TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Software Engineering Course

Content

Module 1 - [Overview of IT Industry]

Module 2 - [Fundamentals of Programing]

Module 3 - [OOP Concept]

Module 4 - [HTML & CSS]

Module 5 - [Database]

Module - 1 [Overview of IT Industry]

What is Program

- Program- It is a set of Instructions

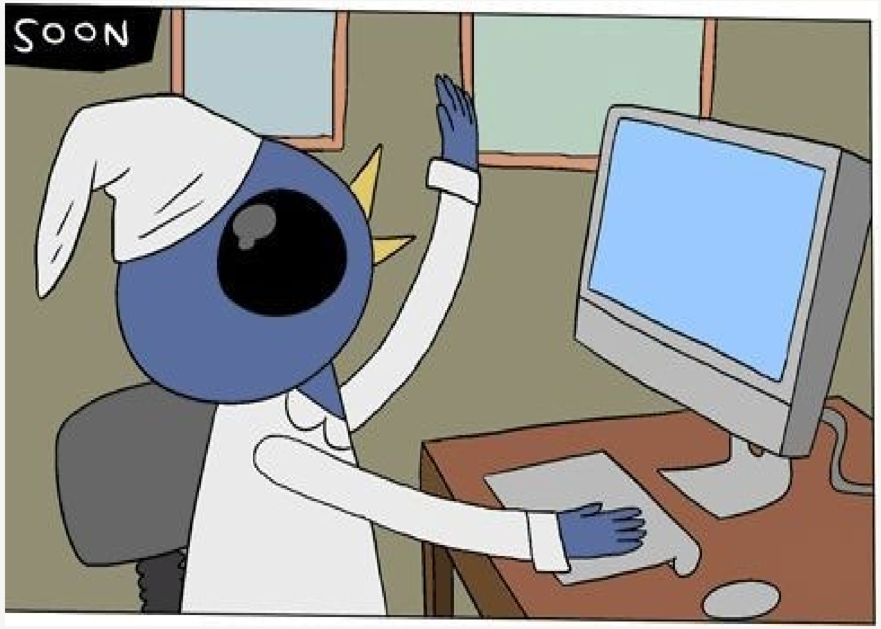
```
>Hello world!
>_
```

Ex. 1- Instructions to your Pet Ex. 2- Starting your Computer

What is Programming

- Programming- To create a Program.

Ex. 1- Using Keyboard & Mouse



Types of Programming Language

- Procedural Programming
  - Ex.- C Language

Types of Programming Language

- Object Oriented Programming
  - Ex.- C++ Language

Types of Programming Language

- Logical Programming
  - Ex.- Prolog Language

Types of Programming Language

- Functional Programming
  - Ex.- Python Language

World Wide Web - WWW

- known as a Web
- is a collection of websites or web pages stored in web servers
- connected to local computers through the internet
- These websites contain text pages, digital images, audios, videos, etc.
- Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc.
- The WWW, along with internet, enables the retrieval and display of text and media to your device.

WWW – How Internet Works?

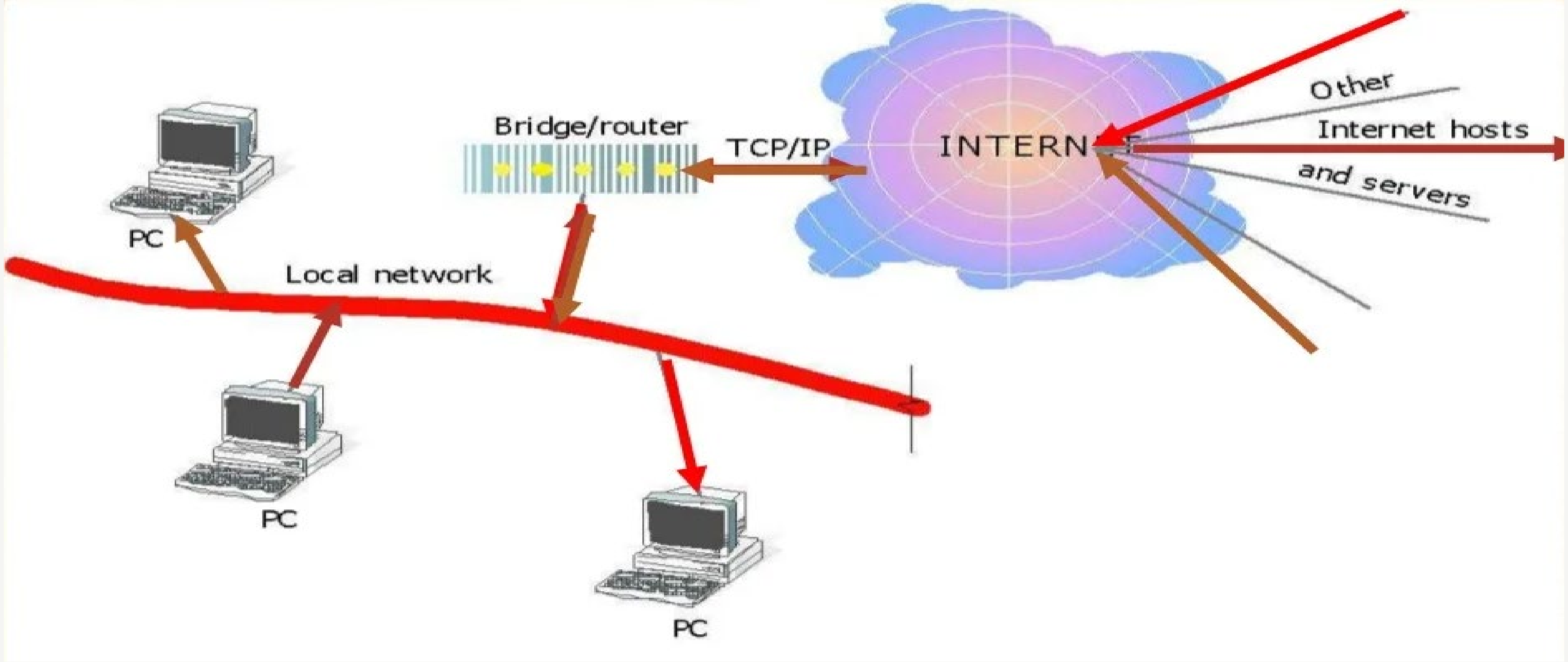
- A server is where websites are stored, and it works a lot like your computer's hard drive.

