Name :- Gajera Pal.

Module 1 – Overview of IT Industry

**Q1:- Explain in your own words what a program is and how it functions. What is Programming?**

**ANS:-**

A computer program is a sequence or set of instructions in a programming language for a computer to execute.

**Q2:- What are the key steps involved in the programming process? Types of Programming Languages?**

**ANS:-**

**Key Steps in the Programming Process:**

1. Problem Analysis
2. Design
3. Coding
4. Testing
5. Debugging
6. Maintenance

**Types of Programming Languages**

1. Low-level Languages:
2. High-level Languages:
3. Scripting Languages:
4. Database Query Languages

**Q3:- What are the main differences between high-level and low-level programming languages?  
ANS:-**

**High-level languages:**

* Easier to use, closer to human language (e.g., Python, Java).
* Abstracts hardware details.

**Low-level languages:**

* Closer to machine code (e.g., Assembly, Machine Language).
* Requires manual control of hardware resources.

**Q4:- Describe the roles of the client and server in web communication**

**ANS:-**

**1. Client:**

* Role: The client is typically a web browser or application that sends requests to the server for resources, like web pages, images, or data.
* **Function:**
  + The client initiates the communication by making requests (such as entering a URL or submitting a form).
  + It sends these requests over the internet using the HTTP/HTTPS protocol.

**2. Server:**

* Role: The server is a system that hosts the website or application, and it processes requests from clients.
* **Function:**
  + The server receives requests from the client, processes them (e.g., fetching data from a database or serving a web page), and sends back a response.
  + The server stores and manages the content or data that the client requests.

**Q5:- Explain the function of the TCP/IP model and its layers. Client and Servers?**

**ANS:-**

The TCP/IP model is a framework used for communication over the internet. It defines how data is transmitted between computers and networks. It consists of four layers:

1. Application Layer
2. Transport Layer
3. Internet Layer
4. Link Layer

**Q6:- Explain Client Server Communication?**

**ANS:-**

1. Client sends a request (e.g., for a web page or data).
2. Server processes the request and retrieves the necessary data.
3. Server sends the response back to the client.
4. The client displays or uses the received data.

**Q7:- How does broadband differ from fiber-optic internet? Protocols ?**

**ANS:-**

**Key Differences:**

* Speed: Fiber-optic is faster, often offering speeds up to 1 Gbps or more.
* Technology: Fiber-optic uses light through fiber cables, while broadband can use DSL, cable, or satellite.

**Common Internet Protocols:**

* HTTP/HTTPS: Used for web page communication.
* TCP/IP: Core protocols for data transmission.
* FTP: Transfers files between devices.

**Q8:- What are the differences between HTTP and HTTPS protocols? Application Security.**

**ANS:-**

|  |  |  |
| --- | --- | --- |
|  | **HTTP** | **HTTPS** |
| **Security** | It is an insecure protocol. Data sent over HTTP is transmitted in plain text, meaning it can be intercepted and read by anyone with access to the data transmission path. | It is a secure version of HTTP. It uses SSL/TLS encryption to encrypt data, ensuring that information transmitted between the client (e.g., web browser) and server is secure and cannot be easily intercepted or tampered with. |
| **Encryption** | No encryption, so data is sent in plain text. | Uses SSL/TLS encryption to secure the communication. |
| **Port** | Uses port 80 for communication. | Uses port 443 for secure communication. |
| **URL** | URLs begin with http://. | URLs begin with https://. |

**Q9:- What is the role of encryption in securing applications?**

**ANS:-**

**1. Confidentiality:**

* Role: Encryption ensures that sensitive data (like passwords, personal information, or payment details) is unreadable to unauthorized parties.

**2. Data Integrity:**

* Role: Encryption helps verify that the data has not been tampered with during transmission. If the data is altered in any way, it will be detected when trying to decrypt it.

**3. Authentication:**

* Role: Encryption also supports the verification of the sender's identity, ensuring that the data came from a trusted source and hasn't been spoofed.

**4. Protection Against Eavesdropping:**

* Role: It prevents unauthorized parties (such as hackers or malicious third parties) from reading the communication between users.

