MODULE 1

SE-Overview of IT Industry

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

* Program is a set of instructions written in a programming language that tells computer what to do.
* How a program functions:

1. Written by a programmer.
2. Complied or Interpreted.
3. Executed by the CPU.
4. Input/Output.
5. What are the main differences between high-level and low-level programming languages?

|  |  |  |
| --- | --- | --- |
| Sr. No | High-Level | Low-Level |
| 1 | Closer to human language: Easier to read, write and understand | Closer to machine language: Harder to read but more control over hardware. |
| 2 | Abstraction: They hide complex hardware | Minimal Abstraction: work directly with memory and CPU instructions. |
| 3 | Portable: Can run on different types of hardware with minimal charges. | Not portable: Usually written for specific hardware or processors. |
| 4 | Slower: Generally Slower | Faster and more efficient: because they interact directly with hardware |
| 5 | Example: Python | Example: Assembly |

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

* Roles of client and server in web communication:
* Client:

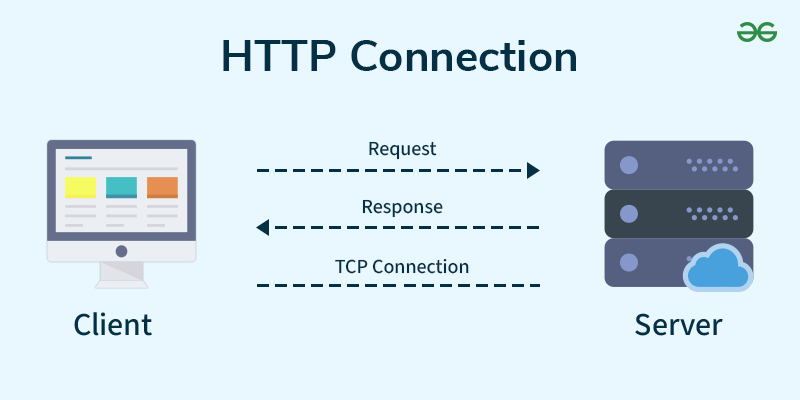
1. Usually a web browser or mobile app.
2. Initiates requests to the server.
3. Receives and displays data from the server.

* Server:

1. A computer/system that waits for client requests.
2. Processes the request, fetches or generates data and sends a response.
3. Can host websites, APIs, database, etc.
4. Explain the function of the TCP/IP model and its layers.

|  |  |  |
| --- | --- | --- |
| Layer | Client Role | Server Role |
| Application | Send HTTP request. | Receives HTTP request, send response. |
| Transport | Breaks data into packets. | Reassembles packets, ensures reliable delivery. |
| Internet | Adds IP addresses to route data. | Uses IP identify source and destination. |
| Network Access | Sends bits over physical medium. | Receives bits, passes up the layers. |

* **Lab Exercise**: Design a simple HTTP client-server communication in any language.



1. Explain Client Server Communication?

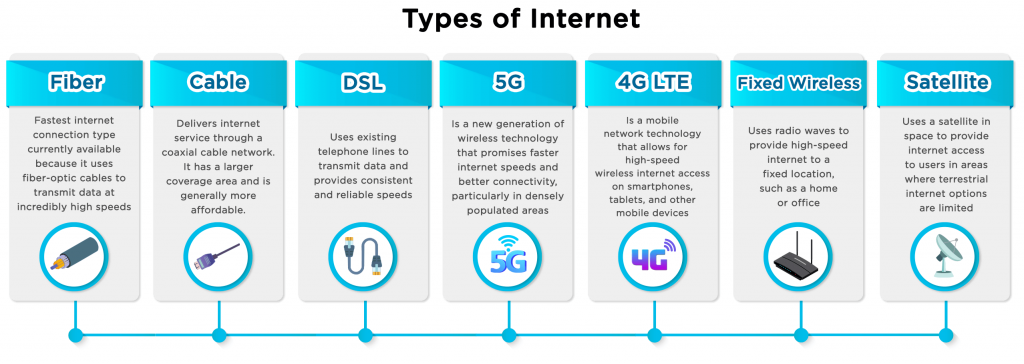
* Client Program:

1. Running on end host.
2. Request services.
3. Example: Web Browser.

* Server Program:

1. Running on end host.
2. Requests Services.
3. Example: Web Browser.
4. How does broadband differ from fiber-optic internet?

