

Module 1 – Overview of IT Industry

What is a Program?

Program is set of instructions.

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

Program is a set of instructions that help a programmer to perform a specific functions.

What is Programming?

The process of writing programs and instructions to perform a specific task is known as programming.

THEORY EXERCISE: What are the key steps involved in the programming process?

The key steps involved in programming process are Planning, Designing, Coding, Testing, Debugging and Shipping/Uploading the code or project.

Types of Programming Languages.

There are mainly four types of programming languages: Procedural programming language, Object oriented language, functional language and logical programming language.

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

High level: Easy to Read Understand and modify, Needs a Compiler, Slow Execution speed, used to write application programs.

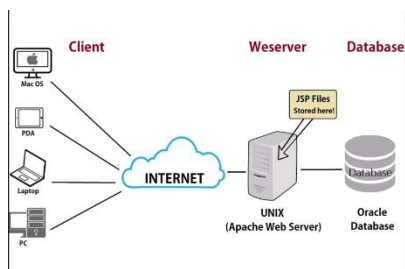
Low level: Difficult to read Understand and modify, Doesn't need a compiler, Fast Execution speed, used to write system programs.

World Wide Web & How Internet Works

The WWW helps us to browse websites and webpages over the internet. It helps us to connect with people, data such as text, images, videos, etc and also provides us a platform to share the same.

Internet is a collection of connected networks. When we share some data then at the base it is transferred from your device via packets to your router and further through internet routes to the desired location.

LAB EXERCISE: Research and create a diagram of how data is transmitted from a client to a server over the internet



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

In the client server architecture the client acts as a receiver and server acts as a sender. The client requests some data from the server and the server sends the requested data back to the client in the form of response.

THEORY EXERCISE: Explain Client Server Communication.

The client sends a request to the server, specifying the desired service or data. The server receives the request and processes it, retrieves the data from a database and sends back to the client in the form of response.

Types of Internet Connections.

Types of internet connections are wireless, cable network, fiber optic, DSL, etc.

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

Fiber Optic offers the fastest speeds, excellent, reliable but is costly, limited availability and more prone to damage.

Cable internet provides High speed and is widely used but is costly and could provide low speed based on location.

Wireless Is widely used, affordable, and easy to set up but the big limitation is this speed and reliability based on locations.

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

Broadband is a general term for high-speed internet connections, while fiber-optic internet is a specific type of broadband that uses fiber optic cables to transmit data.

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

SQL Injection: Attackers inject malicious SQL code through input fields to manipulate databases. To avoid this Use prepared statements and validate all user inputs.

Cross-Site Scripting: Attackers inject malicious scripts into webpages viewed by others.

Sanitize user inputs and use output encoding; implement Content Security.

Broken Authentication: Poor authentication allows attackers to compromise user accounts. To avoid this Use strong password policies, multi-factor authentication, and secure session management.

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

Encryption protects data by converting it into unreadable form, ensuring confidentiality during storage and transmission.

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

5 Applications are Opera Browser, VS Code, Chatgpt, Windows, VLC, etc.

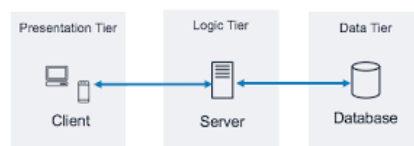
THEORY EXERCISE: What is the difference between system software and application software?

System software is a software that runs the computer hardware and basic functions.

Example operating system.

Application software is a software that helps users perform specific tasks. Example Word, browsers, etc.

LAB EXERCISE: Design a basic three-tiers software architecture diagram for a web application



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

Modularity in software architecture makes software easier to build, test, update, etc. It lets you break a system into smaller, manageable parts called modules that can work together to make the software.

THEORY EXERCISE: Why are layers important in software architecture?

Layers help organize our code by separating the User interface, logic of the application, and data. This makes the application cleaner, easier to maintain, and more flexible to change in future if need arises.

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

A development environment is important because it gives developers the tools, functions, and settings needed to build and test software effectively. It helps maintain consistency across teams, speeds up the development process, and reduces the chances of bugs by catching issues early.

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

Source code is human-readable code written by programmers often in high level programming languages.

Machine code is binary code understood and executed by computers in low level machine language.

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

Version control helps programmer to tracks code changes, enables collaboration, prevents conflicts, and allows rollback to previous versions of the code if in case of any problems.

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

GitHub helps students collaborate, track projects, showcase projects, learn about Git, and access open-source code uploaded by other programmers, it also provides github pages where we can publish our websites.

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

Open-source software are software whose source code is publicly available and it is modifiable and free to use.

Proprietary software on the other hand has restricted access to the source code, it is owned by companies, and usually paid.

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

GIT improve collaboration in a software development by letting team members work on code simultaneously, track changes, merge updates, and resolve conflicts efficiently.

