Module – 1 OVERVIEW of IT INDUSTRY THEORY EXCERISE

1. **What is a program?**

   Ans. = A program is a sequence of instructions written using a programming language to perform a specific task or solve a particular problem. These instructions are executed by a computer's processor in a step-by-step manner.

2. **What is programming ?**

   Ans. = Programming is the process of writing instructions that a computer can understand and follow to do a specific task.

   - It's how we communicate with computers using special languages like Python, Java, or C++.
   - These instructions are called a program**.**

3. **What is key steps in programming process ?**

Ans. = The programming process involves several important steps to go from an idea to a working computer program.

- Define the Problem
- Plan the Solution
- Write the Code
- Test the Program
- Debug and Improve
- Document the Code
- Maintain the Program

4. **Different between high level and low level languages ?**

   Ans. =

Feature	High-Level Language	Low-Level Language
Definition	A language that is easy for humans to read and write	A language that is close to machine code
Examples	Python, Java, C++, JavaScript	Assembly language, Machine language

Readability	Easy to understand	Hard to read; uses symbols and numbers
Abstraction Level	High (closer to human thinking)	Low (closer to hardware)
Speed of Execution	Slower (needs translation to machine code)	Faster (runs closer to the hardware)
Ease of Programming	Easier to write and debug	Difficult; requires deep hardware knowledge
Portability	Portable across different systems	Not portable; specific to one type of CPU
Use Case	Application development, web, mobile apps, etc.	Writing device drivers, embedded systems, OS parts

5. Roles of the client and server in web communication ?

Ans. = 1. Client

Role:

□ The client is usually a web browser (like Chrome, Firefox, or Safari).

□ It sends requests to the server and displays the response (like a web page).

Example:

When you visit **www.example.com** your browser (the client) asks the server for that page.

2. Server

Role:

□ The server is a computer or system that stores and serves web content.

□ It responds to client requests with the needed data.

Example:

The server receives the request for www.example.com finds the page, and sends it back to the browser.

6. Function of the TCP/IP model and its layers?

Ans. = The TCP/IP model is the foundation of how devices communicate over the internet. It defines how data is sent, received, and routed between computers.

❖ Layers of the TCP/IP Model

I. Application Layer
II. Transport Layer
III. Internet Layer
IV. Network Access Layer

7. Explain client server communication ?

Ans. = Client-server communication is a model in which:

- client (like a web browser or app) sends requests for data or services.
- **A** server (a powerful computer or system) responds with the requested information or action.
- They communicate over a network, usually using the Internet**.**

8. How does broadband differ from fiber optic internet ?

Ans. =

Feature	**Broadband**	**Fiber Optic Internet**
Definition	A general term for high-speed internet that can use various technologies (DSL, cable, satellite, fiber, etc.)	A type of broadband that uses light signals through thin glass or plastic fibers to transmit data
Technology Used	Can be DSL (copper wires), cable (coaxial), satellite, or fiber	Uses fiber optic cables made of glass/plastic strands
Speed	Varies widely depending on type; often slower than fiber (e.g., DSL up to 100 Mbps, cable faster)	Extremely fast, often ranging from 100 Mbps to 10 Gbps and beyond
Latency (Delay)	Higher latency, especially with satellite or DSL	Very low latency — great for gaming, video calls, streaming
Reliability	Can be affected by distance (DSL), weather (satellite), or cable quality	Very reliable and stable connection, less prone to interference
Distance Limitations	Performance can degrade over longer distances (DSL especially)	Can transmit data over long distances without loss
Availability	Widely available, even in rural areas through DSL or satellite	Increasingly available but mostly in urban/suburban areas
Cost	Generally cheaper, but varies by technology and provider	Usually more expensive to install but prices are dropping

