in contrast, is the blueprint for a software system, encompassing the overall structure and relationships between its components.

12. Why are layers important in software architecture? Software Environments

**Ans.** Layers are crucial in software architecture because they enable modularity, separation of concerns, and independent scaling, leading to more maintainable, reusable, and scalable applications. By dividing an application into distinct layers, each with its own responsibility, development, testing, and maintenance become significantly easier.

13. Explain the importance of a development environment in softwareproduction .Source Code

**Ans.**

A development environment is crucial in software production as it provides a controlled and isolated space where developers can write, test, and debug code without affecting the live application or users. This isolated workspace allows for experimentation, error-free code changes, and efficient workflows, ultimately leading to higher quality software and reduced development time.

14. What is the difference between source code and machine code?   
 Github and Introductions

**Ans.** Source code is the human-readable programming code written by developers, while machine code is the binary code that computers directly execute. Source code is translated into machine code by a compiler or interpreter before a computer can run it. GitHub is a platform that hosts and manages repositories of source code using the Git version control system, facilitating collaboration and version control.

15. Why is version control important in software development?  
 Student Account in Github

**Ans.** Version control, particularly using systems like Git and platforms like GitHub, is crucial in software development because it allows developers to track changes to their codebase, collaborate effectively, and manage different versions of the project. This system helps to avoid errors, maintain code integrity, and facilitate efficient teamwork. For students, a GitHub account provides a great platform to learn and practice using version control, fostering valuable skills for future careers.

16. What are the benefits of using Github for students?  
 Types of Software

**Ans.**  GitHub benefits students by offering a platform for collaborative coding, version control, and showcasing their projects. It also provides access to resources like the GitHub Student Developer Pack, which includes free tools and services, and encourages participation in the open-source community.

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

GIT and GITHUB Training

Ans. Open-source software is free to use, modify, and distribute, while proprietary software is not. Open-source code is publicly available, allowing anyone to inspect and contribute, while proprietary software has restricted access and licensing.

18. How does GIT improve collaboration in a software development team?  
 Application Software

**Ans.**  Git enhances collaboration in software development by providing version control, branching and merging capabilities, and facilitating distributed development. It allows developers to track changes, work concurrently on different features, and easily merge their contributions without disrupting the main codebase.

19. What is the role of application software in businesses?  
 Software Development Process

**Ans.**  Application software plays a crucial role in businesses by enabling various tasks and processes, ultimately increasing productivity and efficiency. It facilitates data management, communication, automation of repetitive tasks, and improved customer engagement, leading to better decision-making and overall business performance. The software development process, which involves designing, coding, testing, and maintaining applications, is vital for creating and evolving these tools to meet evolving business needs

20. What are the main stages of the software development process?  
 Software Requirement

**Ans.** The main stages of the software development process, often referred to as the Software Development Life Cycle (SDLC), typically involve planning, analysis, design, coding, testing, deployment, and maintenance. A crucial stage within this process is requirement gathering, where the needs of the users and the system are identified and documented

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

Software Analysis

**Ans.** The requirements analysis phase is critical in software development because it establishes a clear understanding of the project's goals, user needs, and potential issues early on, preventing costly rework later. It ensures the developed software aligns with stakeholder expectations and minimizes risks associated with scope creep or feature creep.

22. What is the role of software analysis in the development process?  
System Design

**Ans.** Software analysis is crucial in the development process because it establishes the foundation for successful software design and implementation. It involves understanding and documenting the user's needs, requirements, and the system's functionalities, ensuring the final product meets all specified needs. This analysis guides the design phase, ensuring that the software architecture and implementation are aligned with the identified requirements.

23. What are the key elements of system design?  
Software Testing

**Ans.**  Key elements of system design include architecture, modules, components, interfaces, and data management. These elements are crucial for building a well-designed software system. Software testing plays a vital role in verifying the system's functionality and ensuring it meets requirements.

24. Why is software testing important?  
 Maintenance

**Ans.**  Software testing is crucial for maintenance because it ensures the stability and reliability of the software after its initial release. By identifying and fixing bugs, vulnerabilities, and compatibility issues, testing helps to minimize disruptions and reduce long-term maintenance costs. It also ensures that modifications or updates don't negatively impact existing functionality, ultimately improving the user experience and customer satisfaction

25. What types of software maintenance are there?  
 Development

**Ans.**  Software maintenance is broadly categorized into four main types: corrective, adaptive, perfective, and preventive. Corrective maintenance focuses on fixing bugs and errors after the software is released. Adaptive maintenance involves modifying the software to adapt to changes in the environment or user requirements. Perfective maintenance aims to improve the software's performance, functionality, or usability. Preventive maintenance is proactive, focusing on preventing future issues by modifying the software to enhance its robustness and reliability.

26. What are the key differences between web and desktop applications? • Web Application

**Ans.** Web and desktop applications differ primarily in accessibility and deployment. Web apps are accessed through a web browser and can be used on any device with an internet connection, while desktop apps are installed locally on a specific computer. Web apps offer flexibility and accessibility, but may have slower performance and limited access to hardware features. Desktop apps offer better performance and access to hardware features, but are tied to a specific device and require installation.

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

Designing.

**Ans.**  Web applications offer several advantages over desktop applications, primarily due to their accessibility, ease of maintenance, and cross-platform compatibility. They are accessible from any device with an internet connection, eliminating the need for installation and updates, which simplifies the user experience. Additionally, web apps can be developed and maintained more cost-effectively, and they often scale better to accommodate larger user bases.

28. What role does UI/UX design play in application development? Mobile Application

**Ans.** UI/UX design plays a crucial role in mobile application development, directly impacting usability, user satisfaction, retention, and app success. It's essential for creating intuitive and aesthetically pleasing interfaces that encourage user engagement and loyalty. A well-designed app can increase user engagement, leading to users spending more time on the app.

29.   
 What are the differences between native and hybrid mobile apps?  
• DFD (Data Flow Diagram)

**Ans**. Native and hybrid mobile apps differ primarily in their development approach, performance, and cost. Native apps are built using platform-specific languages, offering superior performance and access to device features, but require separate development for each platform. Hybrid apps are built using web technologies and run within a native shell, offering cross-platform compatibility and faster development at a lower cost, but potentially sacrificing some performance and access to native features.

30. What is the significance of DFDs in system analysis?  
• Desktop Application

**Ans.**

Data Flow Diagrams (DFDs) are crucial in system analysis, especially for desktop applications, because they provide a visual representation of how data moves within a system. This visual clarity helps in understanding system processes, identifying potential bottlenecks, and facilitating communication between technical and non-technical staff.

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

**Ans.** Desktop applications generally offer better performance and more robust features but require installation and updates, while web applications are accessible from anywhere and offer automatic updates, but may have slower performance and security concerns

32. How do flowcharts help in programming and system design?

**Ans.** Flowcharts help in programming and system design by providing a visual representation of the logic and flow of a process, making it easier to understand, plan, and debug. They break down complex problems into smaller, manageable steps, allowing for better organization and communication.