- Once the request arrives, the server retrieves the website and sends the correct data back to your computer.

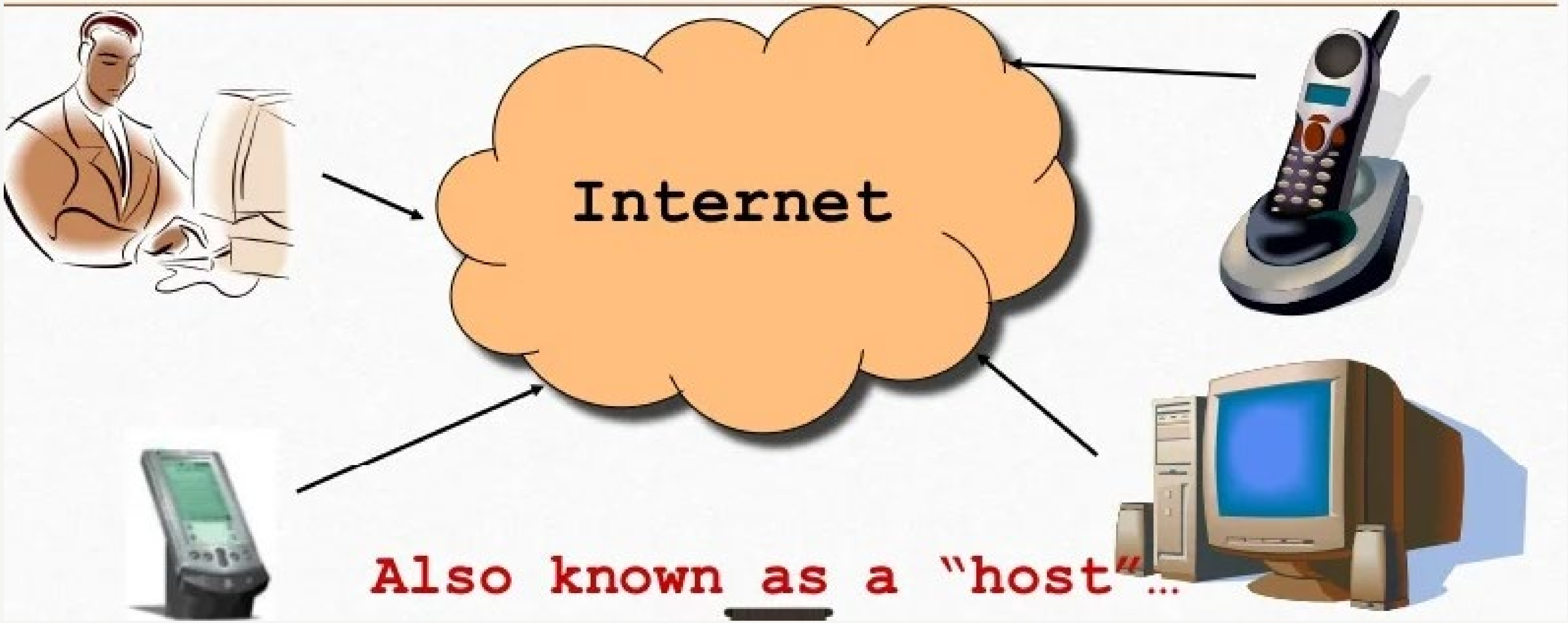
- One of the best features of the Internet is the ability to communicate almost instantly with anyone in the world.

- **Email** is one of the oldest and most universal ways to communicate and share information on the Internet, and billions of people use it. **Social media** allows people to connect in a variety of ways and build communities online.

WWW – How Internet Works?