9. What are the different between HTTP and HTTPS protocols ?

Ans. =

Feature	HTTP (HyperText Transfer Protocol)	HTTPS (HTTP Secure)
Full Form	HyperText Transfer Protocol	HyperText Transfer Protocol Secure
Security	Not secure — data is sent in plain text	Secure — data is encrypted using SSL/TLS
Port Number	Uses port **80**	Uses port **443**
Encryption	No encryption; data can be intercepted easily	Encrypts data to protect it from hackers
Use Case	General web browsing where security isn't critical	Online banking, shopping, login pages — where security is essential
Performance	Slightly faster since no encryption overhead	Slightly slower due to encryption but usually unnoticeable
URL Prefix	Starts with `http://`	Starts with `https://`
Trust Indicator	No padlock icon in browsers	Shows padlock icon in browsers indicating secure connection

10. Role of encryption in securing applications.

Ans. = Encryption is the process of converting data into a coded form that only authorized parties can understand. It protects sensitive information from being read by unauthorized users.

- Protects Data Privacy
- Secures Communication
- Maintains Data Integrity
- Authenticates Users and Systems
- Builds User Trust
- Compliance with Regulations

11. System software vs application software.

Ans. =

Feature	System Software	Application Software
Purpose	Manages and controls computer hardware and provides a platform for running application software	Helps users perform specific tasks or applications
Examples	Operating Systems (Windows, macOS, Linux), Utility programs, Device drivers	Word processors, Web browsers, Games, Media players
Function	Runs in the background and manages system resources	Directly used by the user for tasks like writing, browsing, or

		gaming
Dependency	Runs independently and allows application software to function	Depends on system software to run
Interface	Usually runs without direct user interaction, mostly works behind the scenes	Has a graphical user interface (GUI) for user interaction
Installation	Comes pre-installed or installed during OS setup	Installed by users as needed
Examples of Tasks	File management, memory management, hardware control	Creating documents, editing photos, playing music

12. Significance of modularity In software architecture .

Ans. = Modularity means designing software as a collection of separate, independent modules (or components), each responsible for a specific part of the functionality.

- Improves Maintainability
- Enhances Reusability
- Simplifies Development
- Supports Scalability
- Increases Reliability
- Facilitates Understanding

13. Important layer in software architecture ?

Ans. = In layered software architecture, the system is divided into logical layers, each with a specific responsibility. This makes the software easier to build, test, maintain, and scale.

Layer	**Role / Function**
1. Presentation Layer (UI Layer)	- Interface between the user and the system
2. Application Layer (Service Layer)	- Contains business logic
3. Business Logic Layer	- Defines the core operations of the application
4. Data Access Layer (DAL)	- Manages communication with databases
5. Database Layer (Data Layer)	- Stores actual data in database

14. Importance of a development environment.

Ans. = A development environment is the set of tools, software, and systems that developers use to write, test, and debug code.

- Code editor or IDE (e.g., VS Code, IntelliJ)
- Compiler or interpreter
- Debugger
- Version control tools (e.g., Git)
- Testing tools
- Simulators or emulators

15. Source code vs machine code.

Ans. =

Aspect	Source Code	Machine Code
Definition	Human-readable instructions written by a programmer using a programming language	Binary code (0s and 1s) that the computer's processor understands directly
Written In	High-level languages like Python, Java, C++	Low-level binary format specific to the CPU
Readability	Easy for humans to read and edit	Not readable by humans (pure binary)
Execution	Needs to be compiled or interpreted	Executed directly by the computer's CPU
Example	`Print ("Hello, world!")`	`01001000 01100101 01101100 01101100 01101111`
Modifiability	Easy to change and update	Difficult to change directly
Role in Programming	Starting point of software development	Final product after compilation or interpretation

16. Importance of version control.

Ans. = Version Control is a system that helps developers track, manage, and control changes to source code or project files over time.

The most popular version control system is **Git**, often used with platforms like **GitHub**, **GitLab**, or **Bitbucket**.

- Tracks Every Change
- Enables Team Collaboration
- Supports Rollback and Recovery

- Manages Different Versions (Branching)
- Improves Code Quality
- Facilitates Continuous Integration/Deployment (CI/CD)
- Documentation of Project Progress

17. Benefits of using github for students.

Ans. = GitHub is more than just a place to store code — it's a powerful tool that offers real-world experience, collaboration opportunities, and career advantages for students learning programming or software development.