|  |  |  |
| --- | --- | --- |
| Feature | Broadband (General) | Fiber-Optic Internet |
| Definition | High-speed Internet. | Specific type of broadband using fiber. |
| Speed | Varies (depends on type: DSL, Cable). | Very fast. |
| Technology | Copper wires, coaxial cables, etc. | Light signals via glass fibers. |
| Reliability | Can suffer from interference. | Highly reliable, weather-resitant. |
| Availability | Widely | Still expanding in some areas. |

* **Lab Exercise:** Research different types of internet connections (e.g., broadband, fiber, satellite)and list their pros and cons.

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

|  |  |  |
| --- | --- | --- |
| Feature | HTTP | HTTPS |
| Full Form | Hyper Text Transfer Protocol. | Hyper Text Transfer Protocol Secure. |
| Security | Not Secure. | Secure. |
| Data Transfer | Plain Text (can be intercepted). | Encrypted (protected from hackers). |
| Port used | Port 80. | Port 443. |
| URL Prefix | http: // | https: // |
| Trust | Not trusted for sensitive data. | Trusted for login, payment, personal info. |
| Certificate | Dose not use SSL certificate. | Uses SSL/TSL certificate (HTTPS padlock) |

* **LAB EXERCISE**: Simulate HTTP and FTP requests using command line tools (e.g., curl).

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

* Encryption plays a critical role in securing applications by protecting data from unauthorized access. It ensures that even if someone intercepts the data, they cannot read or use it without the encryption key.

1. Data Protection.
2. Secure Communication.
3. Authentication.
4. Data Integrity.
5. Compliance.

* **LAB EXERCISE**: Identify and explain three common application security vulnerabilities. Suggest possible solutions.
* Three common application security vulnerabilities are:

1.SQL injection –

SQL injection exploits flaws in database queries, allowing attackers to manipulate data or gain unauthorized access.

* Solution-

Implement robust input validation and parameterization to prevent malicious SQL code from being injected into the application's queries. Use prepared statements or parameterized queries to treat user input as data, not code, and avoid executing SQL statements directly with user input.

2.Cross-Site Scripting (XSS) –

XSS vulnerabilities occur when an attacker injects malicious scripts (typically JavaScript) into a web page that another user will view. These scripts can then execute in the victim's browser, potentially stealing cookies, redirecting the user to malicious websites, or hijacking the user's session.

* Solution-

Use output encoding to escape any user-provided input before it's displayed in the web page. Employ a content security policy (CSP) to restrict the types of scripts that can be loaded on the web page.

3.Broken Authentication/Authorization-

Broken authentication vulnerabilities occur when an application's authentication mechanisms are flawed, allowing attackers to bypass access controls and gain unauthorized access. This can include weak password policies, inadequate session management, or the use of predictable login credentials.

* Solution-

Implement strong authentication mechanisms, such as multi-factor authentication (MFA) and secure password storage. Use secure session management techniques to prevent session hijacking, and regularly audit and update your authentication protocols.

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

* Application Software:

1. It is a most common type of software application.
2. It is a computer software package that perform a specific function for a user.
3. An application can be self-contained or it can be group of programs that run the application for user.
4. Example: Microsoft Office, Paint, Powerpoint etc.

* System Software:

1. Theses software programs are designed to run a computers application programs and hardware.
2. It coordinates the activities and functions of the hardware and software.
3. It controls the operations of the computers hardware and provides an environment or platform for all other type of software to work in.
4. The OS is the best example of this, it manages all other computer programs.
5. Example: Firmware, computer language translators and system utilities.

* **LAB EXERCISE**: Identify and classify 5 applications you use daily as either system software or application software.

**1.Operating System (e.g., Windows, macOS):-**

This is system software. Operating systems are essential for managing hardware and providing a foundation for other software to run.

2. Web Browser (e.g., Chrome, Firefox):-

This is application software. Web browsers are tools designed to access and navigate the internet, a user-oriented task.

3. Microsoft Word:-

This is application software. Word processors are designed to create, edit, and format text documents, a specific user task.

4.Adobe Photoshop:-

This is application software. Image editing software is used for tasks like photo manipulation and graphic design, which are specific user tasks.

