**Module 1 – Overview of IT Industry**

1. What is a program?

Ans: A program is a set of instructions written in a specific programming language that’s tells computer what to do and program is a collection of code that performs a specific task when executed by a computer.

1. Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.

Ans.

#include<stdio.h>

Main() {

Printf(“hello world”);

}

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

Ans. A program is written by a programmer using a programming language (like, C, Java, Python) that tells a computer what task to perform and how to do it. A program is like a step-by-step guide for the computer.

* Programming is the process of writing instructions (called code) that a computer can understand and execute to perform specific tasks. Programming helps us create software and application we use every day

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

Ans. Programming is a step-by-step process that involves creating, testing, and running computer programs. Below are the main steps involved:

* Writing the code
* Writing the actual code using a programming language like C, Java, python and follow the logic and structure.
* Testing and Debugging
* Test the program to check if it works correctly and if there are errors (called bugs), fix them. This process know as debugging.
* Execution
* Run the program and observe the output. Make sure it performs the required tasks as expected.

Types of programming:

1. Procedural Programming: Procedural programming language that organizes code into a series instruction, or procedures, that are executed in a specific order. Based on steps (procedures or functions). The program is written as a sequence of instructions.

Example: C language

1. Object-Oriented Programming (OOP): Is a programming paradigm that uses “objects” as the fundamental building blocks. Objects can contain data and have actions they can perform. In OOP, computer programs are designed by making them out of objects that interact with one another.

Example: Java, C++

1. Logical Programming: logical programming is a programming, database on formal logic. A logic program is a set of sentences in logical form, representing knowledge about some problem domain. It is used in artificial intelligence, natural language processing, database management and etc.

Example: prolog language

1. Functional programming: In computer science, functional programming is a programming paradigm where programs are constructed by applying and composing functions.

Example: python language

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

Ans.

Programming languages can be broadly categorised into two types:

1. High-level languages: these languages like python and java that are closer to human language, making them easier to code and debug.

* A high-level language is human readable programming language that simplifies coding by hiding complex hardware details, letting developers focus on logic and functionality.
* Designed to make writing code simpler and faster.
* Allow developers to build large programs more easily.
* Easier to find and fix mistakes.
* Good for beginners and widely used for everyday software.
* Examples include JavaScript, Ruby, Swift and PHP.

1. Low-level languages: A low-level language is a machine-oriented programming language that provides minimal abstraction from hardware offering direct control over memory and system resources for maximum performance and efficiency.

* Provide direct access to the computer ‘s hardware.
* Required detailed knowledge of how a computer works.
* Less user-friendly, making.

1. Word Wide Web and how Internet works.

Ans. The Word Wide Web is a system of interconnected document and others resources, accessed using the internet, while the internet is a vast global network of interconnected computer network.

