

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

Ans;A program is instructions for a computer to execute specific tasks. It contains code written in a programming language which may be interpreted, compiled or assembled into machine readable form and then executed.

2.What is Programming?

Ans;Programming is the process of writing instructions for a computer to follow, or code, in order to solve problems. It's a collaboration between humans and computers.

A programmer breaks down a problem into a series of logical steps, called an algorithm.

The programmer encodes the algorithm into a programming language that a computer can understand.

The computer executes the code to perform the tasks.

3.What are the key steps involved in the programming process?

Ans.The key steps in the programming process are analyzing the problem, designing the program, coding, testing, and maintaining.

Problem analysis

Review the program specifications to determine what the system should do

Clarify the needs and objectives of the program

Determine the inputs and outputs

Design

Choose algorithms, data structures, and interfaces

Design the program logic and flow

Consider trade-offs between performance, memory usage, and other factors

Coding

Select a coding language and follow its syntax rules

Write the program in the selected language

Testing

Test the program for errors

Ensure the software meets quality standards and functions as intended

Conduct unit tests, integration tests, system tests, and user acceptance tests

Maintenance

Ensure the software continues to meet changing business needs

Gather requirements, design, develop, test, and deploy changes

Documentation

Create documentation that includes user guides, code comments, and system architecture diagrams

Document for users, operators, and future programmers

#### 4.Types of Programming Languages?

Ans;There are several types of programming languages, including machine language, assembly language, high-level languages, functional languages, procedural languages, logic languages, and more.

Low-level and high-level languages

Machine language: The computer's processing unit directly understands machine language.

Assembly language: A list of basic instructions that's harder to read than high-level languages.

High-level languages: Designed to be easy to read and understand.

#### 5.What are the main differences between high-level and low-level Programminglanguages?

Ans;Low-level languages require more coding and debugging, which increases development time. High-level languages have a more natural and readable syntax, which makes it easier for programmers to read and understand the code. Low-level languages have a more cryptic syntax that is difficult to read and understand

The choice between high-level and low-level languages depends on project requirements. Many developers and projects benefit from a combination of both.

#### 6.World Wide Web & How Internet Works

Ans;The World Wide Web (WWW), often simply called "the Web," is a system of interconnected web pages accessible through the internet, where users can navigate

between different websites using hyperlinks, essentially a network of information accessed through web browsers, while the internet itself is a vast network of interconnected computer networks that allows data to be transferred between devices globally; meaning the Web is a specific application that runs on top of the broader internet infrastructure.

7. Describe the roles of the client and server in web communication.

Ans; In web communication, a “client” is the device or application that initiates a request for information or services from a “server,” which is a computer system that stores and processes the requested data, then sends the response back to the client; essentially, the client asks for something, and the server provides it, following a request-response model.

8. Network Layers on Client and Server

Ans; In a network communication between a client and server, both devices utilize all layers of the OSI model, including the Physical, Data Link, Network, Transport, Session, Presentation, and Application layers, performing the same functions on each side, but interacting with each other based on the client’s request and the server’s response; essentially, the client initiates communication by sending data down the layers to the network, while the server receives data and sends back a response through the same layers.

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

Ans; Menu

TCP/IP Model

TCP/IP model is the practical model that every networking equipment manufacturer follows today whether it is Cisco, Huawei, Nokia, Ericsson or Juniper. TCP/IP Model is a standard reference Model for conceptualizing data communications between networks.

The TCP/IP is a practical model that provides end to end data communication specifying how data should be encapsulated, routed and finally received at destination. It characterizes and standardizes the communication functions of a network system without regard to its underlying internal structure and technology. It is also called the ‘Internet Model’ or the ‘DoD Model’. TCP/IP is a Vendor Neutral Model

10. Client and Servers

Ans; In a client-server model, a client is a device or software that requests services or information from a server, which is a device or software that provides those services or information.

How do clients and servers communicate?

Requests

Clients send requests to servers for services or information. For example, a user might request a file or log in to a system.

Responses

Servers respond to client requests with data or status information. For example, a server might respond to a login request with an authentication result.

11.Explain Client Server Communication

Ans; Client-server communication is a model where a client sends a request to a server, and the server responds with data or services. This model is used in many applications, including cloud computing, web browsing, and mobile apps.

How it works

A client, like a web browser or mobile app, sends a request to a server.

The server processes the request and sends back the requested data or services.