5.Antivirus Software (e.g., McAfee, Norton):-

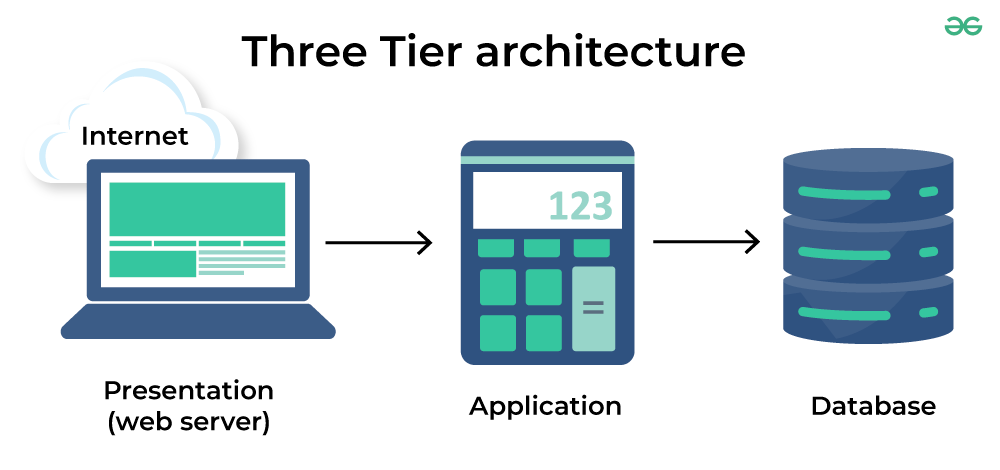
This is application software. Antivirus software is a specific application designed to protect the system from malware, a user-oriented task.

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

* The significance of modularity in software architecture is that it helps create software that is organized, flexible, scalable, and easier to manage.
* Modularity is important:

1. Improved Maintainability.
2. Reusability.
3. Scalability.
4. Parallel Development.
5. Testability.
6. Better Organization.

* **LAB EXERCISE:** Design a basic three-tier software architecture diagram for a web application.



1. Why are layers important in software architecture?

* Layers are crucial in software architecture because they help organize complex systems, making them easier to develop, manage and scale.
* Layers are important in Software Architecture:

1. Separation of Concerns.
2. Maintainability.
3. Reusability.
4. Modularity.
5. Scalability.
6. Testability.
7. Flexibility.

**LAB EXERCISE**: Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.

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

1. Application Software plays a vital role in business by helping organization perform specific tasks more efficiently and effectively.

1.Improves Productivity-

Software like word processors, spreadsheets, and presentation tools help employees create documents, analyze data, and communicate ideas quickly.

2.Streamlines Operations-

Business-specific applications automate and simplify routine processes, reducing human error and saving time.

3.Enhances Communication-

Tools like email clients, messaging apps, and video conferencing software support faster and more organized internal and external communication.

4.Supports Decision-Making-

Data analysis and business intelligence software provide insights that help leaders make informed strategic decisions.

5.Improved Customer Service-

CRM software helps manage customer interactions, track service issues, and enhance satisfaction.

6.Ensures Compliance-

Applications help businesses stay compliant with regulations by tracking data, managing documents, and generating necessary reports.

7.Enables E-commerce-

Software platforms allow business to sell products and service online, manage online payments, and handle logistics.

**Lab Exercise:** Write a report on the various types of application software and how they improve productivity.

1.Word processing Software-

* Example- Microsoft Word, Google Docs.
* Purpose- Used to create, edit, and format text documents.
* Productivity Benefits-
* Speeds up document creation.
* Offers templates for quick formatting.
* Enables real-time collaboration and editing.

2.Spreadsheet Software-

* Example- Microsoft Excel, Google Sheets.
* Purpose- Handles numerical data, calculations, and data analysis.
* Productivity Benefits-
* Automates complex calculations.
* Visualizes data through charts and graphs.
* Supports data sorting and filtering for quick analysis.

3.Presentation Software-

* Example- Microsoft PowerPoint, Canva, Google Slides.
* Purpose- Used to create slide-based presentations for meetings and communication.
* Productivity Benefits-
* Enchances the delivery of ideas and proposals.
* Offers pre-designed templates for quick slide creation.
* Supports multimedia integration for engaging presentations.