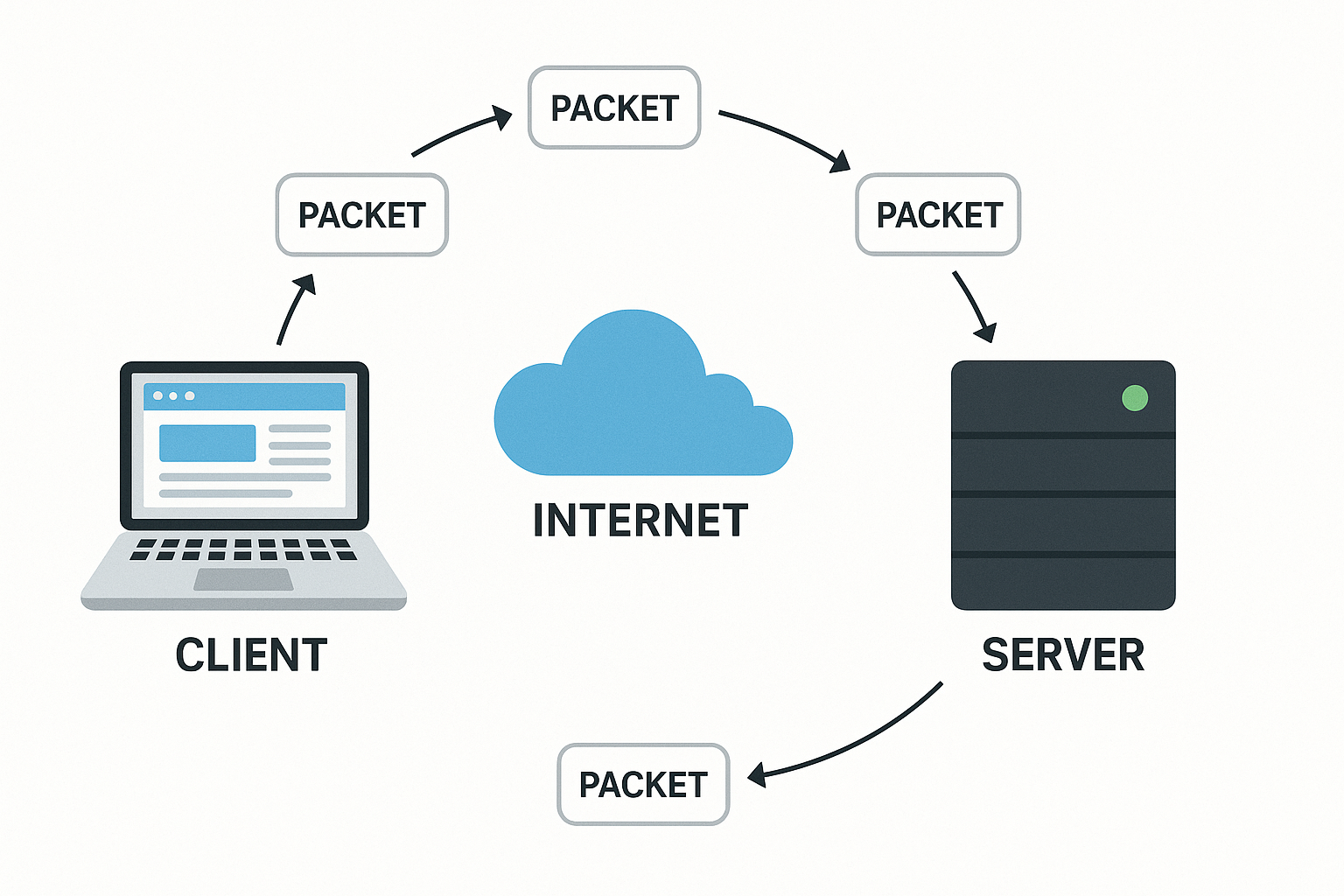
* The WWW relies on the internet for its functionality. Essentially, the internet is the infrastructure, and the WWW is a service built on top of that infrastructure.

1. Research and create a diagram of how data is transmitted from a client to a server over the internet.

Ans. **Step-by-Step Process of Data Transmission (Client to Server)**

1. **User Sends Request (Client Side)**:  
   The client (e.g., browser or app) sends a request for a web page (e.g., www.example.com).
2. **DNS Resolution**:  
   The domain name is converted to an IP address using the **DNS server**.
3. **Data is Packetized**:  
   The data is broken into small pieces called **packets**.
4. **Packets Go Through Layers**:  
   The data moves through the **TCP/IP model layers**:
   * Application Layer
   * Transport Layer (adds TCP header)
   * Internet Layer (adds IP header)
   * Network Access Layer (adds MAC info)
5. **Router & ISP Transmission**:  
   The data packets travel via the user's **router** and **ISP** to the Internet.
6. **Routing Over the Internet**:  
   The packets pass through multiple **routers** across the Internet to reach the server.
7. **Server Receives the Request**:  
   The server reassembles the packets and sends back a response in the same layered process.
8. Research and create a diagram of how data is transmitted from a client to a server over the internet.