**5. Prevention of Data Theft:**

* Role: Even if encrypted data is stolen, the attacker cannot use it without the decryption key, which protects against data theft.

**Q10:- What is the difference between system software and application software?**

**ANS:-**

System software is the software designed to run and manage computer hardware, providing a platform for applications to function. It includes the operating system.

Application software is designed to perform specific tasks or solve particular problems for users, such as word processors (e.g., Microsoft Word), web browsers (e.g., Chrome), or games.

**Q11:- What is the significance of modularity in software architecture? Layers in Software Architecture?**

**ANS:-**

1. **Ease of Maintenance**: Isolating changes to specific modules reduces the impact on other parts of the system.
2. **Reusability**: Modules can be reused across different applications or systems.
3. **Scalability**: New functionality can be added without disrupting existing code.
4. **Improved Collaboration**: Different teams can work on separate modules simultaneously.
5. **Simplified Debugging**: Issues are easier to identify and resolve in isolated modules.

**Layers in Software Architecture:**

1. **Presentation Layer**: Deals with user interfaces and user experience.
2. **Application Layer**: Contains the business logic and core functionality.
3. **Data Layer**: Manages data storage and retrieval.
4. **Infrastructure Layer**: Handles underlying services like networking, security, and hardware interaction.

**Q12:- Why are layers important in software architecture?**

**ANS:-**

Layers in software architecture are important because they:

1. **Separation of Concerns**: Each layer handles a specific aspect of the system, making it easier to understand, develop, and maintain.
2. **Modularity**: Layers allow independent development and testing, reducing complexity.
3. **Reusability**: Layers can be reused across different applications or projects, especially the core business logic and data layers.
4. **Flexibility**: Changes in one layer (e.g., UI changes) don't impact others (e.g., data storage), promoting easier updates and scalability.
5. **Maintainability**: Isolated layers make it simpler to identify and fix issues without affecting the whole system.

**Q13:- Explain the importance of a development environment in software production?**

**ANS:-**

1. **Efficiency**: Streamlined tools (editors, compilers, debuggers) accelerate development.
2. **Consistency**: Ensures all developers work with the same setup, minimizing environment-related issues.
3. **Error Reduction**: Integrated features like error highlighting and real-time debugging help catch issues early.
4. **Collaboration**: Supports version control and collaboration tools, enabling team-based development.
5. **Testing & Deployment**: Facilitates easy testing, integration, and deployment of software.

**Q14:- What is the difference between source code and machine code?**

**ANS:-**

Source code is the human-readable set of instructions written by programmers using high-level programming languages (e.g., Python, Java, C++).

Machine code is the low-level, binary code (composed of 0s and 1s) that a computer's processor can directly execute.

**Q15:- Why is version control important in software development?**

**ANS:-**

Version control is important in software development because it:

1. **Tracks Changes**: Keeps a history of code changes, allowing developers to revert to previous versions if needed.
2. **Collaboration**: Enables multiple developers to work on the same project without overwriting each other's work.
3. **Code Integrity**: Helps manage and resolve conflicts in code when changes are made by different team members.
4. **Backup**: Provides a safety net by maintaining backups of the code.
5. **Efficiency**: Facilitates branching and merging, allowing teams to work on features or fixes in parallel.

**Q16:- What are the benefits of using Github for students?**

**ANS:-**

**Benefits of using GitHub for students** **include**:

1. **Collaboration**: GitHub allows students to work on projects together, share code, and contribute to open-source projects.
2. **Version Control**: It helps students track changes to their code, revert to previous versions, and manage project history.
3. **Portfolio Building**: Students can showcase their projects and coding skills to potential employers by hosting repositories on GitHub.
4. **Learning Git**: GitHub helps students learn Git, a widely-used version control system in the software industry.
5. **Access to Open Source**: Students can contribute to or learn from millions of open-source projects, enhancing their learning experience.

