**MODULE - 1**

Q1. Explain in your own words what a program is and how it functions.

Ans. A program is a set of instruction that is define to our machine What to do and

How to do. And It fuction like Follow a set of rules in a systematic way like follow rules

Step by step.

Q2. What are the key steps involved in the programming process?

Ans. There are many key steps involved in programming process

1. Set of instructions is clear that programmer will understand easily that code and write a code easily.
2. Understand requirnment
3. Plan or implement
4. Test or debug

Q3. What are the main differences between high-level and low-level programming languages?

Ans. 1. Low-level programming is machine dependent language but high level programming is user friendly language.

2. low-level is understood binary language(0s and 1) but high level programming is not understood.

3.example of low level- binary language and example of high level language – Php , c, cpp , html , python.

Q4. Describe the roles of the client and server in web communication?

Ans. In Web communication A client is send a request or asking for a web page to server by Http protocols and Server is Send a response back to user this lifecycycle is called client-server communication.

Q5. Explain the function of the TCP/IP model and its layers?

Ans. Layers –

1. Application layer

Function – provides services to users. Example –HTTP,FTP,SMTP.

1. Transport Layer

Function- Ensures reliable communication between two devices. Example – TCP,UDP.

1. Internet Layer

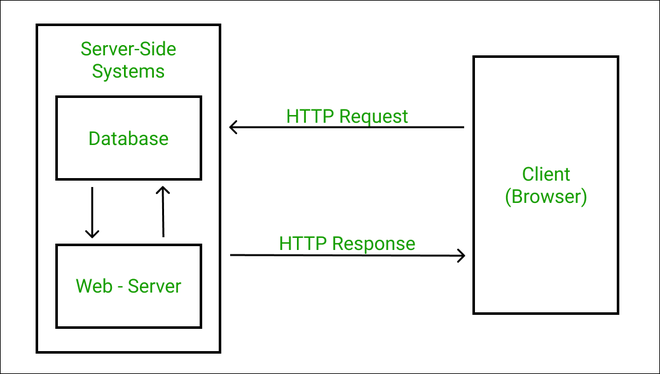
Function – send a data to network via route and provide logical address(IP). Example – IP,ICMP.

1. Network Layer

Function – send a data via internet network aur physical Medium.

Q6. Explain Client Server Communication?

Ans. In Client-Server communication A client is send a request or asking for a web page to server by Http protocols and Server is Send a response back to user this lifecycycle is called client-server communication.



Q7. How does broadband differ from fiber-optic internet?

Ans.

|  |  |
| --- | --- |
| Broadband Fiber optic | |
| General High Speed Internet connection. | It is More Higher than broadband and it is made by fiber optic cables |
| Medium of Broadband is copper wire,Coaxial cable,Wireless signals | Glass/Plastic Fiber cables |
| Speed 10mbps to 500mbps | Speed 100mbps to 10+Gbps |
| Upload speed is slow | Upload and download speed is equal |

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

Ans.

|  |  |
| --- | --- |
| HTTP HTTPS | |
| HTTP refers to Hyper Text Transfer Protocol | HTTP refers to Hyper Text Transfer Protocol Secure |
| It is not secure(data is send in plain text ) | It is Use Encryption to send a data like SSL/TLS |
| Hackers can easily be access | Not easy to access |
| Url http:// | https:// |

Q.9 What is the role of encryption in securing applications?

Ans. Encryption is main role in securing application when we use web application that we can see that in web application there are two type of text normal text and cipher text in normal text(name,password) etc but when this normal text change in cipher text (symbolic and mathamathic algorithm) this cipher text not understable to user that is called encryption and it is best way to secure application.

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

Ans.

|  |  |
| --- | --- |
| Application Softwear System Softwear | |
| Software designed to help the user perform specific tasks or applications. | Software designed to manage and control computer hardware and provide a platform for running applications. |
| To solve particular user problems or perform user-specific tasks. | To run, manage, and control the computer system as a whole. |
| MS Word, Excel, PowerPoint, Photoshop, Web browsers, Games. | Windows, Linux, macOS, Android, Device Drivers, Utility Programs. |
| Cannot run without system software. | Can run without application software. |

Q.11 What is the significance of modularity in software architecture?

Ans.

** Easier maintenance** – changes in one module don’t affect others.

 **Reusability** – modules can be reused in different projects.

 **Scalability** – new features can be added without disturbing the whole system.

 **Improved testing & debugging** – errors are easier to isolate.

 **Team collaboration** – different teams can work on different modules simultaneously.

Q.12 Why are layers important in software architecture?