Ans. Data Transmission from Client to Server



Data transmission from a client server over the internet follows the client server model. A client, typically a web browser or email client, initiates a request to a server, which then processes the request and sends back a response. This communication relies on networking protocols like HTTP/HTTPs, TCP/IP, or WebSocket.

1. Client Initiates Request: the client (e.g., a web browser) sends a request the server, such as asking for a webpage or uploading a file.
2. Network Requests: The client packages the request and sends it over the network, utilizing protocols like TCP/IP.
3. Internet Transmission: The network requests traverser the internet passing through routers and other network request and processes it.
4. Server Receives Request: The server receives the network request and processes it.
5. Server Response: The server performs the necessary actions (e.g., retrieving data, running a program) and sends a response back to the client.
6. Response Transmission: The Server’s response travels back through the internet, utilizing the same or similar protocols.
7. Client Receives Response: The client receives the response from the server, which it then displays or processes.
8. : Describe the roles of the client and server in web communication. Network Layers on Client and Server.
9. Describe the roles of the client and server in web communication. Network Layers on Client and Server.

Ans. In web communications, the client roles within the network layers. The client, typically a web browser or application, initiates requests for resources or services from the server.

* The server, in turn, processes these requests and sends back the requested data or performs the requested action.
* This interaction is governed by protocols like HTTP and relies on network infrastructure for transmission.

1. Design a simple HTTP client-server communication in any language.

Ans. # simple\_server.py

from http.server import BaseHTTPRequestHandler, HTTPServer

class SimpleHandler(BaseHTTPRequestHandler):

def do\_GET(self):

# Send response status code

self.send\_response(200)

# Send headers

self.send\_header('Content-type', 'text/html')

self.end\_headers()

# Send response body

message = "<html><body><h1>Hello from Server!</h1></body></html>"

self.wfile.write(message.encode())

# Server settings

host = "localhost"

port = 8080

server = HTTPServer((host, port), SimpleHandler)

print(f"Server started at http://{host}:{port}")

server.serve\_forever()

Run this file using: python simple\_server.py

# simple\_client.py

import requests

response = requests.get('http://localhost:8080')

print("Status Code:", response.status\_code)

print("Response Body:\n", response.text)

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

Ans. The TCP/IP model (Transmission control protocol/ internet protocol) is a four-layer networking framework that enables reliable communication between devices over interconnected networks.

* It provides a standardized set of protocols for transmitting data across interconnected networks, ensuring efficient for understanding and working with modem networks.
* TCP/IP enables interoperability between diverse system over various network types (e.g., copper, fibre, wireless). It ensures seamless communication across LANs, WANs, and the internet. Without TCP/IP, large-scale global networking would not be possible.

Layers of TCP/IP Model

* It’s composed of four interconnected layers compared to the seven in the OSI model. Each layer performs a specific task on the data that is being transmitted over the network channel, and data moves from one layer to another.

1. Application layer
2. Transport layer (TCP/UDP)
3. Network/Internet Layer (IP)
4. Network Access Layer
5. Client and Server

Ans.

Client

* When we talk about a “client”, it refers to a device (usually a computer, smartphone, or application) that requests and receives services from a server.
* The client is the entity that initiates communication, asking for data or resources from the server. For instance, web browser like google chrome, Mozilla Firefox, or Safari are common client applications that request data from a server to render web pages.

Server

A Server, on the other hand, is a remote computer or system that provides data, resources, or services to clients. It listens to incoming client requests, processes them, and sends the required information back. A server can handle multiple client requests simultaneously.

1. Explain Client Server Communication

Ans. In an Operating System, Client Server Communication refers to the exchange of data and services among multiple machines or processes. In client client-server communication system one process or machine or process acts like a server for providing those services or data to the client machine.

Different Ways of Client-Server Communication

1. Sockets Mechanism
2. Remote Procedure Call
3. Message passing
4. Inter-Process Communication
5. Distributed File Systems
6. Types of Internet Connections

Ans. An Internet connection is a means by which individual devices or local networks are linked to the global internet, allowing them to communicate and exchange data.