**Q17:- What are the differences between open-source and proprietary software?**

**ANS:-**

|  |  |  |
| --- | --- | --- |
|  | open-source | proprietary software |
| **Source Code Access**: | The source code is publicly available, allowing anyone to view, modify, and distribute it.   | The source code is closed and owned by the creator, and users cannot modify or distribute it. |
| Cost | Often free to use, though some may offer paid versions or support. | Typically requires a paid license for use, with ongoing costs for updates or support |
| Licensing | Distributed under licenses that promote freedom to modify and share, like GPL or MIT. | Restricted by the software maker's terms, limiting usage, modification, and distribution. |
| Support | Support is often community-driven or provided by third parties. | Official support is provided by the software vendor. |

**Q18:- How does GIT improve collaboration in a software development team?**

**ANS:-**

1. **Version Control**: Git tracks changes to the codebase, allowing team members to work on different parts of the project without overwriting each other's work.
2. **Branching and Merging**: Developers can create branches to work on new features or bug fixes independently.
3. **Conflict Resolution**: If two developers make conflicting changes to the same part of the code, Git helps identify.

**Q19:- What is the role of application software in businesses?**

**ANS:-**

application software helps businesses work more efficiently. It includes programs like word processors, spreadsheets, and tools for managing customers or finances

**Q20:- What are the main stages of the software development process?**

**ANS:-**

The main stages of the software development process are:

1. **Planning**: Define the project scope, goals, and requirements.
2. **Design**: Create the software architecture and design its components.
3. **Development**: Write the code based on the design specifications.
4. **Testing**: Ensure the software works correctly by identifying and fixing bugs.
5. **Deployment**: Release the software to users.
6. **Maintenance**: Provide updates and fixes to ensure long-term functionality.

**Q21:- Why is the requirement analysis phase critical in software development?**

**ANS:-**

The requirement analysis phase is critical in software development because it ensures that the project meets the users' needs and expectations.

**Q22:- What is the role of software analysis in the development process?**

**ANS:-**

1. **Clarifying Requirements**: Ensuring that all stakeholders' needs are understood and documented clearly.
2. **Feasibility Study**: Analyzing technical, financial, and operational feasibility to determine whether the project is viable.
3. **Design Foundation**: Laying the groundwork for system design by identifying key components and interactions.
4. **Risk Identification**: Highlighting potential risks early in the project to mitigate issues later in development.

**Q23:- What are the key elements of system design?**

**ANS:-**

Key elements of system design include:

1. **Architecture Design**: Overall system structure and component interactions.
2. **Data Design**: Organizing and storing data effectively.
3. **Interface Design**: User and system interaction, including UIs and APIs.
4. **Component Design**: Defining individual software components.
5. **Security Design**: Ensuring data protection and secure access.
6. **Performance Design**: Optimizing speed, scalability, and efficiency.
7. **Error Handling**: Managing errors and system logs.

**Q24:- Why is software testing important?**

**ANS**

Software testing is important because it ensures the software functions as expected, is free of defects, and meets user requirements.

**Q25:- What types of software maintenance are there?**

**ANS**

1. **Corrective Maintenance**: This type focuses on fixing defects or bugs in the software that were discovered after its release.
2. **Adaptive Maintenance**: Adaptive maintenance involves making changes to the software to keep it compatible with new environments, technologies, or platforms.
3. **Perfective Maintenance**: Perfective maintenance aims at enhancing the software’s performance, usability, or features.
4. **Preventive Maintenance**: Preventive maintenance is focused on reducing the likelihood of future issues by identifying potential problems and addressing them before they occur.

**Q26:- What are the key differences between web and desktop applications?**

**ANS**

Key differences between web and desktop applications:

1. **Platform**:
   * **Web applications** run on a web browser and are accessible via the internet, making them platform-independent.
   * **Desktop applications** are installed on a specific operating system (Windows, macOS, etc.) and are limited to that platform.
2. **Access**:
   * **Web applications** require an internet connection for full functionality.
   * **Desktop applications** can often run offline, once installed.
3. **Updates**:
   * **Web applications** are updated centrally, meaning users always access the latest version.
   * **Desktop applications** require manual updates, either by the user or through automatic update features.
4. **Performance**:
   * **Web applications** may be slower due to dependency on internet speed and browser capabilities.
   * **Desktop applications** generally offer better performance, as they directly utilize system resources.
5. **Installation**:
   * **Web applications** don’t require installation; they run directly in a browser.
   * **Desktop applications** need to be installed on a device before use.