The client receives the data or services.

What it's used for

Cloud computing

The client-server model is the foundation of cloud computing. It allows developers to scale infrastructure and deploy applications close to end users.

Web browsing

When a user requests a webpage, the browser acts as the client and sends a request to the web server. The server responds with the requested data, which allows the user to access the webpage.

12 .Types of Internet Connections

Ans; There are several types of internet connections, including dial-up, cable, fiber optic, satellite, and wireless.

## Wired connections

DSL: A common broadband connection that uses copper wires

Cable: A common broadband connection that uses cables

Fiber optic: A fast and reliable broadband connection that uses fiber-optic cables

## Wireless connections

Wi-Fi: A wireless broadband connection that uses Wi-Fi networks

Wi-MAX: A wireless broadband connection that uses Wi-MAX networks

5G: A wireless broadband connection that uses millimeter wave (mmWave) bands

## Satellite connections

Satellite broadband: An internet connection that uses satellites

Satellite internet: An internet connection that uses satellites

## Mobile connections

Mobile broadband: An internet connection that uses mobile devices

Cellular network: An internet connection that uses cellular networks

When choosing an internet connection, you can consider things like speed, reliability, latency, and cost.

## 13.How does broadband differ from fiber-optic internet ? protocol

Ans; Broadband is a general term for high-speed internet connections using various technologies like cable, DSL, or satellite, while fiber-optic internet specifically refers to a much faster internet connection that transmits data using light signals through glass fibers, making it significantly quicker and more reliable than traditional broadband options like cable or DSL; essentially, fiber-optic is a specific type of broadband technology that utilizes superior transmission methods through fiber optic cables.

Key differences:

Transmission method:

Broadband uses electrical signals through copper wires (like in cable or DSL), while fiber-optic uses light signals through glass fibers.

Speed:

Fiber-optic internet is considerably faster than most traditional broadband options.

Reliability:

Fiber-optic connections are generally more reliable due to less interference and signal degradation.

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

Ans; HTTPS is just HTTP with encryption. The primary distinction between these two names is that HTTPS is more secure than HTTP since it uses TLS (SSL) encryption for all HTTP requests and answers, even the standard ones. In this article, we are going to discuss what is HTTP and HTTPS and their advantages and disadvantages in brief. Here Below is a detailed discussion on HTTP and HTTPS.

### HyperText Transfer Protocol (HTTP)

- HyperText Transfer Protocol (HTTP) is a protocol using which hypertext is transferred over the Web.
- Due to its simplicity, HTTP has been the most widely used protocol for data transfer over the Web but the data (i.e. hypertext) exchanged using HTTP isn't as secure as we would like it to be.
- In fact, hyper-text exchanged using HTTP goes as plain text i.e. anyone between the browser and server can read it relatively easily if one intercepts this exchange of data.
- The acronym for Hypertext Transfer Protocol is HTTP.
- The web server delivers the desired data to the user in the form of web pages when the user initiates an HTTP request through their browser. Above the TCP layer lies an application layer protocol called HTTP. It has given web browsers and servers certain standard principles that they can use to talk to one another.

15.Application Security

Ans; Application security protects software applications and data from cyber threats. It includes security measures during development, deployment, and after.

How is application security implemented?

Authentication: Verifies a user's identity before granting access. This can be done using a username and password, biometric data, or a one-time password.

Encryption: Translates data into another form that only people with a secret key or password can read.

Firewalls: Block unauthorized access and filter out malicious traffic.

Web application firewalls (WAF): Monitor and filter HTTP traffic between the internet and a web application.

Security monitoring: Monitors applications for suspicious activity and responds to incidents.

Anti-virus software: Removes malicious software like viruses, worms, and ransomware.

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

Ans; Encryption plays a crucial role in securing applications by transforming sensitive data into an unreadable format, ensuring that even if an attacker gains access to the data, they cannot decipher it without the proper decryption key, effectively protecting confidential information from unauthorized access and potential breaches, whether the data is stored on a device (“at rest”) or transmitted across a network (“in transit”) within an application.

Key points about encryption in application security:

Data Protection:

By scrambling data with a mathematical algorithm, encryption prevents unauthorized users from reading sensitive information like login credentials, financial details, medical records, and more, even if they intercept the data.

Confidentiality:

Encryption guarantees that only authorized parties with the correct decryption key can access the original data, maintaining confidentiality and privacy.