* There are many connections that can be used for internet access. All the connections have their own speed range that can be used for different purposes like for home, or for personal use.

Types of Internet Connections

1. Dial-Up Connections
2. Broadband Connections
3. DSL (Digital Subscriber Line)
4. Cable
5. Satellite Connections
6. Wireless Connection
7. Cellular
8. Research different types of internet connections (e.g., broadband, fibre, satellite) and list their pros and cons.
9. Dial-Up Connections

A dial-up Connections is established between your computer and the ISP server using a modem. A dial-up Connection is a cheap and traditional connection that is not connection is very slow.

Pros: fast speeds, widely available, often bundled with other services.

Cons: Speeds can be affected by network congestion, can be slower than fibre.

1. Broadband Connection

Broadband refers to high-speed internet access that is faster than traditional dial-up access. It is provided through either cable or telephone composition. It does not require any telephone connection that’s why here we can use telephone and internet connection simultaneously.

1. DSL (Digital Subscriber Line)

It provides an internet connection through the telephone line(network). DSL is a from of broadband communication that is always on, there is no need to dial a phone number to connect.

Pros: Relatively inexpensive, widely available, uses existing telephone lines.

Cons: Slower speeds than fiber and cable, speeds can be affected by distance from the provider’s equipment activities.

1. Cable

It is a from of broadband access cable modem that can provide extremely fast access to the internet. The speed of this connection varies which can be different for uploading data transmission or downloading.

1. Satellite Connection

This type of connection is provided mainly in rural areas where a broadband connection is not yet offered. It accesses the internet via a satellite that is in earth’s orbit. The signal travels from a long distance that is from earth to satellite and back again which provides a delayed connection.

Pons: Available in remote areas where other connections are not available.

Cons: High latency (delay), speeds can be affected by weather conditions, can be expensive

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

Ans. Broadband is general term for high-speed internet access, while fiber-optic internet is a specific type of broadband technology that uses light to transmit data through thin glass or plastic fibers.

* Fiber-optic internet is generally faster and more reliable than other forms of broadband, such as DSL or cable, due to its use of light signals and less susceptibility to interference.

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

Ans. HTTPS is just HTTP with encryption. The primary distinction between these two names is that HTTPS is more secure than HTTP requests and responses, even the standard ones.

|  |  |
| --- | --- |
| HTTP | HTTPS |
| HTTP stands for Hyper Text Transfer Protocol. | HTTPS for Hyper Text 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 work at application layer. | HTTP works at transport layer. |

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

Ans. Data encryption is the process of converting readable information (plaintext) into an unreadable format (ciphertext) to protect it from unauthorized access.

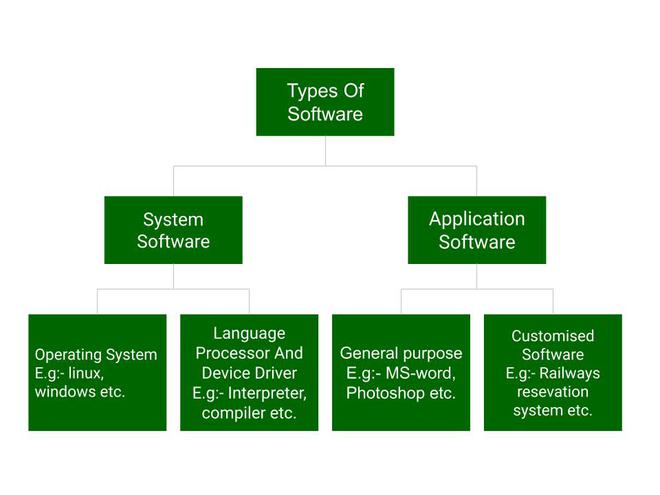
* It is a method of preserving data confidentiality by transforming it into ciphertext, which can only be decoded using a unique decryption key produced at the time of the encryption or before it.
* The conversion of plaintext into ciphertext is known as encryption. By using encryption keys and mathematical algorithms, the data is scrambled so that anyone intercepting it without the proper key cannot understand the contents.

