Theory Exercises – Module 1: Overview of IT Industry

---

1. What is a program and how does it function?
A program is a set of instructions written in a programming language that tells a computer how to perform a specific task. It processes input data, executes logic, and produces output to achieve desired results.

---

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

1. Problem Analysis
2. Planning the solution (algorithm)
3. Coding the program
4. Testing and debugging
5. Deployment
6. Maintenance and updates

---

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

- High-level languages (e.g., Python, Java) are closer to human language, easy to read/write, and portable.
- Low-level languages (e.g., Assembly, Machine code) are hardware-specific and offer better performance and control.

---

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

- Client: Sends requests to a server for data or services (e.g., a web browser).
- Server: Processes requests and sends back the appropriate responses (e.g., webpage data).

5. Explain the function of the TCP/IP model and its layers.
The TCP/IP model organizes communication over networks into four layers:

1. **Application Layer** – Interface for applications (HTTP, FTP)
2. **Transport Layer** – Reliable data transmission (TCP/UDP)
3. **Internet Layer** – Routes data (IP)
4. **Network Access Layer** – Physical data transmission (Ethernet)

---

6. Explain Client-Server Communication.
Client-server communication involves clients (users) requesting services or data, and servers responding to those requests. It's a structured interaction used in web, email, and database systems.

---

7. How does broadband differ from fiber-optic internet?

- Broadband: Uses copper cables, offers moderate speed and reliability.
- Fiber-optic**:** Uses light signals through fiber cables, offers high speed and superior reliability.

---

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

- HTTP: Unencrypted, vulnerable to attacks.
- HTTPS**:** Encrypted using SSL/TLS, provides secure data transfer between client and server.

---

9. What is the role of encryption in securing applications?
Encryption protects sensitive data by converting it into unreadable form, ensuring confidentiality, integrity, and security during data storage and transmission.

---

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

- System software**:** Manages hardware and system operations (e.g., OS).
- Application software**:** Helps users perform specific tasks (e.g., word processing).

11. What is the significance of modularity in software architecture?
Modularity breaks down a system into independent, reusable modules. It enhances code maintainability, scalability, debugging, and teamwork efficiency.

---

12. Why are layers important in software architecture?
Layers separate concerns (e.g., UI, business logic, data access), making the system easier to develop, manage, test, and scale.

---

13. Explain the importance of a development environment in software production.
A development environment provides the necessary tools, libraries, and isolated space to write, test, and debug code efficiently before deployment.

---

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

- Source Code: Human-readable code written by programmers.
- Machine Code: Binary instructions understood and executed by the computer's processor.

---

15. Why is version control important in software development?
Version control tracks code changes, enables collaboration, and allows developers to revert to previous versions, manage updates, and prevent conflicts.

---

16. What are the benefits of using GitHub for students?
GitHub helps students collaborate on projects, manage code, build portfolios, track progress, and learn real-world development practices.

---

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

- Open-source: Source code is publicly available and modifiable.
- Proprietary: Owned by companies, closed-source, with restricted usage rights.

Theory Exercises – Module 1 (Part 2)

---

18. How does Git improve collaboration in a software development team?
Git enables multiple developers to work simultaneously without conflicts. Features like branches, merge, and version history improve coordination, backup, and code review.

---

19. What is the role of application software in businesses?
Application software automates tasks, manages data, supports communication, and enhances productivity across business departments like finance, HR, and operations.

---

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

1. Requirement Analysis
2. System Design
3. Implementation (Coding)
4. Testing
5. Deployment
6. Maintenance

---

21. Why is the requirement analysis phase critical in software development?
It ensures a clear understanding of user needs, reduces misunderstandings, prevents costly changes later, and lays the foundation for successful system design.

---

22. What is the role of software analysis in the development process?
Software analysis identifies what the system should do by studying user needs, defining functions, and determining system boundaries and interactions.

---

23. What are the key elements of system design?
Key elements include:

- System Architecture
- Data Flow
- Interface Design
- Security Design
- Component and Module Design

---

24. Why is software testing important?
Testing identifies bugs, ensures software correctness, verifies functionality, and improves reliability, user satisfaction, and product quality.

---

25. What types of software maintenance are there?

1. Corrective – Fixing bugs
2. Adaptive – Adjusting to new environments
3. Perfective – Enhancing performance or features
4. Preventive – Avoiding future issues

---

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

Aspect	**Web Applications**	**Desktop Applications**
Accessibility	Via browser, from any device	Installed on a specific device
Updates	Centralized and automatic	Manual and user-dependent
Performance	Internet-dependent	Generally faster and offline-ready

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

- No installation required
- Cross-platform support
- Easy centralized updates
- Accessible from anywhere via the internet

---

28. What role does UI/UX design play in application development?
UI/UX design ensures a smooth, intuitive, and enjoyable user experience, which increases user engagement, reduces errors, and supports application success.

---

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

Aspect	Native Apps	Hybrid Apps
Performance	High	Moderate
Development	Platform-specific (e.g., iOS/Android)	Single codebase (e.g., React Native)
Features	Full device access	Limited or via plugins

---

30. What is the significance of DFDs in system analysis?
DFDs (Data Flow Diagrams) visually map out how data flows in a system. They help analysts and developers understand system operations, detect inefficiencies, and design better processes.

---

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

Pros (Desktop)	Cons (Desktop)
Works offline	Limited portability
Better performance	Requires manual updates
Pros (Web)	**Cons (Web)**
Accessible anywhere	Needs internet connection
Easier updates	May be slower

32. How do flowcharts help in programming and system design?
Flowcharts provide a visual overview of logic and processes. They simplify understanding, enhance planning, and help identify problems early in the design phase.