Network layers on Client & Server

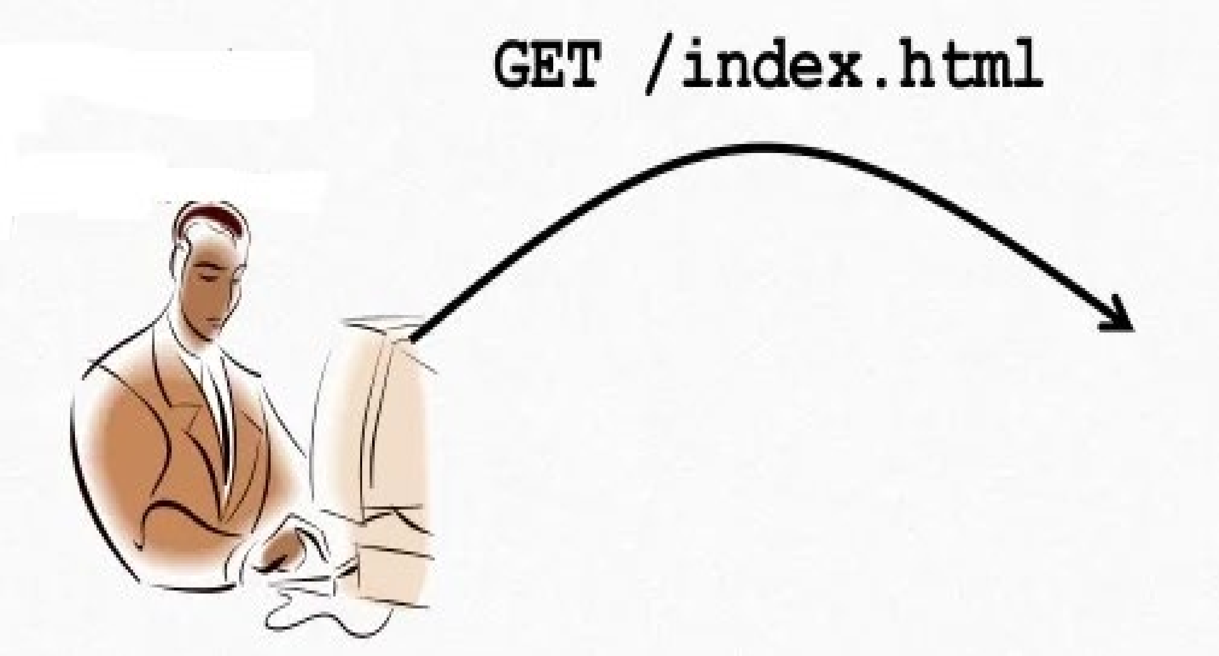


Also known as a "host"…

Client & Servers

1. **Client Program**

- Running on end host
- Requests Services
- E.g. Web Browser



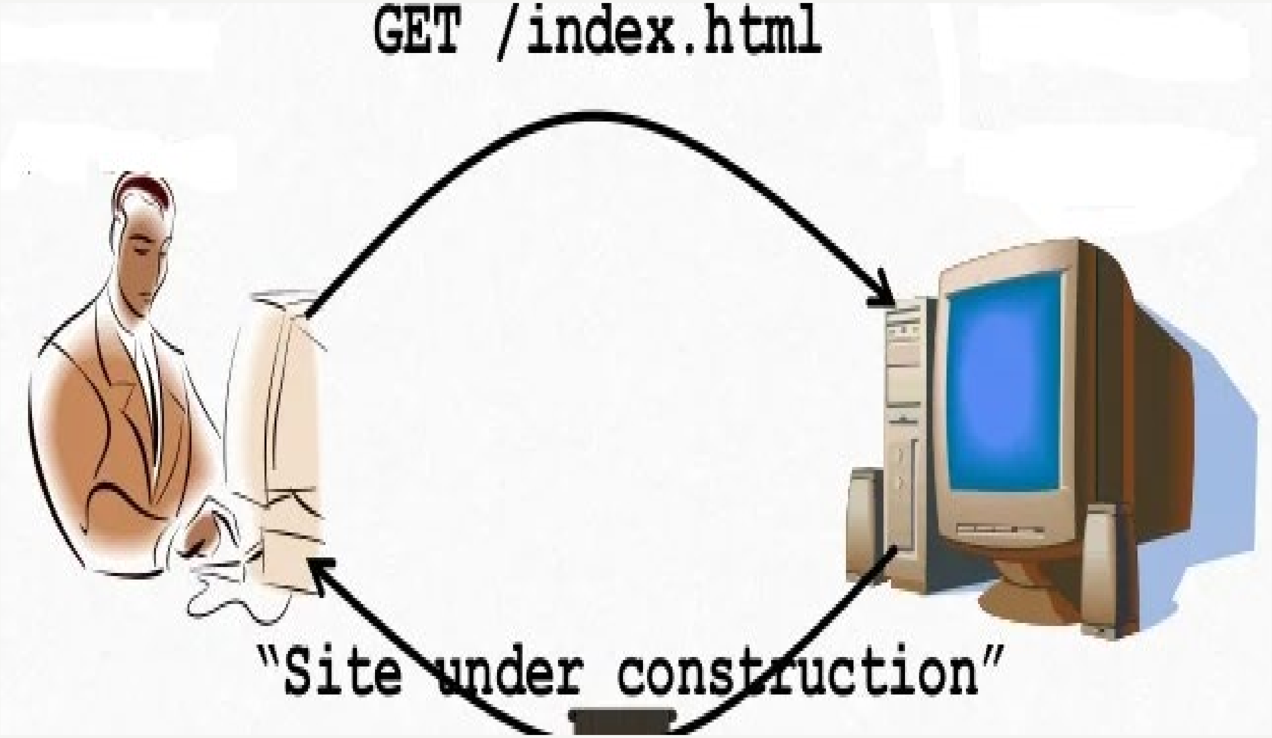
Client & Servers

1. Client Program

- Running on end host
- Requests Services
- E.g. Web Browser

2. Server Program

- Running on end host
- Provides Services
- E.g. Webserver



Client – Servers Communication

1. **Client “sometimes on”**

- Initiates a request to the server when interested
- E.g. Web Browser on your laptop or cell phone
- Doesn’t communicate directly with other clients
- Needs to know the server’s address

2. Server is “always on”

- Services requests from many client hosts
- E.g.Web Server for the www.example.com web site
- Doesn’t initiate contract with the clients
- Needs a fixed, well-known address

Types of Internet connection

1. Digital subscriber line(DSL)
2. Cable Internet
3. Fiber Optic
4. Satellite Internet
5. Wireless
6. Broadband over Power lines(BPL)

Major Applications of the Internet

- 4 major applications of internet, which are given below.