Multiple Application Layers:

Encryption can be implemented at different levels within an application, including database storage, data transmission between servers, and user input fields, providing comprehensive protection.

17. Software Applications and Its Types

Ans; cific tasks for users. There are many types of application software, including spreadsheet, web browser, multimedia, presentation, and educational software.

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

Ans System software manages a computer's hardware and resources, while application software helps users perform specific tasks.

#### System software

Manages a computer's hardware and resources like memory, processors, and devices

Runs in the background to maintain the computer's basic functions

Provides a platform for application software to run on

Typically bundled with a computer's operating system

#### Application software

Helps users perform specific tasks like creating documents, spreadsheets, databases, and publications

Improves productivity and efficiency

Runs on the platform provided by system softwares

### 19. Software Architecture

Ans Software architecture is the structure of a software system that defines how its components interact and are organized. It includes the rules, guidelines, and patterns that help ensure the system meets its requirements.

### 20. What is the significance of modularity in software architecture?

Ans; Modularity is important in software architecture because it makes code more organized, reusable, and maintainable. It also improves collaboration and scalability.

#### Benefits of modularity

Code organization: Modular code is easier to understand and navigate.

Collaboration: Developers can work on different modules simultaneously.

Testing: Testing is more efficient because issues can be identified in specific functions.

Scalability: Modular software can be easily adapted to different platforms or environments.

Longevity: Modular software can extend the lifespan of a product.

Portability: Modular software can operate across various systems.

Maintenance: Changes can be made to individual modules without impacting the entire system.



## 21. Layers in Software Architecture

Ans The layers of software architecture are the presentation layer, application layer, business logic layer, and data access layer.

### Presentation layer

The top layer of the architecture that handles the user interface

Also known as the UI layer, it's the most visible layer

It defines the application's look and presentation to the end-users

In web applications, it's often implemented using HTML, CSS, and JavaScript

### Application layer

Serves as a bridge between the presentation and domain layers

Controls data flow between the presentation and domain levels

Handles user requests, and executes business logic

Security and permission rules must be implemented at this layer

## 22. Why are layers important in software architecture?

Ans .Layers are crucial in software architecture because they promote “separation of concerns,” allowing each layer to focus on a specific set of functionalities, making the system more modular, maintainable, easier to test, and adaptable to changes, by isolating components and enabling independent development and modification within each layer.

Key benefits of using layers in software architecture:

Improved maintainability:

By dividing the application into distinct layers, developers can easily identify and modify code related to a specific function without impacting other parts of the system.

Enhanced modularity:

Each layer acts as a self-contained unit, allowing for independent development, testing, and deployment.

Simplified debugging:

When an issue arises, developers can pinpoint the problematic layer more easily due to the clear separation of concerns.

Scalability:

If a specific layer needs to be scaled up or down, it can be done without affecting other parts of the application.

Faster development:

Different teams can work on different layers simultaneously, leading to faster development cycles.

## 23. Software Environments

Ans

A software environment is a collection of programs, libraries, and utilities that allow users to perform specific tasks.

24.

Explain the importance of a development environment in software production.

Ans .A development environment is important in software production because it helps developers create high-quality software with fewer errors. It also helps developers test and debug code without affecting end users.

How a development environment helps

Streamlines workflows: Developers can automate manual processes like testing, debugging, and updating.

Reduces errors: Developers can test functions before releasing software to users.

Improves productivity: Developers can work more efficiently with tools like integrated development environments (IDEs).

Enhances software quality: Developers can ensure that software functions as intended.

Reduces discrepancies: Developers can test software in conditions that mimic production.

Helps with integration: Developers can easily integrate new features.

Helps with debugging: Developers can quickly identify bugs.

25.

Source Code

Ans Source code is a group of instructions a programmer writes using computer programming languages. Once a programmer writes a line or set of source code, they can later implement it in a website, application or another type of computer program to give it instructions for functioning.

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

Ans .Source code is provided to a language translator which converts it into machine-understandable code which is called machine code or object code. The computer can not understand direct source code, a computer understands machine code and executes it.

27.Github and Introductions

Ans GitHub is a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to web pages at the same time.

28.Why is version control important in software development?

Ans .Version control is important in software development because it helps teams collaborate, manage changes, and avoid errors.

