Module1

* Theory Exercise

1. Explain in your own words what a program is and how it functions.
   * + - A program is a set of instruction that tells the computer what to do.
       - It functions into three steps (Input, Processing, Output). The set of instruction given to the computer is called input, after it started to process is called processing and the output given by the computer is called output. This is how the program functions in a computer.
2. What are the key steps involved in the programming process?

* Problem Definition, Planning and design, Coding, Testing and debugging, Refinement and Optimization, Deployment, Maintenance and Update.

1. What are the main differences between high-level and low-level programming languages?

* High level language= It is closer to human language. They are easy to read and it have easy syntax. Not need of much specification for the programming.
* Low level language= It is much closer to the machine language. They are hard to read and understanding the syntax compared to high level language. Needs to specify everything for the programming.
* Conclusion= High level= Easier for humans

Low level= Easier for machines

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

* Client:
* **Initiates communication**: The client sends a request to the server for resources like web pages, images, or data.
* **User interface**: It provides the interface through which users interact with web applications.
* **Processes server responses**: After receiving data from the server, the client renders it.
* **Examples**: Chrome, Firefox, mobile apps, etc.
* Server:
* **Responds to requests**: The server listens for incoming requests and sends back appropriate responses.
* **Hosts resources**: Stores and manages web content, databases, and application logic.
* **Handles business logic**: Processes data, performs operations, and enforces rules before sending responses.
* **Examples**: Apache, Nginx, Node.js servers, etc.

1. Explain the function of the TCP/IP model and its layers.
2. Explain Client Server Communication.

* **Client Initiates Request**
  + A user opens a browser and types a URL.
  + The browser (client) sends a request to the web server.
* **Server Processes Request**
  + The server receives the request.
  + It processes it—fetching data, running logic, or accessing databases.
* **Server Sends Response**
  + The server sends back the requested data (e.g., a web page).
  + The client displays it to the user.
* **Connection May Close or Persist**
* Depending on the protocol (like HTTP/1.1 vs HTTP/2), the connection may close or stay open for further requests

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

|  |  |  |
| --- | --- | --- |
| Feature | Broadband | Fiber-Optic Internet |
| Medium | Copper wires (DSL, coaxial cable) | Glass or plastic fibers |
| Signal Type | Electrical signals | Pulses of light |
| Speed | 10 Mbps to 1 Gbps | 50 Mbps to 2 Gbps or more |

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

|  |  |  |
| --- | --- | --- |
| Feature | HTTP | HTTPS |
| Full Form | HyperText Transfer Protocol | HyperText Transfer Protocol |
| Security | No encryption; data sent as plain text | Encrypted using SSL/TLS |
| Port Used | Port 80 | Port 443 |
| Data Protection | Vulnerable to interception and tampering | Protects data from eavesdropping and tampering |
| Certificate | No certificate required | Requires SSL/TLS certificate from a trusted CA |
| Trust Indicator | No padlock icon in browser | Padlock icon and “https://” in URL |
| SEO Benefits | No ranking boost | Preferred by search engines; boosts SEO |
| Performance | Slightly faster due to no encryption overhead | Slightly slower but more secure |

1. What is the role of encryption in securing applications?

* Encryption plays a **critical role** in securing applications by acting as a digital lockbox for sensitive data. It transforms readable information (plaintext) into an unreadable format (ciphertext), ensuring that only authorized parties with the correct key can access it.

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

|  |  |  |
| --- | --- | --- |
| Feature | System Software | Application Software |
| Purpose | Manages hardware and provides a platform for apps | Helps users perform specific tasks |
| User Interaction | Works mostly in the background | Directly interacts with the user |
| Language Used | Written in low-level languages (e.g., C, Assembly) | Written in high-level languages (e.g., Java, Python) |
| Examples | Operating systems, device drivers, compilers | Word processors, web browsers, games |
| Execution | Starts automatically when the system boots | Runs when the user initiates it |
| Dependency | Independent; required for the system to function | Dependent on system software to run |
| Functionality | Controls system resources and hardware | Performs user-defined tasks like editing or browsing |

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

* Modularity is like the secret sauce behind scalable, maintainable, and flexible software systems. It’s the principle of **breaking down a complex system into smaller, independent, and cohesive components** called modules that can be developed, tested, and maintained separately.

1. Why are layers important in software architecture?

* Layers are the **architectural backbone** of many software systems—they bring order, clarity, and scalability to complex applications.

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

* A **development environment** is absolutely vital in software production—it’s the workspace where ideas become code and code becomes functioning software. Think of it as the digital equivalent of a workshop for a craftsman.