1. Software Applications and Its Types.

Ans. In a computer system, the software is basically a set of instructions or commands that tell a computer what to do. In other words, the software is a computer program that provides a set of instructions to execute a user’s commands and tell the computer what to do. For example, like MS-Word, MS-Excel, PowerPoint, etc.

Types of Software

It is a collection of data that is given to the computer to complete a particular task.



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

Ans. In the era of Digitalization and Modernization, Software is the very crucial support that allows hardware to perform various useful tasks. There are two categories of software; System Software and Application Software, these types perform different work which is why it is crucial to understand the difference.

* Application software is created to help users to perform specific tasks directly and System software acts as a mediator between hardware and user applications.
* Computer Software is a sort of program that allows clients to work on different assignments or use them to work on their System. It tells the working and responsibilities of the System.
* Basically, Software is a set of instructions or commands that tells a user how to do and what to do. In this article, we will look into these topics in detail along with their differences.

1. Software Architecture

Ans.

Software Architecture defines fundamental organization of a system and more simply defines a structured solution. It determines how the various components of a software system are assembled, how they relate to one another, and how they communicate. Essentially, it serves as a blueprint for the application and a foundation for the development team to build upon.

Software architecture defines a list of things which results in making many things easier in the software development process.

* 1. System structure: The organization and arrangement of components.
  2. System behaviour: The expected functionality and performance.
  3. Component relationships: How different parts of the system interact.
  4. Communication structure: The way components communicate with each other.
  5. Stakeholder balance: Meeting the needs and expectations of all stakeholders.
  6. Team structure: How the development team is organized and coordinated.
  7. Early design decisions: Making important choices early on to guide development.

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

Ans. The module simply means the software components that are been created by dividing the software.

* The software is divided into various components that work together to form a single functioning item but sometimes they can perform as a complete function if not connected with each other.
* This process of creating software modules is known as Modularity in software engineering. It simply measures the degree to which these components are made up than can be combined. Some of the projects or software designs are very complex that it's not easy to understand its working and functioning.
* In such cases, modularity is a key weapon that helps in reducing the complexity of such software or projects. The basic principle of Modularity is that “Systems should be built from cohesive, loosely coupled components (modules)” which means s system should be made up of different components that are united and work together in an efficient way and such components have a well-defined function.

1. Layers in software Architecture

Ans. Software architecture layers are levels in a software system, where each layer has a specific responsibility. The system is organized in such a way that each layer handles a specific part of the application’s functionality.

4-Layer Architecture:

1. Presentation Layer (UI Layer)
2. Business Logic Layer (BLL)
3. Data Access Layer (DAL)
4. Data Layer (Database)
5. Why are layers important in software architecture?

Ans. Layers are crucial in software architecture because they enable modularity, improve maintainability, and enhance scalability and security. By dividing a system into distinct layers, each responsible for specific functionalities, developers can manage complexity isolate changes, and easily adapt to evolving.

1. Modularity and Separation of Concerns:

Layered architecture promotes a "divide and conquer" approach, breaking down a complex system into smaller, manageable layers.

Each layer focuses on a specific set of functionalities, leading to a clear separation of concerns.

1. Improved Maintainability:

Changes in one layer are less likely to impact other layers, thanks to the separation of concerns.

This isolation simplifies bug fixing and feature additions, reducing the risk of unintended consequences.

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

Ans. A development environment is crucial in a software production because it provides a safe and controlled space for developers to build, test, and refine software without affecting live users or system.

* It streamlines workflows, enhances quality, and enables efficient management of the software development lifecycle.
  1. Isolation and safety:
* The development environment acts as a sandbox, allowing developers to experiment with code, introduce new features, and fix bugs without impacting the live application process.
  1. Enhanced Quality and efficiency:
* By providing a dedicated space for testing, the development environment helps ensure that the software functions as intended before release.