4.Database Management Software-

* Example- Microsoft Azure, Oracle, MySQL.
* Purpose- Organizes and manages large amount of structure data.
* Productivity Benefit-
* Stores and retrieves data efficiency.
* Reduces data duplication and errors.
* Allows for quick reporting and data queries.

5.Communication Software-

* Example- Microsoft Teams, Slack, Zoom.
* Purpose- Facilitates internal and external communication.
* Productivity Benefits-
* Enables instant messaging and video calls.
* Supports file sharing and team collaboration.
* Keeps teams connected regardless of location.

6.Project Management Software-

* Example- Trello, Asana, Microsoft Project.
* Purpose- Helps in planning, organizing, and tracking project progress.
* Productivity Benefit-
* Assigs tasks and deadlines to team members.
* Tracks time and resources.
* Keeps projects on schedule and budget.

7.Accounting Software-

* Examples- QuickBooks, Xero, FreshBooks.
* Purpose- Manages financial transactions and reporting.
* Productivity benefit-
* Automates billing, payroll, and invoicing.
* Reduces manual errors in accounting.
* Generates financial reports instantly.

8. CRM Customer Relationship Management) Software-

* Examples- Salesforce, HubSpot, Zoho CRM.
* Purpose- Manages customer interactions and sales data.
* Productivity Benefit-
* Centralises customer Information.
* Tracks leads and customer communications.
* Improves customer service and sales performance.

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

1. Software Development Process refers to a methodology with clearly defined process for creating high-quality software.

1. Planning:

This initial phase involves defining project scope, objectives, and resources needed for the development process.

2. Analysis (Requirements Gathering):

This stage focuses on understanding the specific needs and requirements of the users and stakeholders for the software.

3. Design:

In this phase, the software architecture, user interface, and overall design are planned and finalized.

4. Development (Coding):

This is where the actual coding and implementation of the software takes place.

5. Testing (Quality Assurance):

Thorough testing is conducted to ensure that the software functions as intended and meets the requirements.

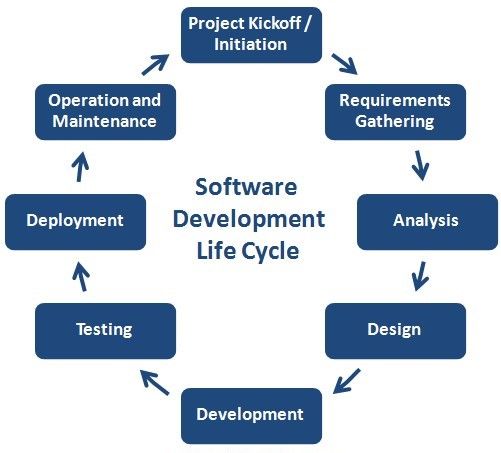
6. Implementation/Deployment:

This stage involves deploying the software to the target environment and making it accessible to users.

7. Maintenance:

This ongoing phase involves bug fixes, updates, and enhancements to keep the software running smoothly and meet evolving user needs.

**Lab Exercise**: Create a flowchart representing the Software Development Life Cycle.



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

* The requirement analysis phase is critical in software development because it ensures the final product meets user needs, reduces development time and costs, and prevents costly rework. By carefully identifying and documenting stakeholder needs, it provides a clear foundation for the entire development process, preventing misunderstandings and improving the overall project outcome.

1. Ensure right solution.
2. Reduces development time and costs.
3. Prevents misunderstanding and disagreements.
4. Identifies and mitigates risks.
5. Improves project success.
6. Optimizes resource allocation.
7. Facilitates testing.

**Lab exercise**: Write a requirement specification for a simple library management system.

1. Introduction:
2. Purpose-

* The purpose of this document is to outline the functional and non-functional requirements for a simple Library Management System that automates and manages the day-to-day operations of a library, such as managing books, members, and issuing/returning books.

1. Scope-

* This system will manage:
* Book cataloguing.
* Member registration.
* Book lending and returning.
* Fine calculation for overdue books.
* Search functionality for books and members.

1. Definitions, Acronyms, Abbreviations.

* LMS: Library Management System.
* ISBN: International Standard Book Number.
* UI: User Interface.

1. Overall Description:
2. Product Perspective-

* The system is a standalone desktop/web-based application for small libraries, such as those in schools or colleges.