1. Social Media Internet Applications

    e.g .Facebook,Instagram,Twitter,LinkedIn…

2. Communications Internet Applications

    e.g. Email,Skype,Zoom,Whatsapp…

3. Entertainment Internet Applications

    e.g. Netflix,Hotstar,Youtube, Amazon prime video…

4. Travel Internet Applications

    e.g Google Trips,Google Map,trivago…

What are protocols?

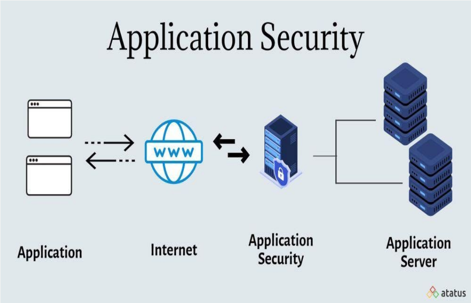
- A Network Protocol is a group of rules accompanied by the network.
- Network protocols will be formalized requirements and plans composed of rules, procedures, and types that describe communication among a couple of devices over the network.
- The protocol can be described as an approach to rules that enable a couple of entities of a communication program to transfer information through any type of variety of a physical medium.
- The protocol identifies the rules, syntax, semantics, and synchronization of communication and feasible error managing methods. In this article, we will discuss the different types of networking protocols.

Types of Protocols

1. HTTP or HTTPS
2. FTP(File Transfer protocols)
3. Email Protocols(POP3,SMTP)
4. TCP(Transmission control protocol) and UDP(User Datagram Protocol)

What is Application Security?

- Application security refers to security precautions used at the application level to prevent the theft or hijacking of data or code within the application.
- It includes security concerns made during application development and design, as well as methods and procedures for protecting applications once they've been deployed.
- All tasks that introduce a secure software development life cycle to development teams are included in application security shortly known as AppSec.
- Its ultimate purpose is to improve security practices and, as a result, detect, repair, and, ideally, avoid security flaws in applications.
- It covers the entire application life cycle, including requirements analysis, design, implementation, testing, and maintenance..



What is Application Security?

- All tasks that introduce a secure software development life cycle to development teams are included in application security shortly known as AppSec.
- Its ultimate purpose is to improve security practices and, as a result, detect, repair, and, ideally, avoid security flaws in applications.
- It covers the entire application life cycle, including requirements analysis, design, implementation, testing, and maintenance..

Software Applications

What is Application Software :

- It is a type of software application that helps in the automation of the task based on the Users Input.
- It can perform single or multiple tasks at the same period of time.
- There are the different application which helps us in our daily life to process our instructions based on certain rules and regulations.
- Application Software helps in providing a graphical user interface to the user to operate the computer for different functionality.
- The user may use the computer for browsing the internet, accessing to email service, attending meetings, and playing games.
- Different high-level languages are used to build application software.

Types of Application Software

- Application software
- System software
- Driver software
- Middleware
- Programming software

Application Software

- The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application.
- An application can be self-contained, or it can be a group of programs that run the application for the user.
- Examples of Modern Applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

Example:Microsoft Office, Paint, Powerpoint etc..

System Software

- These software programs are designed to run a computer's application programs and hardware.
- - System software coordinates the activities and functions of the hardware and software.
- It controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in.
- The OS is the best example of system software; it manages all the other computer programs.
- Other examples of system software include the firmware, computer language translators and system utilities..

Example:Notepad ,Calculator etc..

Driver Software

- Also known as device drivers, this software is often considered a type of system software.
- Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks.
- Every device that is connected to a computer needs at least one device driver to function.
- Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.

Example: Audio Driver,Video Driver etc..

Middleware

- The term *middleware* describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word.
- It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