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

Ans. Source code is human-readable text written in a programming language, while machine code is the low-level, binary code needs to be translated into machine code (by a compiler or interpreter) before a computer can understand and run it.

|  |  |
| --- | --- |
| Source code | Machine code |
| Created by the programmer | Created by the compiler |
| Text rich document | Binary digits make up the object code. |
| Human Readable | Machine Readable |
| Can be changed over time | Needs to complier the source code each time a change is to be made. |
| Not System specific | System specific |

1. Why is version control important in software?

Ans. Version control is crucial in software development for managing changes to code and other digital as sets, ensuring collaboration, and enabling efficient development workflows.

* It acts as a safety net, allowing developers to track, revert, and compare code changes, and it also helps prevent conflicts and maintain a single source of truth.
* Version Control Systems (VCS) are essential tools used in software development and collaborative projects to track and manage changes to code, documents, and other files.
* Whether you're working alone or as part of a team, version control helps ensure that your work is safe, organized, and easy to collaborate on.

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

Ans. GitHub offers significant benefits for students, including free access to professional tools, opportunities for collaboration and project management, and a platform for showcasing work and building a portfolio.

* The GitHub Student Developer Pack provides students with free access to tools like GitHub Pro, cloud services, and developer tools, which can be incredibly valuable for their academic and professional.

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

Ans.

|  |  |
| --- | --- |
| Open-Source Software | Proprietary Software |
| Open-Source is computer software whose source code is available openly on the internet and programmers can modify it to add new features and capabilities without any cost. | Proprietary Software is computer software where the source codes are publicly not available only the company which has created can modify it. |
| Here the software is developed and tested through open collaboration. | Here the software is developed and tested by the individual or organization by which it is owned not by the public. |
| In open-source software the source code is public. | In Proprietary software, the source code is protected. |
| Open-source software can be installed on any computer. | Proprietary software can not be installed into any computer without a valid license. |
| Users do not need to have any authenticated license to use this software. | Users need to have valid and authenticated license to use this software. |
| Usually developed and maintained by non-profit organizations | Usually developed and maintained by for-profit entities. |
| Examples are Android, Linux, Firefox, Open Office, GIMP, VLC, Media player | Examples are windows, macOS, Internet, Explorer, Flash Player |

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

Ans. Git improves collaboration in software development by providing a robust version control system with features like branching, merging, and pull requests.

* These features enable developers to work on different parts of a project concurrently, integrate changes efficiently, and review each other’s work, minimizing conflicts and maximizing productivity.
* When multiple developers are working on the same project, Git collaboration plays an important role in providing easy coordination and efficient code management.

1. Application Software

Ans. The term “Application Software” refers to software that performs specific functions foe a user. When a user interacts directly with a piece of software, it is called application software.

* The sole purpose of application software is to assist the user in doing specified tasks. Microsoft word and Excel, as well as popular web browsers like Firefox and google chrome, are examples of application software.

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

Ans. Many applications software is used in Business. Some is used in Business.