1. User Classes and Characteristics-

* Admin: Manages books and users. Full access.
* Librarian: Manages book issues/returns and fines.
* Member/User: Can search books, view personal borrowing history.

1. Operating Environment-

* Web Browser/Desktop application.
* Database: MySQL or PostgreSQL.
* Backend: Python/Java/Node.js.
* Frontend: HTML/CSS/JavaScript.

1. Functional Requirements:
2. Book Management:;

* Add, edit, delete books.
* View book details.
* Check availability status.

1. Member Management:

* Register, update, delete members.
* Assign members IDs.
* View member details.

1. Book Lending/Returning:

* Issue books to members.
* Record return of books.
* Calculate and record fines for overdue returns.

1. Search Functionality:

* Search books by title, author, ISBN, or category.
* Search members by name or ID.

1. Reports:

* Generates report.
* Overdue books list.
* Member borrowing history.

1. Non-Functional Requirements:
2. Performance:

* The system should support at least 1000 concurrent users.
* Search operations should respond within 2 second.

1. Usability:

* User-friendly interface with clear navigation.
* Minimal training required for users.

1. Security:

* Role-based access control.
* Data encryption for sensitive information.

1. Reliability:

* The system should have 99.9% uptime.
* Daily data backups should be performed.

1. Assumption and Constraints:

* Each member can borrow a maximum of 5 books at a time.
* Books can be borrowed for a maximum of 14 days.
* Fines are calculated at $0.50 per day overdue.

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

* The role of software analysis in the development process is crucial, as it acts as the bridge between the problem and the solution. It involves gathering, understanding, and documenting the needs and requirements of a software system, ensuring the final product meets user expectations and business goals. This phase helps define what the software should do, how it should function, and the constraints it operates under.

1. Understanding the problem:

* It helps in fully understanding the problem the software is intended to solve.
* It involves gathering detailed information from stakeholders to ensure their needs are understood correctly.

1. Defining Requirements:

* Clearly defines functional and non-functional requirements.
* Helps avoid ambiguity, ensuring all team members and stakeholders are on the same page.

1. Establishing Scope and Constraints:

* Sets the boundaries of what the software will and will not do.
* Identifies technical, financial, and timeline constraints.

**Lab Exercise**: Perform a functional analysis for an online shopping system.

* A functional analysis for an online shopping system involves identifying and documenting the specific features and functionalities needed for the system to operate effectively. This includes defining how users interact with the system, what tasks they can perform, and how the system manages products, orders, payments, and other essential aspects. The goal is to ensure the system meets user needs and provides a seamless shopping experience.

1. User Roles:
2. Customers-

* Browse products.
* Search and filter products.
* View product details.
* Add/remove products to/from cart.
* Place orders.
* Track orders.
* Manage profile and shipping address.
* Review and rate products.

1. Administrators-

* Manage user accounts.
* Add/edit/delete products.
* Manage product categories.
* View sales and user activity reports.
* Process and update order status.
* Manages discounts/promotions.

1. Sellers-

* List new products.
* Manage inventory.
* View orders and fulfil them.
* Communicate with customers.
* View sales analytics.

1. Functional Requirements:
2. User Registration and Authentication-

* Sign up/login via email, phone, or social accounts.
* Password recovery/reset.
* Role-based access control (admin, customer, seller).

1. Product Catalog Management-

* Product categorization and tagging.
* Multiple images per product.
* Pricing, discounts, and stock information.
* Product descriptions and specification.

1. Search and Navigation-

* Keyword search.
* Filters (price, brand, rating, etc.)
* Sorting (newest, price, popularity)
* Category-wise navigation.

1. Shopping Cart and Checkout-

* Add/remove/update items in cart.
* Apply discount coupons.
* View cart summary and total.
* Secure checkout process.

1. Payment Processing-

* Integration with payment gateways (e.g., PayPal, Stripe).
* Support for credit/debit cards, net banking, wallets.
* Payment success/failure handling.

1. Order Management-

* Order Confirmation with tracking.
* Order history.
* Cancellation and returns.
* Notification via email/SMS.

1. Revie and Rating System-

* Verified purchases can leave reviews.
* Star ratings and written feedback.
* Moderation for offensive content.

1. Notification-

* Order and payment status.
* Promotional messages.
* System alerts and messages.

16. What are the key elements of system design?