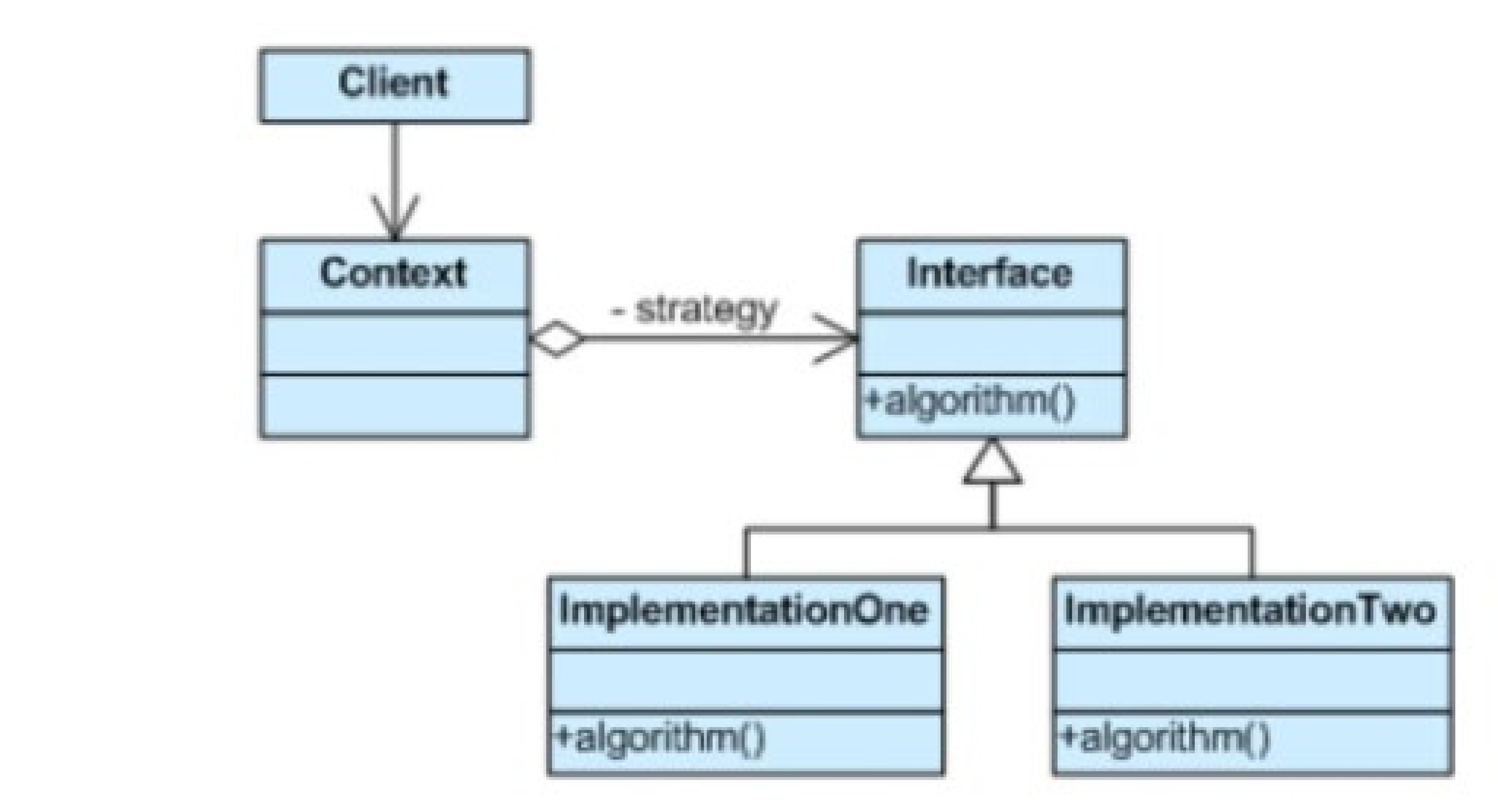
Example: database middleware,application server middleware

Programming Software

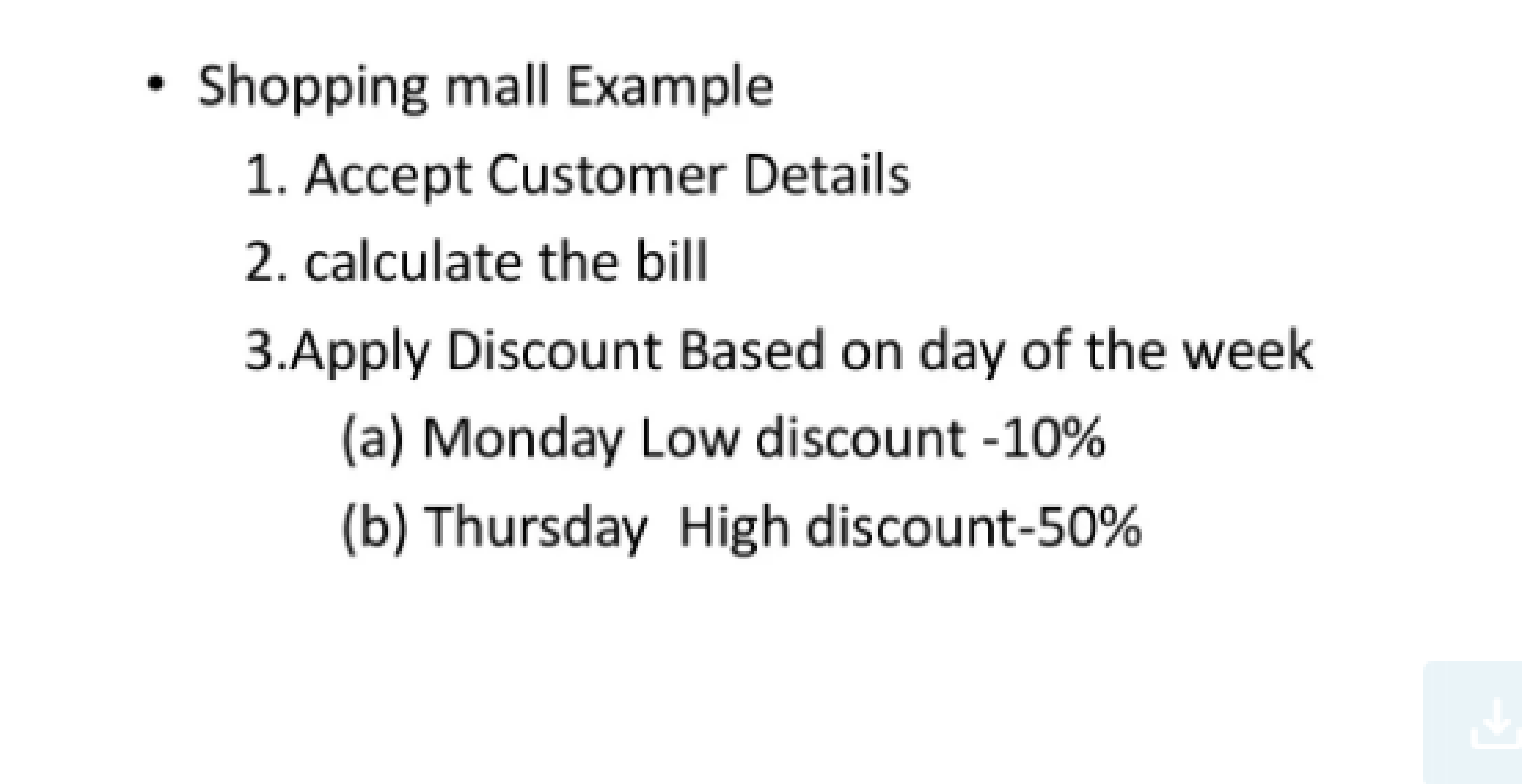
- Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs.
- Examples of programming software include assemblers, compilers, debuggers and interpreters.