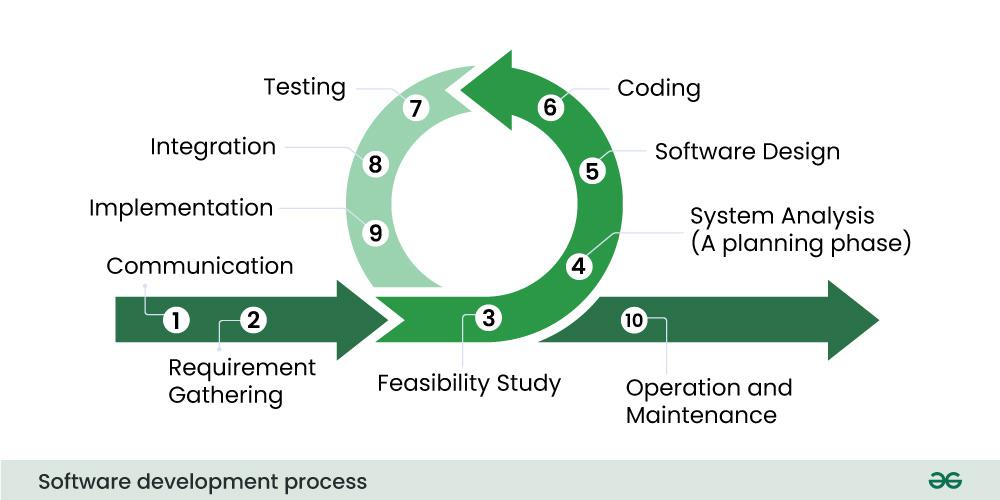
* Customer Relationship Management (CRM) : CRM is a type of technology that can manage the customer, transaction of customer, future transactions, etc. It is very important nowadays. It helps in expanding business to the next level as it stays connected with customers, keeping more revenues and less tensions.
* Enterprise Resource Planning (ERP): ERP is a type of software that handles some basics parts of any operation, resource management. Etc.
* Database: DBMS (Database Management System) is a way to keep in an automatic system. Here, various types of operations can also be performed in the database.

1. Software Development Process

Ans. The software development process is the approach to developing, and delivering, and delivering software applications. This process might include improving design and product management by splitting the work into smaller steps or processes.

* This process might include improving design and product management by splitting the work into smaller steps or processes.

What are the 10 software Development Processes?



1. Communication

The first and foremost step is where the user contacts the service provider software organization and initiates the request for a desired software product.

1. Requirement gathering

In this step, the team of software developers holds discussion with various stakeholders from the problem domain and provides as much as information possible for the requirement of the software product.

1. Feasibility Study

After requirement gathering, with the help of many algorithms, the team analyzes that if the software can be designed to fulfil all requirements of the user and also analyzes if the project is financially, practically and technologically feasible for the organization or not.

1. System Analysis (A planning phase)

Software developer decides on a roadmap for their plan and tries to bring up the best software model stable for the project. System analysis may also include understanding product limitations and identifying and addressing the impact of the project on the organization. The project analyzes the scope of the project and plans the resources accordingly.

1. Software Design

Software design whole knowledge of requirements and analyses are taken together to plan up design of software products. It takes input from the user and information gathered in the requirement-gathering phase. It gives output in the form of logical and physical design.

1. Coding

This step is also known as the programming phase. The implementation of software design starts in the form of writing code in suitable programming and developing error-free programs efficiently.

1. Testing

Software testing is done while done while coding by the tester’ developing team members. Testing is done at various levels module testing, product testing, program testing and user-end testing.

1. Integration

After writing all the codes for the software such as frontend, backend and databases, the software is integrated with libraries, databases and other programs.

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

Ans. The software development process, also Known as the Software development life Cycle (SDLC), Typically involves several key stages.

* Those include Planning and requirement gathering analysis, design, development (or coding), testing, deployment, and maintenance.

1. Software Requirement

Ans. Classification of software Requirement is important in the software development process. It organizes our requirements into different categories that make them easier to manage, prioritize, and track. The main types of software Requirements are functional, non-functional, and domain requirements.

1. Functional Requirements: Functional requirements describe what the software should do. They define the functions or feature that the system must have.
2. Non-Functional Requirements: Describe how the software performs a task rather the quality attributes, performance criteria and constraints.
3. Domain Requirements: Domain requirements are specific to the domain or industry in which the software operates. They include terminology relevant to that particular domain.
4. Why is the requirement analysis phase critical in software development?

Ans. The requirement analysis phase is crucial in software development because it lays the foundation for a successful project. It ensures that the development team understands exactly what needs to be built and how, minimizing misunderstanding and costly rework later on. By thoroughly gathering and analyzing requirements upfront, the team can create software that meets the needs and expectations of all stakeholders.

1. System Design

Ans. System design is the process of defining the architecture, components, modules, interface, and data for a system to satisfy specified requirements. It involves translating user requirements into a detailed blueprint that guides the implementation phase.