Ans .

**1. Presentation Layer**

This is what the **user sees** → screens, buttons, forms, design, etc.  
👉 Example: Login page, product page, chat screen.

**2. Application Layer**

This layer works like a **middleman**.  
It takes the request from the presentation layer and sends it to the business layer, then sends the result back.  
👉 Example: When you click “Login,” it sends your details to check if they are correct.

**3. Business Layer**

This is the **brain of the software**.  
All rules, calculations, and decisions happen here.  
👉 Example: Checking if your password is correct, calculating product price, applying discounts.

**4. Persistence Layer**

This layer’s job is to **talk with the database**.  
It decides what data to **save, update, or read**.  
👉 Example: Saving your order details into the database.

**5. Database Layer**

This is the actual place where the **data is stored**.  
👉 Example: Tables where your account, messages, or orders are kept.

Q.13 Explain the importance of a development environment in software production?

Ans.

* Developers get a safe space to do coding and testing.
* Errors and bugs can be found and fixed early.
* It provides useful tools and software (like IDE, debugger, version control).
* It makes teamwork and collaboration easier.
* It helps in improving the quality and consistency of the final product.

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

Ans.

|  |  |
| --- | --- |
| Source code Machine code | |
| Source code contain syntax,instruction or symbols to run code or program | Machine code contain binary language code like (0s and 1s) |
| We can read Source code easily | We can not read Machine code easily |
| To write a source code it have rules or syntax to make a program. | In this code there is not rules or syntax . |
| Ex – Php , java , c , python | Ex- Binary language(0s and 1s) and machine language |

Q.15 Why is version control important in software development?

Ans.

 **Tracks All Changes**  
Version control keeps a complete history of every change made to the code. If something breaks, you can easily go back to a previous working version.

 **Easy Team Collaboration**  
Multiple developers can work on the same project at the same time without overwriting each other’s code.

 **Safe Feature Development**  
You can create separate branches to work on new features or experiments without affecting the main project. Once the feature is ready, it can be merged into the main codebase.

 **Helps with Testing and Experimentation**  
Try out new ideas in a separate environment. If it works, keep it. If not, just delete the branch—no harm done.

 **Clear History of Who Did What**  
Every commit shows who made the change, what was changed, and why—making it easy to track responsibility and understand code changes.

 **Integration with Project Tools**  
Git works well with platforms like GitHub, GitLab, and Bitbucket, making project management, issue tracking, and collaboration even smoother.

Q.16 What are the benefits of using Github for students?

Ans. There are some points that we can say that it is beneficial for students-

* “GitHub is a platform where developers store code, track changes, and collaborate with others. It makes software development easier and more organized.”

**Key Points about GitHub:**

1. **Version Control** – It keeps track of all changes in code, so you can go back to old versions anytime.
2. **Collaboration** – Multiple developers can work on the same project together.
3. **Repositories (Repo)** – A storage place in GitHub where a project’s code and files are kept.
4. **Open Source Projects** – Many free and open-source projects are available on GitHub for learning and contribution.
5. **Features** – Provides Pull Requests, Issues, Wiki, Project Boards, and Actions (for automation).

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

Ans.

|  |  |
| --- | --- |
| Source code Machine code | |
| |  | | --- | | **Source Code** |  |  | | --- | | Available to everyone (can be viewed, modified, and shared). | | |  | | --- | | **Source Code** |   Not available to users (only the company/developer controls it). |
| |  | | --- | | **Cost** |  |  | | --- | | Usually free or very low cost. | | |  | | --- | | **Cost** |   Mostly paid, requires license or subscription. |
| |  | | --- | | **Support** |  |  | | --- | | Community-based support (forums, contributors). | | **Support**  Official company support (helpdesk, updates). |
| |  | | --- | | **Examples** |  |  | | --- | | Linux, Apache, MySQL, Firefox. | | **Examples**  Windows, Microsoft Office, Photoshop. |

Q.18 How does GIT improve collaboration in a software development team?

### Ans.

### How Git improves collaboration in a software development team:

1. **Branching** – Allows multiple developers to work on the same project at the same time without interfering with each other’s code.
2. **Commit History** – Tracks all changes, so everyone knows who made changes, what was changed, and why.
3. **Merge Management** – Helps in resolving code conflicts when combining different changes.
4. **Remote Collaboration** – Supports teamwork through platforms like GitHub, GitLab, and Bitbucket.
5. **Parallel Development** – Enables developers to work on new features, bug fixes, and testing in parallel without affecting the main codebase.