Collaboration

Multiple developers can work on the same codebase at once without overwriting each other's changes

Developers can review each other's code

Change management

Version control tracks changes to code over time, including who made the changes and when

It keeps a record of every version of the code

Error recovery

Developers can revert to previous versions of the code if new changes cause errors

They can compare earlier versions to help fix mistakes

Source code management

Version control protects source code from unintended errors by restricting who can commit to the main branch

It ensures that programmers have access to all versions of the source code

## Traceability

Version control provides a detailed history of a file's life cycle, including revisions and final approval

This helps teams identify successes and failures

## Conflict resolution

Version control systems provide tools to resolve conflicts that occur when multiple developers make changes to the same file .

## 29.Student Account in Github

Ans GitHub Education offers students real-world experience with free access to various developer tools from GitHub's partners.

## 30.What are the benefits of using Github for students?

Ans GitHub can benefit students in many ways, including access to tools, learning resources, and community support.

## Tools

### GitHub Student Developer Pack

Includes a free GitHub Pro account, domain name, hosting, cloud service credits, and learning resources

### JetBrains

Includes IntelliJ IDEA, PyCharm, and WebStorm, which are IDEs for Java, Python, and JavaScript

### GitHub Copilot

Provides free access to code suggestions to help students learn programming concepts and syntax

## Learning resources

Training modules: Help students learn and innovate

Documentation: Helps students learn from their code and testing activities

Community support

GitHub Campus Experts: Help students build leadership skills and organize community events

Social features: Allow users to interact with other developer communities

Collaboration

Repositories: Allow students to store and manage their projects

Peer reviews: Promote peer reviews, discussions, and solution sharing

### 31.Types of Software

Ans There are several types of software, including application software, system software, device drivers, and more.

Application software

Also known as “apps”, these are programs that help users complete tasks.

Examples include word processors, spreadsheets, and other programs that help with tasks like writing, calculating, and more.

System software

Manages the computer itself.

Examples include operating systems, file management utilities, and disk operating systems (DOS).

Device drivers

Allow the operating system to communicate with hardware devices like printers, scanners, and graphics cards.

Freeware

Proprietary software that’s offered at no cost to users.

Shareware

Proprietary software that’s shared by the owner for trial use at little or no cost.

Programming software

Helps programmers and developers create, debug, and maintain other software programs and applications.

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

Ans The main difference between open-source and proprietary software is ownership. Open-source software is free to use and modify, while proprietary software is owned by a company or individual and may require payment.

	<b>Open-source software</b>	<b>Proprietary software</b>
Ownership	Available for anyone to access and change	Owned by a company or individual
Cost	Free to use	May require payment
Modification	Users can modify and enhance the software	Users may not be able to modify the software
Distribution	Can be used for personal and commercial purposes	May be restricted by license
Examples	Linux, Firefox, Open Office, GIMP, VLC Media Player	Windows, macOS, Microsoft Edge, Google Earth, Microsoft Office

Open-source software is often developed and maintained by non-profit organizations. Proprietary software is often developed and maintained by for-profit entities.

### Pros and cons

Open-source software is often more flexible and cost-effective. Proprietary software may offer more consistency, support, and user-friendliness.

### Security

Open-source software can benefit from community-driven innovation and security. Proprietary software companies have their own policies for disclosing vulnerabilities.

### 33 GIT and GITHUB Training

Ans You can learn Git and GitHub through tutorials, courses, and by working on projects.

#### Tutorials

FreeCodeCamp: Offers courses and articles on Git and GitHub

HubSpot: Has a tutorial that includes steps for getting started with Git and GitHub

TheServerSide: Offers a video tutorial for beginners

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

Ans Git improves collaboration in a software development team by allowing developers to work independently on different features using "branches", track changes made to the code, and easily merge their work back into the main project, minimizing the risk of conflicts and enabling parallel development while maintaining a clear history of code revisions; this is further enhanced through features like pull requests which allow for code review and discussion before merging changes.

### 35.What is the role of application software in businesses?

Ans Application software plays a critical role in businesses by enabling them to perform various functions like managing customer relationships, handling accounting tasks, project management, communication, document creation,

and data analysis, essentially streamlining operations and facilitating efficient day-to-day business activities across different departments within a company.

### 36 Software Development Process

Ans The software development process, also known as the Software Development Life Cycle (SDLC), is a series of steps used to create, test, and maintain software.

