**Overview of IT Industry**

1. **What is Program**

A program is a set of instructions that computer uses to perform a specific function. To use an analogy, a program is like a computer’s recipe. It contains a list of ingredients (called variables, which can represent numeric data, text, or images) and list of directions (called statements) that tell the computer how to execute a specific task.

1. **What is programming?**

Programming refers to a technological process for telling a computer which tasks to perform in order to solve problems. You can think of programming as a collaboration between humans and computers, in which humans create instructions for a computer to follow (code) in a language computers can understand.

1. **Types of Programming Language**

There are three main kinds of programming language:

Machine language

Assembly language

High-level language

1. **World Wide Web**

The World Wide Web (WWW), commonly known as the Web, is an information system that enables information sharing over the Internet through user-friendly ways meant to appeal to users beyond IT specialists and hobbyists. It allows documents and other web resources to be accessed over the Internet according to specific rules of the Hypertext Transfer Protocol (HTTP).

1. **How Internet Works**

Generally, two main components uphold the functionality of the Internet, they are:

Packets

Protocols

In networking, the data which is being transmitted through the internet is sent via small segments/chunks which are later translated into bits, and the packets get routed to their endpoint (destination) through different networking devices i.e. routers or switches. Later, once the packet arrives at the receiver’s end, that small chunks of data get reassembled in order to utilize or check the data that he/she requested. That’s why they are used to push ease in networking and large data can be easily sent by sending small units and this whole process of sending/receiving small bits is known as Packet Switching.

1. **Client and Server**

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.

A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

1. **Types of Internet Connections**

Dial-up connection

ISDN

Leased Line or TI/T3 connections

DSL

Cable TV Internet connections

Satellite Internet Connections

Wireless Internet Connections

1. **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.

Each software program is developed to assist users with the particular process related to productivity, efficiency, and communication. Unlike System Software, Application Software is specific for its functionality and completes the task that they are developed to do. The majority of apps that we see on our smartphones are examples of application software.

Types of Application Software:-

Web browsers

Presentation software

Spreadsheet software

Graphic software

Word processors

Database software

Multimedia software

Education software

Information software

Content access software

1. **Software Architecture**

Architecture serves as a blueprint for a system. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components.

It defines a structured solution to meet all the technical and operational requirements, while optimizing the common quality attributes like performance and security.

Further, it involves a set of significant decisions about the organization related to software development and each of these decisions can have a considerable impact on quality, maintainability, performance, and the overall success of the final product. These decisions comprise of

Selection of structural elements and their interfaces by which the system is composed.

Behavior as specified in collaborations among those elements.

Composition of these structural and behavioral elements into large subsystem.

Architectural decisions align with business objectives.

Architectural styles guide the organization.

1. **Software development process**

The Software Development Process is the structured approach to developing software for a system or project, sometimes called the Software Development Life Cycle (SDLC). There are several approaches (see Software Development Approaches) that can be used to include waterfall, spiral, and incremental development. These different approaches will focus the testing effort at different points in the development process. However, each approach is composed of the same basic steps of development. The incremental development approach typically forms the basis for software development within the larger systems level of Evolutionary Acquisition (EA)

1. **Software Requirement**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

It is a four step process, which includes –

Feasibility Study

Requirement Gathering

Software Requirement Specification

Software Requirement Validation

1. **Software Analysis**

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

1. **Software Design**

Software design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the client’s requirement, as described in SRS (Software Requirement Specification) document, into a form, i.e., easily implementable using programming language.

The software design phase is the first step in SDLC (Software Design Life Cycle), which moves the concentration from the problem domain to the solution domain. In software design, we consider the system to be a set of components or modules with clearly defined behaviors & boundaries.

1. **Software Testing**

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

Software testing can be divided into two steps:

1. Verification: it refers to the set of tasks that ensure that the software correctly implements a specific function.
2. Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

1. **Software Maintenance**

Software Maintenance refers to the process of modifying and updating a software system after it has been delivered to the customer. This can include fixing bugs, adding new features, improving performance, or updating the software to work with new hardware or software systems. The goal of software maintenance is to keep the software system working correctly, efficiently, and securely, and to ensure that it continues to meet the needs of the users.

Software maintenance is a continuous process that occurs throughout the entire life cycle of the software system. It is important to have a well-defined maintenance process in place, which includes testing and validation, version control, and communication with stakeholders.

1. **Mobile application**

A mobile app (or mobile application) is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers.

Mobile apps are sometimes categorized according to whether they are web-based or native apps, which are created specifically for a given platform. A third category, hybrid apps, combines elements of both native and web apps.

In today’s digital age, mobile apps are an essential part of most people’s daily lives. From social networking and entertainment to productivity and business, mobile apps play a vital role in how we interact with technology.