**Q27:- What are the advantages of using web applications over desktop applications?**

**ANS**

Advantages of using web applications over desktop applications:

1. **Accessibility**: Web apps can be accessed from any device with an internet connection and a browser, offering greater flexibility.
2. **Centralized Updates**: Updates are applied automatically, ensuring users always have the latest version without manual intervention.
3. **Cross-Platform Compatibility**: Web apps work across different operating systems (Windows, macOS, Linux) without the need for separate versions.
4. **Reduced Installation**: Users don't need to install or maintain the software on their devices, simplifying access.
5. **Scalability**: Web apps can easily scale to accommodate more users or features, as they are hosted on servers rather than individual devices.

**Q28:- What role does UI/UX design play in application development?**

**ANS**

UI/UX design plays a crucial role in application development by focusing on the user’s experience and interaction with the software.

**UI (User Interface)** design ensures the application is visually appealing, easy to navigate, and intuitive, while **UX (User Experience)** design optimizes the overall usability and satisfaction of users.

**Q29:- What are the differences between native and hybrid mobile apps?**

**ANS**

1. **Platform**:
   * **Native apps** are built specifically for one platform (iOS or Android) using platform-specific programming languages.
   * **Hybrid apps** are built using web technologies (HTML, CSS, JavaScript) and can run on multiple platforms using a single codebase.
2. **Performance**:
   * **Native apps** generally offer better performance as they directly interact with the device’s hardware and operating system.
   * **Hybrid apps** may be slower, as they rely on a web view and don’t have the same level of direct access to device features.
3. **Development Cost**:
   * **Native apps** require separate development for each platform, making them more expensive and time-consuming.
   * **Hybrid apps** use a single codebase, which makes them more cost-effective and quicker to develop for multiple platforms.
4. **User Experience**:
   * **Native apps** provide a smoother and more responsive experience, with better integration into the device's UI/UX.
   * **Hybrid apps** may not always match the native user experience due to limitations in performance and UI elements.

**Q30:- What is the significance of DFDs in system analysis?**

**ANS**

1. **Visualizing Data Flow**: DFDs help analysts and stakeholders understand how data moves through the system, from input to processing to output, simplifying complex processes.
2. **Clarifying System Processes**: They break down the system into smaller, manageable processes, making it easier to identify inefficiencies, bottlenecks, or areas for improvement.
3. **Communication Tool**: DFDs serve as a communication tool between technical teams and non-technical stakeholders, ensuring that everyone has a clear understanding of the system’s functionality.
4. **Identifying Data Sources and Destinations**: They help identify where data originates and where it goes, aiding in understanding system requirements and designing effective solutions.
5. **Documentation and Analysis**: DFDs document the system's functional flow and provide a foundation for future design and development, making it easier to analyze system behavior.

**Q31:- What are the pros and cons of desktop applications compared to web applications?**

**ANS**

**Pros of Desktop Applications**:

1. **Performance**: Typically faster and more responsive as they run directly on the user's device.
2. **Offline Access**: Can be used without an internet connection.

**Cons of Desktop Applications**:

1. **Platform Dependency**: Need separate versions for different operating systems (Windows, macOS, etc.).
2. **Installation Required**: Users must install and update the software.

**Pros of Web Applications**:

1. **Cross-Platform**: Accessible from any device with a browser and internet connection.
2. **No Installation**: Users don’t need to install or update software.

**Cons of Web Applications**:

1. **Requires Internet**: Generally needs a stable internet connection.
2. **Performance**: May be slower due to dependency on browser and network speed.

**Q32:- How do flowcharts help in programming and system design?**

**ANS**

1. **Clarify Logic**: Simplify complex processes and decision-making steps.
2. **Improve Communication**: Facilitate discussion and understanding among team members or stakeholders.
3. **Identify Issues**: Help pinpoint potential problems or inefficiencies in the flow of the process.
4. **Guide Development**: Act as a roadmap for developers, making coding and system design more structured and organized.