Python-Full Stack Assignment

Module 1-Overview of IT Industry

**What is Program?**

* Program is a set of Instruction.

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

* A Program is a set of instruction that a computer follow to perform a specific task. Where the ingredients are data and the step are the instructions.

**What is Programming?**

* Programming is the process of creating a set of instruction.

**What are the key steps involved in the programming process?**

* In Programming process there are six a types of steps :

1. Problem Definitions
2. Planning and Design
3. Coding
4. Testing and Debugging
5. Documentation
6. Maintenance

**Types of Programming Languages?**

* Procedural ,Programming
* Ex.-C language
* Object Oriented Programming

Ex.-C++ language

* Logical Programming

Ex.-Prolog language

* Functional Programming

Ex.-Python language

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

* High-level Programming Language:
* High-level Language are designed to be human-readable and easy to understand.
* Ex. Python, Java , C++,C#, JavaScript , Ruby.
* Low-Level Programming Language:
* Low-level Language are closer to the machine code that the computer directly understand.
* Ex. Assembly Language , Machine code.

**World Wide Web & How Internet Works?**

* WWW: This is like the specific content accessible over the Internet.
* Internet: Imagine it as a vast network of interconnected highways.

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

* The Client starts the communication by sending a request to the server.
* It receives the response from the server and displays it to the user.
* **Server**:
* The server waits for requests from clients. When a request arrives, it processes it .
* After processing the request, the server sends a response back to the client, containing the requested data or information.

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

* The TCP/IP Model: A Foundation for Network Communication.
* The TCCP/IP model is a conceptual framework that defines how data is transmitted over networks.
* There are Four types of Layers:

1.Application Layer

2.Transport Layer

3.Internet Layer

4.Network Interface Layer

**Explain Client Server Communication**

Client-server communication is a fundamental model in networking where a client requests services from a server.

* This Model underpins many internet connections.
* Here are some key Types:

1.Web Browsing (HTTP/HTTPS):

- Client: Web Browser (Ex. Chrome, Firefox )

- Server: Web Server (Ex. Apache, Nginx)

- Communication: Client requests a web page (Ex. https.//www.google.co.uk/).

2.Email (SMTP, POP3, IMAP):

- Client: Email client (Ex. Outlook, Gmail)

- Server:

SMTP (Simple Mail Transfer Protocol)

POP3 (Post Office Protocol)

IMAP (Internet Message Access Protocol)

3. File Transfer (FTP):

- client: FTP client (Ex. FileZilla)

- Server: FTP server

- Communication: Client connects to the FTP server.

4. Database Access(SQL):

- Client: Application (Ex. Database management tool, web application)

- Server: Executes the queries and returns the results to client.

**Types of Internet Connections.**

There are several types of internet connections, including:

* DSL: A fixed connection that uses copper phone lines to send and receive internet data.
* Fiber: A high-speed connection that uses glass or plastic strands to transmit data using light pulses.
* Cable: A type of internet connection.
* Wi-Fi: A wireless connection that uses electromagnetic radiation to connect devices to the internet.
* Dial-up: A type of internet connection.
* Cellular technology: A type of internet connection.
* Integrated Services Digital Network (ISDN): A type of internet connection.

**How does broadband differ from fiber-optic internet? Protocols**

Broadband is a general term for high-speed internet access.

Fiber-optic internet is a specific type of broadband that uses thin strands of glass or plastic to transmit data via pulses of light.

**What are the differences between HTTP and HTTPS protocols? Application Security**

The key difference between HTTP and HTTPS is that HTTP transmits data in plain text, making it unsecure, while HTTPS encrypts data using TLS (Transport Layer Security) .

**What is the role of encryption in securing applications?**

Encryption plays a critical role in securing applications by transforming sensitive data into an unreadable format

**Software Applications and Its Types**

Application Software is a type of computer program that performs specific functions. These functions, performed by application software, can be personal, business as well as educational. Thus, application Software is also known as end-user software or productivity software.

**What is the difference between system software and application software?**

System software is designed to manage a computer's hardware and provide a platform for other programs to run on, essentially acting as the foundation for the system, while application software is designed to perform specific tasks directly for the user, like word processing or web browsing, and relies on the system software to function properly.