Examples : Turbo c,Eclipse,Sublime etc..

Architecture



Applicaton and Examples



- Shopping mall Example

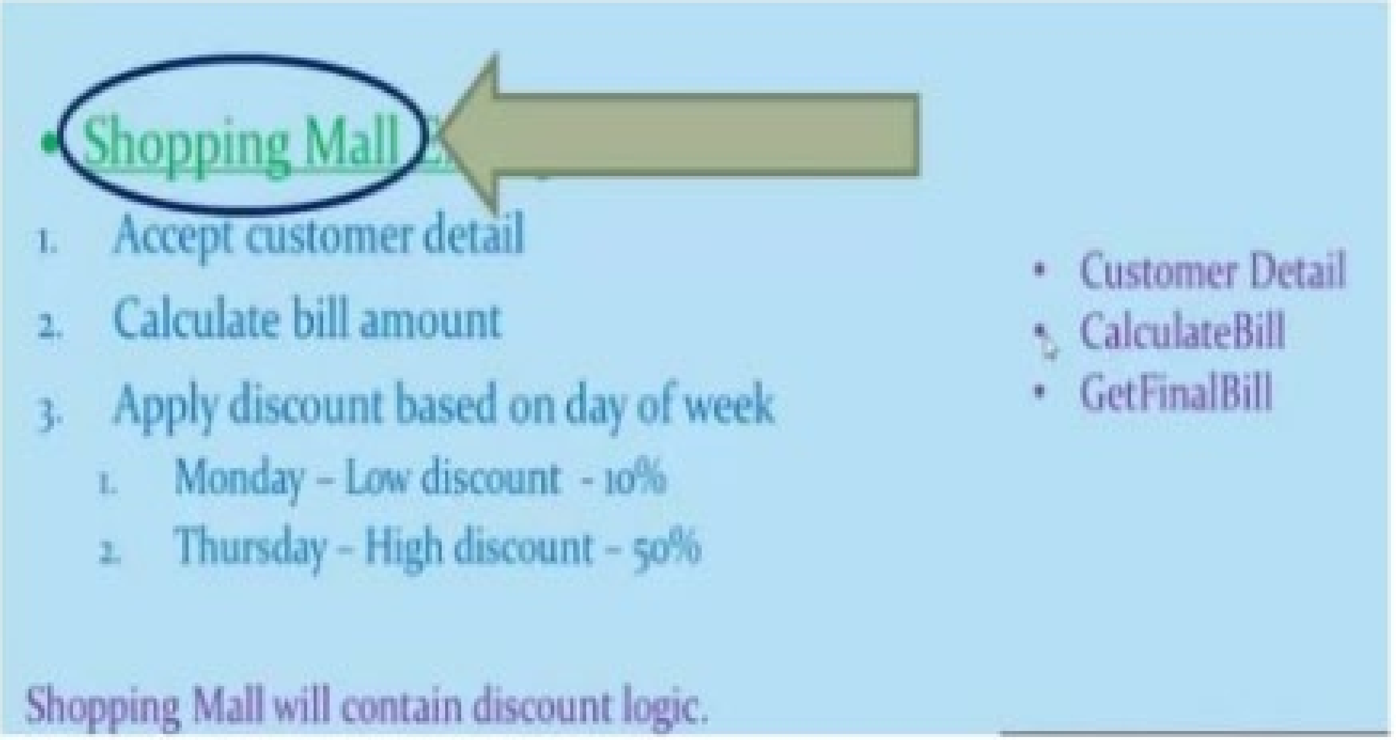
  1. Accept Customer Details

  2. calculate the bill

  3.Apply Discount Based on day of the week

     (a) Monday Low discount -10%

     (b) Thursday High discount-50%