- Real-World Version Control Experience
- Portfolio Building
- Collaboration and Teamwork
- Learning from Open Source Projects
- Free Student Benefits (GitHub Student Pack)
- Backup and Cloud Storage
- Issue Tracking and Documentation Practice
- Community and Networking

18. Open source vs proprietary software.

Ans. =

Feature	Open Source Software	Proprietary Software
Source Code Access	Public — anyone can view, use, modify, and share it	Closed — only the owner/company has access
Cost	Usually free to use	Often requires payment or license
Control	Users have full control over features and changes	Control is limited to what the developer allows
Customization	Highly customizable by anyone	Customization is limited or not allowed
Support	Community-driven (forums, contributors)	Official support from the vendor (may cost extra)
Examples	Linux, Firefox, LibreOffice, GIMP	Windows, Microsoft Office, Adobe Photoshop
Security	Code is open — more people can find & fix bugs quickly	Security depends on the vendor; issues may take time to patch
Licensing	Licensed under open licenses like **GPL**, **MIT**	Licensed under strict terms by the software company
Development Model	Collaborative — often by communities or developers worldwide	Centralized — developed and maintained by a single organization

19. How git improve collaboration.

Ans. = Git is a distributed version control system that makes it easy for multiple people to work on the same project whether it's code, documentation, or any other file.

- **Tracks Every Change**
- Enables Teamwork with Branches
- Supports Safe Merging
- Detects and Resolves Conflicts
- Works Offline
- Enables Code Reviews
- Keeps Backups of Code
- Integrates with CI/CD Tools

20. Role of application software in businesses.

Ans. = Application software is any program designed to perform specific tasks for the user — such as writing reports, managing inventory, or analyzing data.

- Improves Productivity
- Enhances Communication
- Automates Business Processes
- Manages Business Operations
- Supports Decision-Making
- Improves Customer Service
- Enhances Data Management
- Enables Online Presence & Sales
- Ensures Financial Accuracy
- Supports Remote Work

21. Main stages of the software development process.

Ans. = The Software Development Life Cycle (SDLC) is a step-by-step process used to design, develop, test, and maintain software.

- Requirement Gathering & Analysis
- System Design
- Implementation (Coding)
- Testing

- Deployment
- Maintenance & Updates

22. Importance of requrirement analysis.

Ans. = Requirement analysis is the process of understanding, documenting, and validating what the client, end-users, or stakeholders need from a software system before development begins.

It’s the foundation of a successful software project.

- Defines Clear Goals
- Avoids Miscommunication
- Helps Create Better Design
- Saves Time and Money
- Improves Quality
- Supports Testing
- Enables Better Project Planning
- Prepares for Change

23. Role of software analysis.

Ans. = Software Analysis is the process of studying and understanding the requirements and problems that the software aims to solve. It acts as a bridge between the user needs and the software design.

- Understanding User Needs
- Problem Identification
- Requirement Specification
- Feasibility Study
- Foundation for Design
- Risk Reduction
- Improves Communication
- Supports Testing and Validation

24. Key elements of system design.

Ans. = System design is the phase where the blueprint of the software system is created based on requirements. It defines how the system will work, its components, and interactions.

- Architecture Design
- Data Design
- Interface Design
- Component Design
- Security Design
- Performance Design
- Error Handling and Recovery
- Deployment Design
- Documentation

25. Importance of software testing.

Ans. = Software testing is the process of evaluating and verifying that a software application or system meets the specified requirements and works as expected without defects.

- Ensures Quality
- Detects Bugs Early
- Improves Security
- Enhances User Experience
- Saves Time and Money
- Verifies Requirements
- Facilitates Maintenance
- Builds Customer Confidence

26. Types of software maintenance.

Ans. = Software maintenance is the process of modifying and updating software after it is released to correct issues, improve performance, or adapt it to a new environment.