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

Application software helps businesses automate repetitive tasks, improve overall productivity or employees, manage company data, and support decision-making.

LAB EXERCISE: Create a flowchart representing the Software Development Life Cycle (SDLC)



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

Main stages of software development process are as follows: Planning, Analysis, Designing, Implementation, Testing and Maintenance.

LAB EXERCISE: Write a requirement specification for a simple library management system

Requirements for Library Management System are as follows:

Book Management: Proper book arrangement along with ids, categorize in different genres, etc.

User Management: User login and security management, profile edit and update functionality, user saves and purchase details security, etc.

App Management: Scalable and stable database, must have features like search, discover, recent launch, etc.

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

Requirement analysis is critical in software development because it helps us to understand what the client really needs. It gives us the foundation to design, development, and test the application.

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

Software analysis helps us break down what the app should do and how it should function. Its goal is to identify user needs, different features, and overall system requirements, all of which are necessary before starting the project.

LAB EXERCISE: Design a basic system architecture for a food delivery app.

User Side: User login, discover various items, search feature, account related functionality, latest launch feature, etc.

Seller Side: seller login, seller account related functionality, product listing and update function, user hike and orders related functionalities, etc.

Admin Side: handle user and seller requests, handle the user hikes, etc. Look for Vulnerability in the app, etc.

System Side: maintain and scale database of user, seller, products, etc.

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

Key elements of system design are: architecture, modules, data flow, interface, database, performance, security, etc.

LAB EXERCISE: Develop test cases for a simple calculator program

Test cases for simple calculator program are:

Addition: 2+5 Output: 5

Subtraction: 5-3 Output: 2

Multiplication: 2*2 Output: 4

Division: 10/2 Output: 5

Modulus: 10%3 Output: 1

THEORY EXERCISE: Why is software testing important?

Software Testing is done to find out if the application contains any bugs, glitches, and other unintended problems that are in the app and can affect the user experience in future.

THEORY EXERCISE: What types of software maintenance are there?

There are four types of software maintenance: corrective, adaptive, perfective, preventive.

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

Web Application: Runs on the cloud, less secure, needs internet to run, global access, etc.

Desktop Application: Runs on the local machine, more secure as compared to web app, does or does not need internet, local access, etc.

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

The advantages of using web app over desktop app are: More Scalable, can be accessed globally, auto update to recent changes, easy to connect, consume less resources, 24/7 access.

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

UI/UX design shows how users interact with an app. Good design makes the app easy to use which boosts user satisfaction and keeps user engaged with the app and also recommend the app to others as well.

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

Native apps are built specifically for one platform like iOS or Android using specific languages, so that they can run fast and use platform specific features. whereas hybrid apps use languages and programs that can run on all sorts of platforms

LAB EXERCISE: Create a DFD for a hospital management system

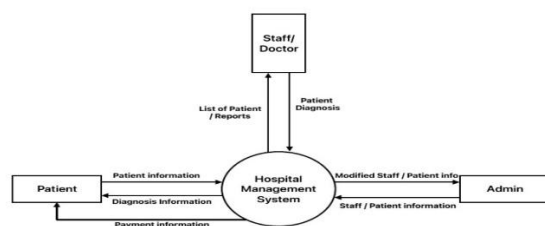


Fig. DFD level 0 of Hospital Management System

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

DFD show a visual representation of how data flows through the system. They help understand processes, identify data sources and destinations, and spot problems early on before making the whole system. They are so simple and easy to understand that even a lay person can also understand the flow of program.

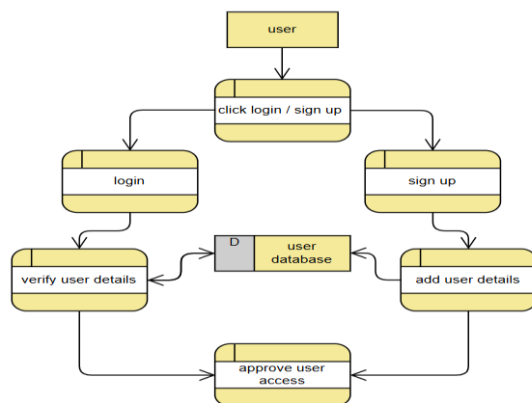
THEORY EXERCISE: What are the pros and cons of desktop applications compared to web applications?

pros and cons of desktop applications compared to web applications are as follows:

pros: Fast, works offline, Secured.

cons: Manual updates, installation required, consume device resources.

LAB EXERCISE: Draw a flowchart representing the logic of a basic online registration system.



THEORY EXERCISE: How do flowcharts help in programming and system design?

Flowcharts help the programmer in programming and system design by giving a clear visual representation of how a program or system would work. They show the logic, steps, and decision points, making it easier to plan, debug, and explain the process to others before writing code.