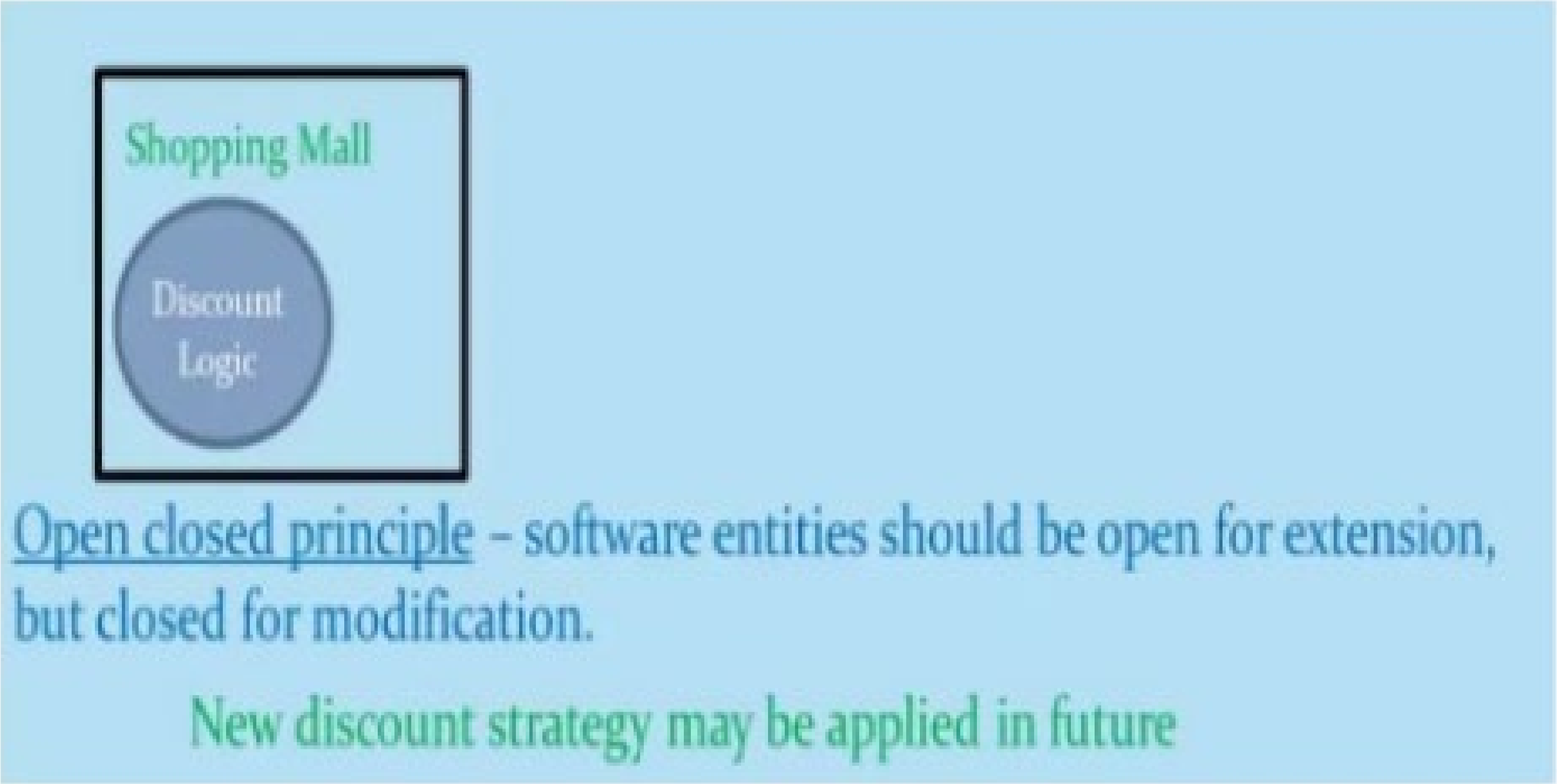


- Shopping Mall

1. Accept customer detail
2. Calculate bill amount
3. Apply discount based on day of week
   1. Monday – Low discount - 10%
   2. Thursday – High discount – 50%

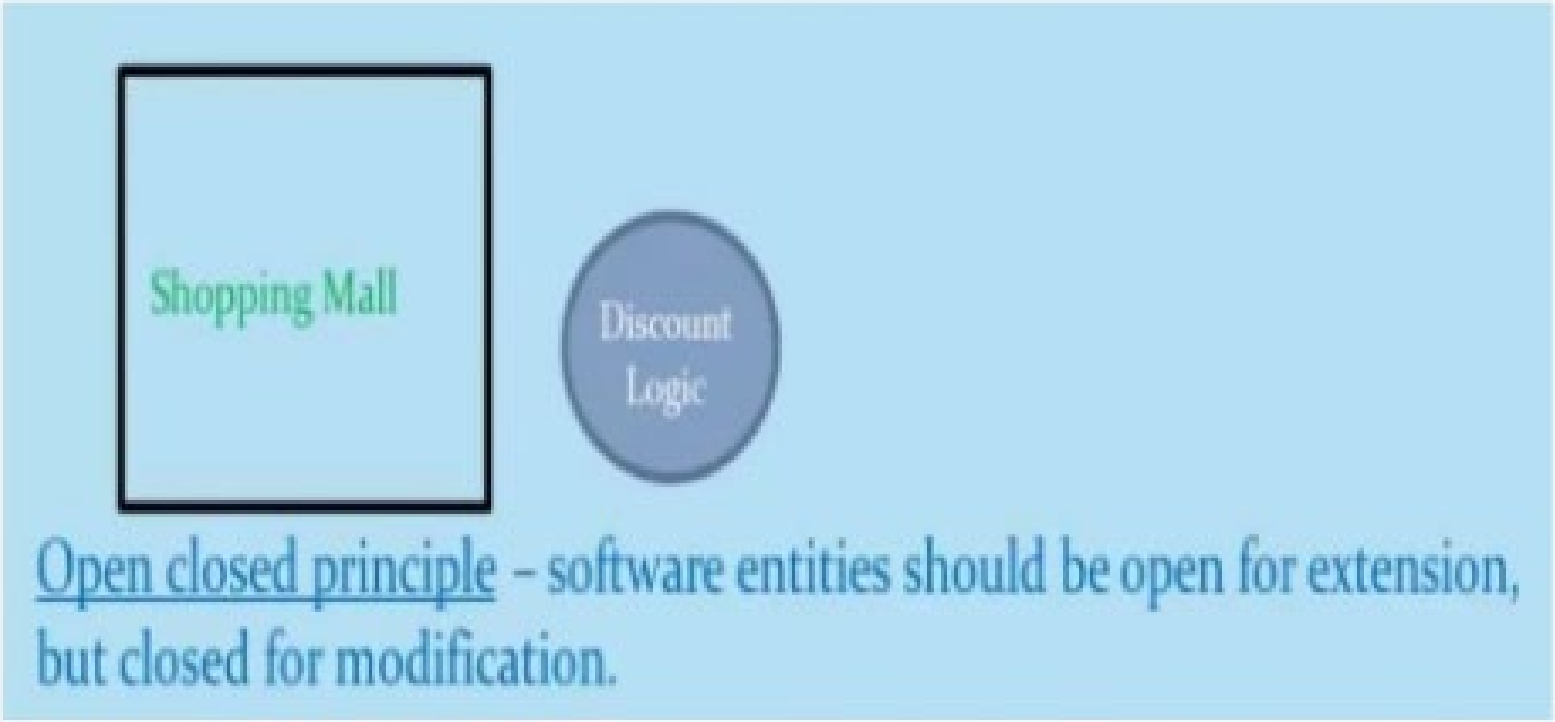
- Customer Detail
- CalculateBill
- GetFinalBill