Steps in the SDLC

Planning: Set the foundation for the project

Analysis: Understand the requirements of the software

Design: Create the initial design plan for the software

Development: Write the code

Testing: Ensure the software works as expected

Deployment: Release the software to users

Maintenance: Keep the software running by fixing bugs, making updates, and monitoring performance

37.

What are the main stages of the software development process?

Ans The main stages of software development are planning, design, coding, testing, deployment, and maintenance.

#### Planning

- Involves cost-benefit analysis, scheduling, and resource estimation
- Requirement analysis is a crucial part of this stage

Design Involves designing the software.

#### Coding

- Involves writing the code for the software

#### Testing



- Involves verifying the software's faults and performance
- Testers look for defects and try to identify any performance issues

## Deployment

- Involves carefully planning, testing, and executing the software's deployment
- How you deploy your software can affect its performance, reliability, security, and user satisfaction

## Maintenance

- Involves keeping the software up and running
- Includes software updates, upgrades, and polishing
- The product can be modified based on user feedback

## 38. Software Requirement

Ans Software requirements are the specifications that define how a software system should behave and what it should do. They also specify the constraints that the software must meet.

## Purpose

- To communicate the needs and expectations of stakeholders
- To guide the design, development, testing, and maintenance of the software
- To ensure that the software meets the needs of users and other stakeholders

## Types of software requirements

- **Business requirements:** High-level objectives of the organization or customer
- **User requirements:** Describe the tasks that users must be able to perform using the new product
- **Functional requirements:** A type of software requirement

Process for writing software requirements Elicitation: Gather information and data from stakeholders, Analysis, Specification, and Validation.

## Software Requirements Specification (SRS)

- A document that lists the requirements, expectations, design, and standards for a future project
- Created by project managers or business analysts working with product stakeholders

## Software Requirements Engineering

- The process of defining, documenting, and maintaining requirements

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

Ans Software analysis is a key step in the software development process that helps ensure the software meets the needs of the users and stakeholders. It also helps identify risks and improve maintainability.

What does software analysis do?

Requirement analysis: Identifies and documents the needs of stakeholders

System analysis: Defines what the software should do to solve a problem

User experience analysis: Considers how the software will help the user solve their problems

Business requirements analysis: Ensures the software meets the business goals

### 40. Software Analysis

Ans Software analysis is the process of understanding and analyzing the requirements of a software system. It also involves designing and testing the software to meet those requirements.

### 41 What are the key elements of system design?

Ans design system includes: Assets, documentation, and processes that determine how an organization creates their digital interfaces. Foundations, tokens, core systems, and components that enable brand-aligned digital design a

### 42. Software Testing

Ans Software testing is the act of checking whether software satisfies expectations. Software testing can provide objective, independent information about the quality of software and the risk of its failure to a user or sponsor<sup>4</sup>

#### 43. Why is software testing important?

Ans Software testing is important because it helps ensure that software is high quality, secure, and meets user needs. It also helps save money and time.

##### Benefits of software testing

Customer satisfaction: Software testing helps create a more stable and user-friendly experience.

Security: Software testing helps identify vulnerabilities and ensure that the product's system is free of threats.

Performance: Software testing helps ensure that the software meets performance requirements.

Cost reduction: Software testing helps avoid costly fixes later in development.

Quality: Software testing helps ensure that the software meets user expectations and industry standards.

Scalability: Software testing helps build trust and satisfaction by assuring a defect-free application.

Bug detection: Software testing helps identify and fix bugs in the software before it's delivered.

#### 44. Maintenance

Ans Maintenance is the work of keeping equipment, machinery, or a facility in good working condition. It can also refer to the support or sustenance provided by a job.

#### 45. What types of software maintenance are there?

Ans The main types of software maintenance are corrective, adaptive, perfective, and preventive.

##### Corrective maintenance

Fixes bugs and errors that occur after software is released

Includes releasing patches to fix security vulnerabilities

Addresses performance issues

Resolves conflicts with other software

##### Adaptive maintenance

Adjusts software to match changes in the environment, such as hardware or software modifications

Ensures software is compatible with the latest technologies and standards

Responds to new operating systems, platforms, and hardware

Perfective maintenance

Improves software functionality, performance, and reliability

Modifies software code to add new features or optimize existing ones

Restructures the software system to improve changeability

Preventive maintenance

Identifies and fixes potential issues before they become critical

Includes keeping a close eye on the system, running tests, and updating components

Prevents potential system vulnerabilities

Reduces the chance of downtime or other disruptions to business operations

46. Development

Ans Development is a process of growth, change, and progress over time. It can refer to the growth of a person, society, or civilization.