The goal is to create a well-organized and efficient structure that meets the intended purpose while considering factors like scalability, maintainability, and performance.

1. What are the key elements of system design?

Ans. Key elements of system design include architecture, data flow, scalability, reliability, security, performance, maintainability, APIs and interfaces, and database design.

These elements ensure a robust, efficient, and adaptable system.

Here's a breakdown of these key elements:

1. Architecture: This encompasses the overall structure of the system, defining how different components interact and work together. It involves choosing architectural patterns (e.g., microservices, monolithic) based on project needs.

1. Data Flow: Understanding how data enters, moves through, and is stored within the system is crucial for performance and efficiency. This includes designing data ingestion, processing, storage, and retrieval mechanisms.

3. Scalability: The ability of the system to handle increased load (users, data, traffic) without significant performance degradation. This often involves both vertical scaling (increasing resources on existing machines) and horizontal scaling (adding more machines).

1. Software Testig

Ans. Software Testing is an process in the Software Development Lifecycle (SDLC). It involves verifying and validating that a software application is free of bugs, meets the technical requirements set by its design and development, and satisfies user requirements efficiently and effectively.

* Software Testing is a process of verifying and validating whether the software product or application is working as expected or not.
* The complete testing includes identifying errors and bugs that cause future problems for the performance of an application.

1. Why is software testing importance and Maintenance?

Ans. Software testing is crucial for ensuring a quality product, reducing costs, and increasing user satisfaction. Maintenance, on the other hand, is equally important as it keeps the software functional, secure, and efficient over time by addressing issues and adapting to changing needs. Together, testing and maintenance contribute to a reliable and valuable software product.

Importance of Software Testing:

1. Early Defect Detection:

Testing identifies bugs and errors early in the development process, preventing them from becoming more costly and time-consuming to fix later.

1. Improved Quality:

Thorough testing ensures the software functions correctly, meets user expectations, and is free from critical bugs, leading to a higher quality product.

1. Enhanced User Experience:

A well-tested software provides a better user experience, leading to increased customer satisfaction and trust.

1. Cost Reduction:

Finding and fixing issues early in the development cycle can significantly reduce the overall cost of development and maintenance.

1. Security:

Testing helps identify and address security vulnerabilities, protecting sensitive data and preventing potential breaches.

1. What types of software maintenance are there?

Ans. There are four main types of software maintenance:

Corrective, adaptive, perfective, and preventive.

These categories address different aspects of keeping software functional and up-to-date throughout its lifecycle.

* Software maintenance refers to the process of modifying and updating a software system it has been delivered to the customer.
* This involves fixing bugs, adding new features, and adapting to new hardware or software environments. Effective maintenance is crucial for extending the software’s lifespan and aligning it with evolving user needs.
* It is an essential part of the software development life cycle (SDLC).

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

Ans. The critical difference between web and desktop applications is where the application logic executes.

* Web applications run application logic on remote servers accessed over the internet or local network.
* The web browser is the client, displaying information and providing the user interface, the web server stores the application data and handles the business the business logic.
* In contrast, desktop application run natively on the user’s device. All application logic and data are processed and stored locally, allowing desktop apps to work offline. The desktop app interface runs on the local device without relying on an internet connection to remote server.

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

Ans. Data Flow Diagrams (DFDs) are crucial in system analysis because they visually represent how data flows through a system aiding in understanding, communication, and improvement. They break down complex processes into manageable components, making it easier to analyze and design systems, and facilitate collaboration among stakeholders.

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

Ans. Web apps are accessible from anywhere with an internet connection, while desktop apps must be installed on each user’s device. Desktop apps perform better, work offline, and keep data more secure. Yet web allows real-time collaboration and avoid version update issues.

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

Ans. Flowcharts are visual tools that significantly aid in both programming and system design. By providing a clear, step-by-step representation of logic and processes. They enhance understanding, communication, and efficiency throughout the development lifecycle.

* Computer programmers use flow charts to show where data enters the programs, what processes the data goes through, and how the data is converted to output.