1. what is program

Ans . A **program** is a set of **instructions written in a programming language** that tells a computer what to do.

2. *What is Programming?*

*Ans.* ***Programming*** *is the process of* ***writing instructions (a program) that a computer can understand and execute*** *to perform specific tasks.*

*3.Types of Programming Languages*

***THEORY EXERCISE****: What are the main differences between high-level and low-level programming*

*languages?*

*Ans.* ***Types of Programming Languages***

*Programming languages are mainly divided into two broad categories:*

***1. Low-Level Languages***

* *Closer to* ***machine (hardware) level****.*
* *Difficult for humans to read and write.*
* *Very fast and memory-efficient.*
* *Examples:*
  + ***Machine Language*** *(binary 0s and 1s).*
  + ***Assembly Language*** *(uses mnemonics like ADD, MOV).*

***2. High-Level Languages***

* *Closer to* ***human language****.*
* *Easy to read, write, and understand.*
* *Must be translated into machine language using a* ***compiler or interpreter****.*
* *Examples: C, C++, Java, Python, etc.*

*4. World Wide Web & How Internet Works*

***THEORY EXERCISE****: Describe the roles of the client and server in web communication*

*Ans. When you use the internet, especially the* ***World Wide Web (WWW)****, information is exchanged between two main entities:* ***clients*** *and* ***servers****.*

** ***Client = Asks (requesting data)***

** ***Server = Answers (providing data)***

*5. Network Layers on Client and Server*

***THEORY EXERCISE****: Explain the function of the TCP/IP model and its layers*

***Ans Client Application Layer:*** *“I want this webpage.”*

1. ***Transport Layer:*** *Breaks request into reliable segments (TCP).*
2. ***Internet Layer:*** *Adds IP addresses (where it’s going).*
3. ***Network Access Layer:*** *Sends bits physically over Wi-Fi/cable.*
4. ***Server does the reverse process*** *and sends back the response.*

*he* ***TCP/IP model*** *makes sure that when you type a website name, your request travels through layers, reaches the server, and the response comes back in the right order without errors.*

*6. Client and Servers*

***THEORY EXERCISE****: Explain Client Server Communication*

*Ans Client–server communication is a two-way interaction where the client sends a request, and the server processes and responds with the required information.*

*7. Types of Internet Connections*

***THEORY EXERCISE****: How does broadband differ from fiber-optic internet?*

*Ans * ***Broadband (DSL/Cable):*** *Uses copper wires, slower, electrical signals.*

** ***Fiber-Optic:*** *Uses glass fibers, much faster, light signals.*

*8. Protocols*

***THEORY EXERCISE****: What are the differences between HTTP and HTTPS protocols?*

*Ans * ***HTTP = Normal communication, not secure.***

** ***HTTPS = Secure communication, encrypted with SSL/TLS.***

*9. Application Security*

***THEORY EXERCISE****: What is the role of encryption in securing applications?*

*Ans Encryption is crucial in application security because it* ***keeps data private, ensures it isn’t tampered with, and secures communication between clients and servers.***

*10. Software Applications and Its Types*

***THEORY EXERCISE****: What is the difference between system software and application software?*

*Ans*

| ***Feature*** | ***System Software 🖥️*** | ***Application Software 📱*** |
| --- | --- | --- |
| *Purpose* | *Controls hardware & runs the system* | *Helps users perform specific tasks* |
| *Interaction* | *Works in the background* | *Directly interacts with the user* |
| *Dependency* | *Needed for applications to run* | *Depends on system software* |
| *Examples* | *OS, drivers, utilities* | *MS Office, browsers, games* |

*11. Software Architecture*

***THEORY EXERCISE****: What is the significance of modularity in software architecture?*

*Ans*

*Modularity is significant in software architecture because it promotes simplicity, reusability, and flexibility, making the system easier to develop, maintain, and scale.*

*12. Layers in Software Architecture*

***THEORY EXERCISE****: Why are layers important in software architecture?*

*Ans*

*n* ***software architecture****, a* ***layer*** *is a logical separation of related functionalities, where each layer has a specific role and communicates with adjacent layers. For example, in a typical* ***three-tier architecture****:*

* ***Presentation Layer*** *→ user interface*
* ***Business Logic Layer*** *→ processing and rules*
* ***Data Layer*** *→ data storage and access*

***Why Layers are Important***

1. ***Separation of Concerns***
   * *Each layer handles a specific responsibility (UI, logic, or data), reducing complexity.*
2. ***Maintainability***
   * *Changes in one layer (e.g., database updates) usually don’t affect others, making maintenance easier.*
3. ***Reusability***
   * *Layers (like authentication, data access, or logging) can be reused across multiple applications.*
4. ***Scalability***
   * *Individual layers can be scaled independently (e.g., load balancing on presentation servers or database clustering).*
5. ***Flexibility and Extensibility***
   * *New features can be added in one layer without rewriting the entire system.*
6. ***Testability***
   * *Layers can be tested separately, ensuring reliable and modular testing (e.g., unit testing business logic without UI).*
7. ***Security***
   * *Layers allow security checks at multiple levels (e.g., user authentication in presentation, data validation in business logic).*
8. ***Standardization***
   * *Enforces a structured approach, making systems easier to design, document, and understand.*