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

Ans The key difference between web and desktop applications is that web applications are accessed through a web browser and require an internet connection to function, while desktop applications are installed directly on a user's device and can often operate offline, providing greater access to local hardware and potentially faster performance; essentially, web apps are accessed online, while desktop apps are installed locally on a computer.

48. Web Application

Ans A web application is software that runs in your web browser. Businesses have to exchange information and deliver services remotely. They use web applications to connect with customers conveniently and securely.

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

Ans The main advantages of using a web application over a desktop application are its accessibility from any device with an internet connection, automatic updates, cross-

platform compatibility, easier maintenance, and reduced installation needs, allowing users to access the application without needing to download or install anything on their local machine; essentially making it more convenient and user-friendly for a wider audience.

## 50.Designing

And to create, fashion, execute, or construct according to plan : devise, contrive. Design a system for tracking inventory. 2. A. : to conceive and plan out in the mind.

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

Ans UI/UX plays a crucial role in software development. It not only enhances the user experience but also increases the usability and functionality of the software. By prioritizing UI/UX in the development process, the software can attract and retain users, making it more successful in the long run.

## 52.Mobile Application

And A mobile application, or app, is a software program that runs on a mobile device like a phone or tablet. Apps can be used for many purposes, including social media, gaming, and note-taking.

Types of mobile apps:

Social media apps: Used to share photos, videos, and facilitate conversations

Game apps: Allow users to play games on their mobile devices

Note-taking apps: Used to keep notes, research, and plans

AR and VR apps: Use augmented reality to overlay digital content onto the real world

Hybrid apps: Combine native and web apps, and have similar functionality to web apps

Progressive web apps: An alternative to traditional mobile apps that don't require app store delivery or installation

Mobile app development tools:

React Native: A cross-platform application development tool that uses JavaScript

Mobile app trends:

Mobile commerce, including in-app payment services like Apple Pay and Google Pay

AR and VR in gaming, entertainment, education, and healthcare

Mobile app vs. other types of apps:

Mobile apps differ from desktop applications, which run on desktop computers

Mobile apps differ from web applications, which run in mobile web browsers

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

Ans Native apps are built for a specific operating system, while hybrid apps can run on multiple platforms.

Native app

Hybrid app

Development

Uses platform-specific languages and tools

Uses web technologies like HTML, CSS, and JavaScript

Performance

Typically has superior performance

May have slightly lower performance

Look and feel

Has a native look and feel

Mimics native look and feel

Maintenance

Requires separate maintenance for each platform

Easier maintenance with a single codebase

Availability

Available for download in platform-specific app stores

Can be made available in multiple app stores

Hybrid apps are a combination of native and web apps. They can be accessed through a web browser or downloaded through an app store.

Hybrid apps can have issues with lagging because they are run on many different operating systems. The appearance of a hybrid app can also vary between users depending on their operating system and software version.

Cross-platform apps are similar to hybrid apps, but they typically use platform-specific languages and tools that are compiled into native code.

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

Ans A Data Flow Diagram (DFD) is significant in system analysis because it provides a visual representation of how data moves through a system, helping analysts understand the relationships between different components, identify potential bottlenecks, and effectively communicate system design to stakeholders by clearly depicting the flow of information and processing steps involved in a system; essentially acting as a critical tool for analyzing and designing new systems or improving existing ones.

54. Desktop Application

Ans A desktop application is a software program that runs on a computer, like a laptop or PC. Desktop applications are installed on a user's device and run directly on it.

Examples of desktop applications:

Browsers like Chrome, Firefox, Edge, and Safari

Photo and video editing software like Adobe Photoshop, iMovie, Premier Pro, and DaVinci Resolve

Programming software like IntelliJ IDEA, Eclipse, and MS Visual Studio

Microsoft Word and , PowerPoint.

55 What are the pros and cons of desktop applications compared to

Webapplications?

Ans Compared to web applications, desktop applications generally offer better performance and offline capabilities due to running directly on the user's machine, but they lack the accessibility of web apps which can be accessed from any device with an internet connection, and may require installation and updates on a per-device basis; making them less platform-independent and potentially more complex to maintain.

56.