Q.19 What is the role of application software in businesses?

Ans.

**Task Management** – Helps in daily business tasks like accounting, data handling, and documentation.

**Efficiency** – Saves time and increases productivity.

**Decision Support** – Provides reports and insights for better decision-making.

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

### Ans.

### Main Stages of Software Development Process (SDLC)

1. **Requirement Gathering** – Collecting information about what the software should do.
2. **Requirement Analysis** – Studying and understanding the requirements in detail.
3. **Designing** – Creating system and software design (architecture, UI, database design).
4. **Implementation (Coding)** – Writing the actual program/code.
5. **Testing** – Checking the software for errors, bugs, and quality.
6. **Deployment & Maintenance** – Delivering the software to users and updating/fixing it when needed.

Q.21 Why is the requirement analysis phase critical in software development?

Ans. T**he requirement analysis phase is critical because it clearly defines what the customer needs, helps avoid misunderstandings, and ensures the software is built in the right way. If this step is wrong, the whole project can fail.**

**Q.22** What is the role of software analysis in the development process?

**Ans.** The role of software analysis is to study and understand user requirements, identify problems, and define what the software should do. It sets a clear foundation for design and development.

**Q.23** What are the key elements of system design?

Ans.

**Key elements of system design:**

1. **Architecture & Database Design** – Defines overall structure and data storage.
2. **Interface Design** – Focuses on user interface and user interaction.
3. **Security & Module Design** – Ensures system safety and breaks the system into manageable parts.

**Q.24** Why is software testing important?

Ans. **Software testing is important because it finds errors and bugs, ensures the software works correctly, and improves its quality and reliability before release.**

**Q.25** What types of software maintenance are there?

Ans.

 **Corrective Maintenance** – Fixing errors and bugs.

 **Adaptive Maintenance** – Updating software for new environments (OS, hardware, etc.).

 **Perfective Maintenance** – Improving performance or adding new features.

 **Preventive Maintenance** – Making changes to avoid future problems.

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

Ans.

|  |  |
| --- | --- |
| Web application Desktop application | |
| Runs in a web browser, no installation needed. | Needs to be installed on the computer. |
| Can be accessed from anywhere with internet. | Accessible only on the system where it is installed. |
| Easy to update (server-side). | Updates must be installed on each device. |
| Gmail, Facebook, Google Docs | MS Word, Photoshop, VLC Media Player |

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

Ans.

* **No Installation Needed** – Runs directly in a browser.
* **Easy Access** – Can be used from anywhere with internet.
* **Automatic Updates** – Always up to date without manual installation.

Q.28 What role does UI/UX design play in application development?

Ans. UI/UX design plays a key role in making an application easy to use, visually appealing, and user-friendly, which improves user satisfaction and overall success of the app.

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

Ans.

| **Feature** | **Native Apps** | **Hybrid Apps** |
| --- | --- | --- |
| **Platform** | Built for a specific platform (Android, iOS). | Single app works on multiple platforms. |
| **Performance** | Faster and more responsive. | Slightly slower due to web wrapper. |
| **Access to Device Features** | Full access to device features (camera, GPS, etc.). | Limited access, depends on framework. |
| **Development Cost** | More expensive (separate code for each platform). | Cheaper (one codebase for all). |
| **User Experience** | Better UI/UX, feels smooth and native. | UI/UX may not be as smooth as native. |
| **Examples** | WhatsApp, Instagram, Spotify | Facebook, Uber, Gmail |

Q.30 What is the significance of DFDs in system analysis?

Ans. The significance of DFDs (Data Flow Diagrams) in system analysis is that they show how data moves through a system, making it easier to understand processes, inputs, and outputs. They help in clear communication between developers and users.

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

Ans. Pros of Desktop Applications:

* Faster performance (runs on system hardware).
* Works without internet.
* Can use full system resources (GPU, storage, etc.).

Cons of Desktop Applications:

* Needs installation on each device.
* Platform-dependent (Windows, macOS, etc.).
* Updates must be installed manually.

Q.32 How do flowcharts help in programming and system design?

Ans. **Flowcharts help in programming and system design by showing the steps of a process visually, making it easier to understand logic, detect errors, and communicate ideas clearly.**