**What is the significance of modularity in software architecture?**

Modularity in software architecture is important because it makes software more organized, easier to maintain, and more flexible

Benefits of modularity

* **Improved design**
* **Better error handling**
* **Increased flexibility**
* **Improved scalability**
* **Increased reusability**
* **Improved collaboration**
* **Reduced complexity**

**Why are layers important in software architecture?**

Layers are crucial in software architecture because they promote "separation of concerns," allowing each layer to focus on a specific set of functionalities, making the code easier to understand, maintain, test, and modify independently, ultimately leading to a more robust and scalable application.

**Software Environments**

A "Software Environment" refers to a collection of tools, programs, and configurations that developers use to build and run software applications, including the operating system, database systems, specific development tools, compilers, and libraries, essentially creating a specific setup to work on a project at different stages of development like testing or deployment

**Explain the importance of a development environment in software production.**

A development environment is important in software production because it allows developers to test and refine code before releasing it to users. It helps ensure that the software is high quality and meets user expectations.

**Source Code**

Source code is generally understood to mean programming statements that are created by a programmer with a text editor or a visual programming tool and then saved in a file.

**What is the difference between source code and machine code?**

Source code is the human-readable code written by a programmer in a programming language, while machine code is the low-level binary representation of that code which a computer's CPU can directly understand and execute

**Github and Introductions**

GitHub is a web-based platform that uses Git to help developers manage and track code changes. It allows multiple people to collaborate on projects, track revisions, and contribute to code.

**Why is version control important in software development?**

Version control is important in software development because it allows developers to manage and track changes to code, and collaborate on projects simultaneously.

Benefits of version control

* **Collaboration**
* **Faster development**
* **Bug fixing**
* **Project progress tracking**

**Student Account in Github**

A GitHub Student Account is a free account that gives students access to professional developer tools and resources. The account is part of GitHub Education, a portal that provides students with benefits and resources to help them learn and develop their skills.

**What are the benefits of using Github forstudents?**

GitHub can help students learn to code, collaborate, and build projects. Benefits include:

* **Access to premium tools**
* **Real-world experience**
* **Collaboration**
* **Learning resources**
* **Code review**
* **Automating tasks**
* **Access to open-source code**
* **Streamlined autograding**

**Types of Software**

There are many types of software, including operating systems, system software, application software, programming software, and more.

Operating systems

* Manage resources and provide services for other software

System software

* Acts as an intermediary between hardware and application software

Application software

* Created for end users

Programming software

* Created for computer developers and programmers to produce code

Freeware

* Small applications that can be downloaded and used for free on most operating systems

Graphics software

* Helps users edit or change visual data or images

Spreadsheet software

* Enables users to perform numerical functions and explore numbers

Closed source software

* Software for which the source code is not freely available

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

Open Source software is the software that is *available to users with source code*. Source code is a part of a program or software. Users can *modify, inspect and enhance it*to improve the software. Users prefer open source software because of following reasons-

* More control over the software
* More secure
* Stable
* High quality results
* Helps in becoming a better programmer as you can learn and develop from the source code to make new softwares.

**Proprietary Software:** Proprietary software is computer software where the source codes are publicly not available only the company that has created them can modify it. Here the software is developed and tested by the individual or organization by which it is owned not by the public. Number of installations of this software into computers

* Restrictions on sharing of software illegally
* Time period up to which software will operate
* Number of features allowed to use

**GIT and GITHUB Training**

Git and GitHub training is available online from a number of sources, including Udemy, TheServerSide, and W3Schools.

* **Udemy**: Offers a course called "Become an Expert in Git & GitHub in 4 Hours" that covers Git theory, commands, and real-world applications
* **TheServerSide**: Offers a full tutorial for beginners on Git and GitHub
* **W3Schools**: Offers an introduction to Git and GitHub

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

Git enhances collaboration in a software development team by allowing multiple developers to work on the same project simultaneously on different branches, preventing accidental overwriting of each other's changes.

Key features of Git that improve collaboration:

* **Branching system**
* **Pull Requests**
* **Version control**
* **Distributed nature**
* **Commit history**

**Application Software**

Application software, also known as an app, is a computer program that helps users perform specific tasks. These tasks can be personal, educational, or business-related.

**What is the role of application software in businesses?**

Application software plays a critical role in businesses by enabling them to perform specific tasks, manage data efficiently, automate processes, facilitate communication, and ultimately improve productivity across various departments.

Key functions of application software in businesses:

* **Data management**
* **Process automation**
* **Communication tools**
* **Customer relationship management (CRM)**
* **Financial management**
* **Project management**
* **Human resource management (HRM)**
* **Supply chain management**

**Software Development Process**

The software development process is a series of steps that are followed to design, create, test, and maintain software applications. It's also known as the software development life cycle (SDLC).