*13 Software Environments*

***THEORY EXERCISE****: Explain the importance of a development environment in software production*

*Ans*

*There are different Types of environments in Industry.*

*1.The analysis and design environment*

*2.The development environment*

*3.The common build environment*

*4.The testing environment*

*5.The production environment*

*14. Source Code*

***THEORY EXERCISE****: What is the difference between source code and machine code?*

*Ans*

| ***Aspect*** | ***Source Code*** | ***Machine Code*** |
| --- | --- | --- |
| ***Readability*** | *Human-readable* | *Computer-readable (binary)* |
| ***Written By*** | *Programmers* | *Generated by compiler/interpreter* |
| ***Execution*** | *Needs compilation/interpretation* | *Executes directly on CPU* |
| ***Form*** | *High-level (C, Java, Python, etc.)* | *Low-level (0s and 1s)* |
| ***Modifiability*** | *Easy to modify and maintain* | *Very hard to modify* |

*15. Github and Introductions*

***THEORY EXERCISE****: Why is version control important in software development?*

*Ans Version control is important because it ensures* ***collaboration, safety, and reliability*** *in software development, allowing teams to manage changes systematically and deliver high-quality software.*

*16. Student Account in Github*

***THEORY EXERCISE****: What are the benefits of using Github for students?*

*Ans GitHub benefits students by providing a platform to* ***learn, collaborate, build a portfolio, and access free professional tools****, preparing them for real-world software development and career opportunities.*

17. *Types of Software*

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

*Ans*

| ***Aspect*** | ***Open-Source Software*** | ***Proprietary Software*** |
| --- | --- | --- |
| ***Source Code*** | *Public and modifiable* | *Closed, not available to users* |
| ***Cost*** | *Usually free* | *Paid (license/subscription)* |
| ***Customization*** | *Fully customizable* | *Limited or not allowed* |
| ***Support*** | *Community-driven* | *Vendor-provided official support* |
| ***Ownership*** | *Shared by community* | *Controlled by company/individual* |
| ***Examples*** | *Linux, Firefox* | *Windows, Photoshop* |

*18. GIT and GITHUB Training*

***THEORY EXERCISE****: How does GIT improve collaboration in a software development team?*

*Ans*

*Git improves collaboration by enabling* ***parallel work, safe experimentation, version tracking, and easy sharing****, ensuring that teams can build software together smoothly and efficiently.*

*19. Application Software*

***THEORY EXERCISE****: What is the role of application software in businesses?*

*Ans Application software in businesses plays the role of* ***enhancing productivity, supporting decision-making, improving customer service, and streamlining operations****, making organizations more efficient and competitive.*

*20 Software Development Process*

***THEORY EXERCISE****: What are the main stages of the software development process?*

*Ans*

| ***Stage*** | ***Purpose*** | ***Output*** |
| --- | --- | --- |
| *Requirement Analysis* | *Identify user needs* | *Requirement Document* |
| *System Design* | *Plan system structure & UI* | *Design Document/Architecture* |
| *Implementation* | *Write the code* | *Source Code / Modules* |
| *Testing* | *Ensure quality and correctness* | *Tested Software* |
| *Deployment* | *Release software to users* | *Live System* |
| *Maintenance* | *Fix & improve software after release* | *Updated/Improved System* |

*21. Software Requirement*

***THEORY EXERCISE****: Why is the requirement analysis phase critical in software development?*

*Ans The requirement analysis phase is critical because it sets the* ***direction, clarity, and scope*** *of the project, ensuring that the final software meets user needs while saving time, cost, and effort.*

*22. Software Analysis*

***THEORY EXERCISE****: What is the role of software analysis in the development process?*

*Ans The role of software analysis is to* ***refine requirements, assess feasibility, and create clear functional specifications****, ensuring that the design and development phases build exactly what the users need.*

*23. System Design*

***THEORY EXERCISE****: What are the key elements of system design?*

*Ans The key elements of system design include* ***architecture, data, interface, module, security, workflow, infrastructure, and reliability planning****. Together, these form the blueprint that guides successful system development.*

*24. Software Testing*

***THEORY EXERCISE****: Why is software testing important?*

*Ans Software testing is important because it ensures that the software is* ***error-free, secure, reliable, and meets user needs****, while also reducing cost, risk, and time-to-market.*

*25. Maintenance*

***THEORY EXERCISE****: What types of software maintenance are there?*

*Ans * ***Corrective*** *→ Fix errors*

** ***Adaptive*** *→ Adjust to environment changes*

** ***Perfective*** *→ Improve or enhance*

** ***Preventive*** *→ Avoid future issues*

*26. Development*

***THEORY EXERCISE****: What are the key differences between web and desktop applications?*

*Ans * ***Web apps*** *= Browser-based, accessible anywhere, no install, need internet.*

** ***Desktop apps*** *= Installed locally, faster, can work offline, OS-dependent.*

*27. Web Application*

***THEORY EXERCISE****: What are the advantages of using web applications over desktop applications?*

*Ans Web applications are* ***more accessible, easier to update, cost-effective, and platform-independent*** *compared to desktop applications.*

*28. Designing*

***THEORY EXERCISE****: What role does UI/UX design play in application development?*

*Ans UI/UX design makes applications* ***easy, enjoyable, and efficient*** *to use, ensuring both users and businesses achieve their goals.*

*29. Mobile Application*

***THEORY EXERCISE****: What are the differences between native and hybrid mobile apps?*

*Ans * ***Native apps*** *= High performance, best UX, but costlier and time-consuming to build.*

** ***Hybrid apps*** *= Faster, cheaper, cross-platform, but less smooth and powerful than native.*

*30. DFD (Data Flow Diagram)*

***THEORY EXERCISE****: What is the significance of DFDs in system analysis?*

*Ans DFDs are significant in system analysis because they* ***simplify understanding, improve communication, clarify requirements, and provide a strong foundation for designing efficient systems****.*