There are **4 main types of software maintenance**:

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

27. Web application vs desktop application.

Ans. =

Feature	Web Application	Desktop Application
Definition	Runs in a **web browser** using the internet	Installed and runs directly on a computer
Installation	No installation required (access via URL)	Must be installed on each device
Platform Dependency	Usually cross-platform (works on any OS/browser)	Often platform-specific (e.g., Windows, macOS)
Updates	Updated centrally on the server	Must be updated manually on each machine
Internet Required	Yes, typically needs an active internet connection	No (can work offline unless it connects to a server)
Performance	Depends on internet speed and browser	Usually faster and more powerful
Security	Requires strong web security (HTTPS, auth, etc.)	More control over local data, but risks if device is infected
Examples	Gmail, Google Docs, Facebook	Microsoft Word, Adobe Photoshop, VLC Media Player

28. Advantages of web application over desktop application.

Ans. =

- No Installation Required
- Automatic Updates
- Cross-Platform Compatibility
- Accessible from Anywhere
- Better Collaboration
- Lower Maintenance Costs
- Centralized Data Storage
- Easier Scalability
- Improved Security Control

29. Role of ui/ux design in application development .

Ans. = UI (User Interface) and UX (User Experience) design play a **crucial role** in how users interact with and feel about an application. It’s not just about how the app looks — it’s about how well it works for the people using it.

- Improves Usability
- Boosts User Satisfaction
- Increases User Retention
- Enhances Accessibility
- Strengthens Brand Identity
- Reduces Development Costs
- Supports Business Goals

30. Native vs hybrid mobile apps

Ans. = Both **native** and **hybrid** mobile apps are used to deliver mobile experiences — but they differ in how they’re built, how they perform, and where they run.

Feature	Native App	Hybrid App
Definition	Built specifically for one platform (iOS or Android)	Built using web technologies (HTML, CSS, JavaScript) and wrapped to run on multiple platforms
Technology Used	Swift (iOS), Kotlin/Java (Android)	Ionic, React Native, Flutter, Cordova, etc.
Performance	Faster and more responsive	Slightly **s**lower due to web layer
Platform Support	Platform-specific (iOS or Android)	Cross-platform (iOS and Android from one codebase)
Access to Device Features	Full access to all device APIs and hardware features	Limited access; needs plugins for advanced features
User Experience (UX)	Consistent with platform standards (native look/feel)	Might not feel 100% native
Development Time	Longer, separate codebases for each platform	Faster, one codebase for all platforms
Maintenance	Harder to maintain two codebases	Easier with one shared codebase
Examples	WhatsApp (native), Instagram (native)	Instagram (originally hybrid), Twitter (partially hybrid)

31. Significance of DFDs in system analysis.

Ans. = A Data Flow Diagram (DFD) is a visual representation that shows how data moves through a system — what inputs are processed, where they go, and what outputs are produced.

In system analysis, DFDs are essential tools for understanding and designing systems before development begins.

- Clarify System Requirements
- Improve Communication
- Identify Data Sources and Destinations
- Reveal Redundancies and Inefficiencies
- Support Structured Design
- Aid in Documentation
- Helps in Testing and Validation

32. Pros and cons of desktop application .

Ans. = pros=

Advantage	Description
1. Works Offline	No internet needed once installed — can be used anytime.
2. Better Performance	Typically faster and more responsive than web apps.
3. Full Access to System Resources	Can use hardware (like printers, scanners, GPU) more efficiently.
4. More Feature-Rich	Can include complex functions not limited by browser capabilities.
5. Greater Security Control	Data can be stored and protected locally, with full user control.

Cons=

Disadvantage	Description
1. Requires Installation	Must be downloaded and installed on each device individually.
2. Limited Portability	Only available on the device it's installed on (unless synced).
3. Harder to Update	Users must manually install updates or patches.
4. Platform Dependent	Often works only on specific operating systems (Windows/macOS).
5. Higher Maintenance	More effort required for version control, support, and backups.

33. How flowcharts help in programming and system design.

Ans. =

- Visual Understanding of Logic
- Simplifies System Design
- Improves Communication
- Helps in Problem Solving

- Assists in Documentation
- Guides Development
- Useful in Testing

  Common Flowchart Symbols:

  - Rectangle
  - Diamond
  - Parallelogram
  - Oval
  - Arrow