**What are the main stages of the software development process?**

The main stages of the software development process are planning, design, development, testing, deployment, and maintenance. These stages are known as the Software Development Life Cycle (SDLC).

**Software Requirement**

A software requirement is a description of what a software system should do and how it should behave. It also specifies the constraints that the software must meet.

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

he requirement analysis phase is critical in software development because it establishes a clear understanding of what the software should achieve by identifying and documenting the needs and expectations of all stakeholders.

Key reasons why requirement analysis is critical:

* **Reduces Scope Creep**
* **Improves Communication**
* **Accurate Project Estimation**
* **Identifies Potential Issues**
* **Quality Assurance**
* **User Satisfaction**

**Software Analysis**

Software analysis is the process of identifying the requirements needed to create a fully functional software product. It involves analyzing stakeholder requirements, documenting and validating software requirements, and identifying potential issues.

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

Software analysis is a crucial step in the software development process that helps ensure the software meets the needs of the users and stakeholders

**System Design**

Systems Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

**What are the key elements of system design?**

The process of specifying a computer system’s architecture, components, modules, interfaces, and data is known as [system design](https://www.geeksforgeeks.org/what-is-system-design-learn-system-design/).

**Software Testing**

Software testing is the process of evaluating a software program to ensure it works as expected. It involves running the software in different environments and scenarios to identify any defects.

**Why is software testing important?**

Software testing is important because it ensures that software is reliable, secure, and performs as expected. It also helps identify and fix issues early in the development process, which can save time and money.

**Maintenance**

Maintenance is the act of keeping something in good condition, such as equipment, machinery, or a facility. It can also refer to the upkeep of property.

**What types of software maintenance are there?**

There are four types of software maintenance:

* Corrective Software Maintenance
* Adaptive Software Maintenance
* Perfective Software Maintenance
* Preventive Software Maintenance

**Corrective Software Maintenance**

Corrective software maintenance is what one would typically associate with the maintenance of any kind. Correct software maintenance addresses the errors and faults within software applications that could impact various parts of your software, including the design, logic, and code.

**Adaptive Software Maintenance**

Adaptive software maintenance becomes important when the environment of your software changes. This can be brought on by changes to the operating system, hardware, software dependencies, Cloud storage.

**Perfective Software Maintenance**

Perfective software maintenance focuses on the evolution of requirements and features that existing in your system. As users interact with your applications, they may notice things that you did not or suggest new features that they would like as part of the software.

**Preventive Software Maintenance**

Preventative Software Maintenance helps to make changes and adaptations to your software so that it can work for a longer period of time. The focus of the type of maintenance is to prevent the deterioration of your software as it continues to adapt and change.

**Development**

Development is defined as a process that results in growth, progress, positive change, or the addition of physical, economic, environmental, social, and demographic components to an existing system or environment.

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

The key difference between web and desktop applications is that web applications are accessed through a web browser and can be used on any internet-connected device, while desktop applications are installed directly on a user's computer .

**Web Application**

A web application, or web app, is a software program that runs on a web browser and is accessed over the internet. Web apps are coded in a browser-supported language like HTML, JavaScript, or Java.

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

The main advantages of using a web application over a desktop application are: no installation required, accessibility from any device with an internet connection, automatic updates, cross-platform compatibility, easier maintenance, and scalability; essentially, web apps are more convenient to access and update, and can be used across different devices without needing to be installed on each one individually.

Key benefits of web applications:

* **Platform independence**
* **No installation needed**
* **Automatic updates**
* **Accessibility from anywhere**
* **Easier maintenance**
* **Scalability**
* **Cost-effective development**

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

UI/UX design plays a fundamental role in the success of any application. It's not just about making things look pretty; it's about creating a seamless and enjoyable experience for the user.

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

When considering mobile app development, the choice between native and hybrid apps is a crucial one.

**What is the significance of DFDs in system analysis?**

Data Flow Diagrams (DFDs) are a vital tool in system analysis, playing a significant role in understanding and visualizing how data moves through a system.

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

When comparing desktop applications and web applications, it's essential to consider the trade-offs between them.

**How do flowcharts help in programming and system design?**

Flowcharts are incredibly valuable tools in both programming and system design, offering a visual representation of processes that can greatly enhance understanding and efficiency.