1. **Desktop Application**

Desktop applications are programs designed to run on computers and use system resources to perform their functions. These programs run on top of operating systems such as Linux, Microsoft Windows, and macOS.

Some top examples of desktop applications include Microsoft Office, VLC Media Player, Windows File Explorer, and Activity Monitor on macOS.

1. **Flow Chart**

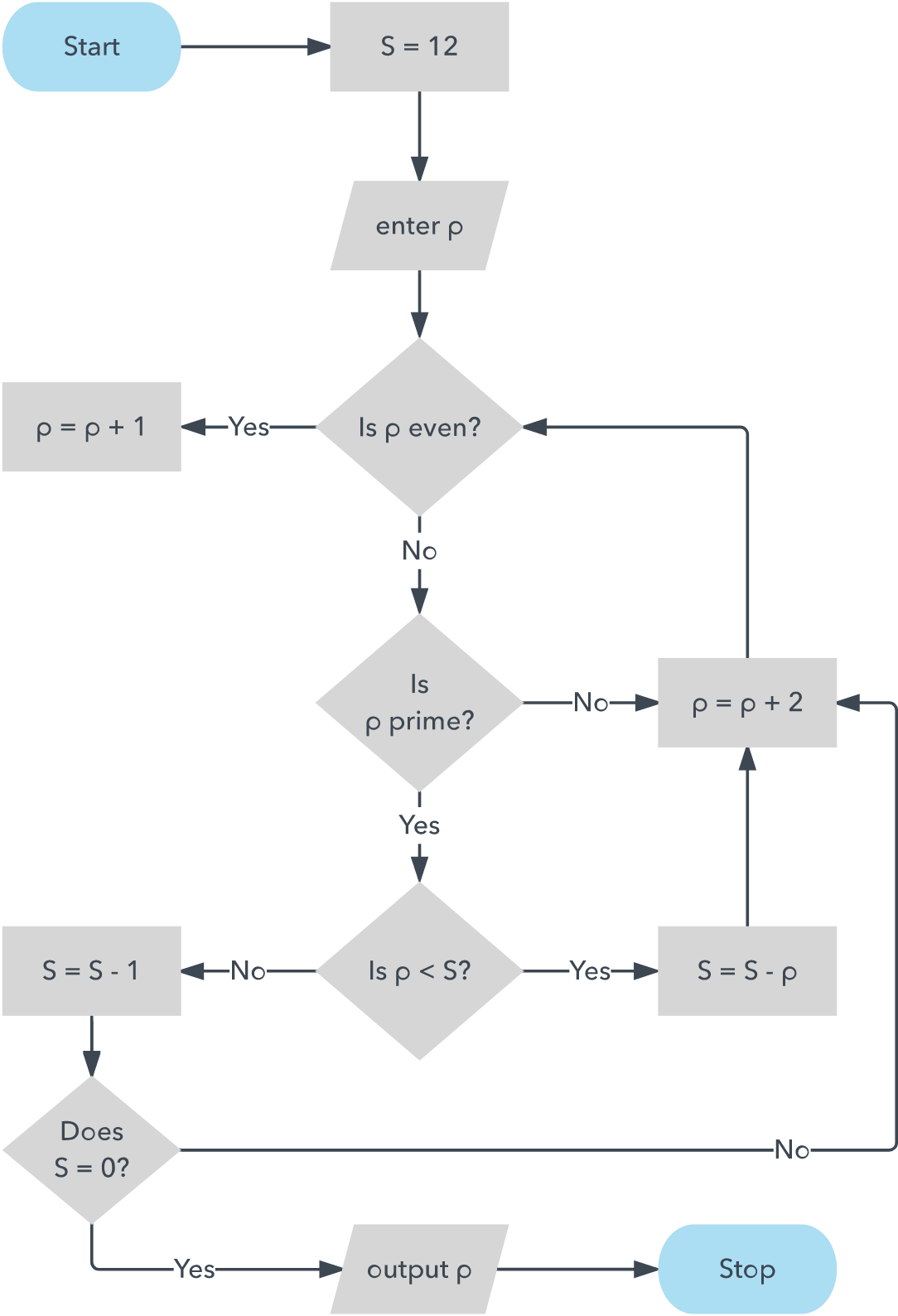
A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams.

Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.

If we consider all the various forms of flowcharts, they are one of the most common diagrams on the planet, used by both technical and non-technical people in numerous fields.

Flowcharts are sometimes called by more specialized names such as Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, Business Process Modeling and Notation (BPMN), or Process Flow Diagram (PFD). They are related to other popular diagrams, such as Data Flow Diagrams (DFDs) and Unified Modeling Language (UML) Activity Diagrams.

****