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

|  |  |  |
| --- | --- | --- |
| Feature | Source Code | Machine Code |
| Definition | Human-readable instructions written in programming languages | Binary instructions understood by the computer's CPU |
| Written By | Programmers | Generated by compilers or assemblers |
| Languages Used | High-level (e.g., Python, Java, C++) or assembly | Binary (0s and 1s) or hexadecimal |
| Readability | Easy for humans to read and modify | Not readable by humans; meant for machines |
| Purpose | To define logic, behavior, and structure of programs | To execute instructions directly on hardware |
| Execution | Needs to be compiled or interpreted | Executed directly by the CPU |
|  |  |  |

1. Why is version control important in software development?

* Tracks changes made to code over time
* Allows collaboration among multiple developers
* Helps revert to previous versions when needed
* Prevents loss of work through backups
* Makes it easier to identify and fix bugs
* Supports working on multiple features in parallel
* Maintains a clear history of project development

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

* Benefits of Using GitHub for Students:
* Provides free cloud storage for code and projects
* Offers collaboration tools for group assignments
* Tracks change and maintain version history
* Allows showcasing projects to potential employers
* Gives access to open-source projects for learning
* Integrates with various development tools
* Offers free student benefits through GitHub Student Pack

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

|  |  |  |
| --- | --- | --- |
| Feature | Open-Source Software | Proprietary Software |
| Source Code | Public and editable | Private and restricted |
| Cost | Usually, free | Usually paid |
| Customization | Highly customizable | Limited or not allowed |
| Support | Community-based (optional paid support) | Official vendor support |
| Licensing | Flexible, often under GPL or MIT | Strict, with usage limitations |
| Security | Security | Controlled by vendor |

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

* git improves collaboration in a software development team by:
* Tracks all changes made to the codebase
* Allows multiple developers to work on the same project simultaneously
* Supports branching for developing new features without affecting main code
* Merges changes from different team members efficiently
* Maintains a history of who made which changes and why

Helps resolve conflicts when changes overlap

* Enables reverting to previous versions if issues occur

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

* Role of Application Software in Businesses:
* Automates routine business tasks
* Improves productivity and efficiency
* Facilitates data storage, management, and analysis
* Enhances communication and collaboration
* Supports decision-making with accurate information
* Improves customer service and engagement

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

* Main Stages of the Software Development Process:
* Requirement analysis
* System design
* Implementation
* Testing and debugging
* Deployment
* Maintenance and updates

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

* The requirement analysis phase is critical in software development because it identifies the needs and expectations and ensuring that the project starts with clear objectives

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

* The role of software analysis in the development process is to thoroughly examine and understand the problem, requirements, and constraints before designing the solution. It involves studying user needs, system functionalities, and technical feasibility to ensure the software run properly.

1. What are the key elements of system design?

* Key Elements of System Design**:**
* System architecture design
* Data design
* Interface design
* Process design
* Security design

1. Why is software testing important?

* Software testing is important because it ensures that the software works as intended and meets user requirements. It helps detect and fix bugs before the product is released, improving the overall quality, reliability, and security of the application.

1. What types of software maintenance are there?

* The steps to maintain the software are:
* Corrective maintenance
* Adaptive maintenance
* Perfective maintenance
* Preventive maintenance

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

* Key Differences between Web and Desktop are:
* Platform dependency
* Installation requirement
* Accessibility
* Update process
* Performance
* Internet requirement
* Storage location

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

* Advantages of Using Web Applications over Desktop Applications**:**
* Accessible from any device with an internet connection
* No installation required
* Easy and quick updates for all users
* Cross-platform compatibility
* Centralized data storage
* Easier collaboration and sharing

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

* UI/UX design plays a vital role in application development by enhancing user satisfaction and engagement through intuitive and visually appealing interfaces. It improves usability and accessibility, ensuring that users can navigate the application smoothly and efficiently.

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

|  |  |  |
| --- | --- | --- |
| Feature | Native Apps | Hybrid Apps |
| Platform | Built for a specific OS (iOS or Android) | Built for multiple platforms using one codebase |
| Languages Used | Swift/Objective-C (iOS), Kotlin/Java (Android) | HTML, CSS, JavaScript with frameworks like Ionic or Cordova |
| Performance | High performance and responsiveness | Moderate performance; depends on web rendering |
| Access to Device Features | Full access to hardware (camera, GPS, sensors) | Limited or indirect access via plugins |
| User Experience (UX) | Seamless, platform-consistent UI and gestures | May feel less native; UI can vary across devices |
| Development Time | Longer; separate codebases for each platform | Faster; single codebase for all platforms |
| Maintenance Cost | Higher; updates needed for each platform | Lower; one update applies to all |
| App Store Deployment | Distributed via app stores | Also distributed via app stores |

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

* The significance of Data Flow Diagrams (DFDs) in system analysis is that they visually represent how data moves within a system, showing inputs, processes, outputs, and storage points.

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

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Data can be stored locally for more control | Risk of data loss if local storage is compromised |
| Can be accessed from any device with internet | Requires stable internet connection |
| Updates are instant and available to all users | Performance may depend on server and network speed |
| Lower installation and maintenance costs | May have reduced performance compared to desktop apps |

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

* Flowcharts help in programming and system design by visually representing the sequence of steps, decisions, and processes involved in a program or system. They make complex logic easier to understand by breaking it into clear, graphical symbols and flows. This improves communication between developers, designers reduce misunderstandings, and helps identify errors or inefficiencies early. Flowcharts also serve as a useful reference during coding and debugging, ensuring that the final implementation follows the planned logic accurately.