Shopping Mall will contain discount logic.

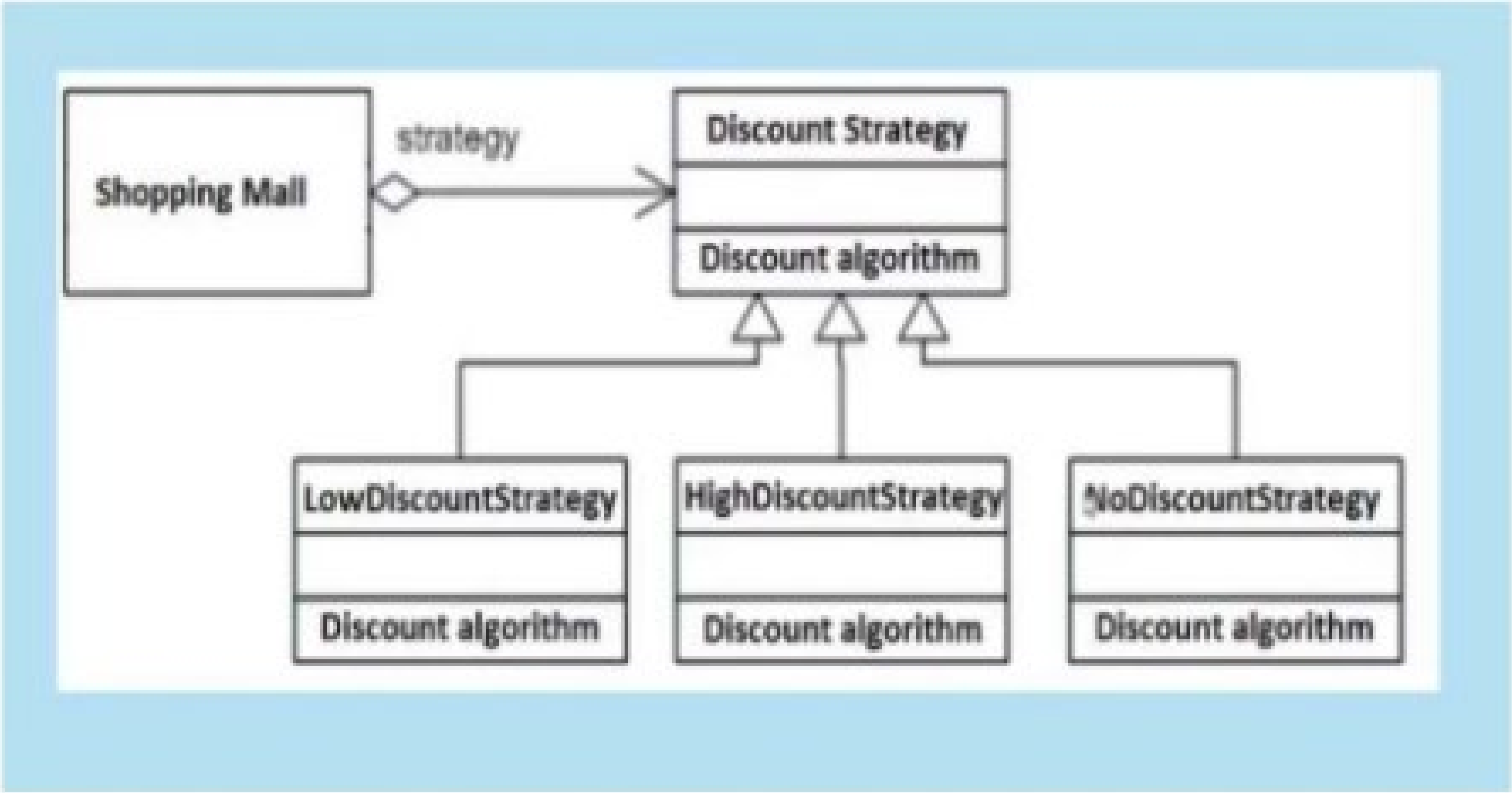


Open closed principle – software entities should be open for extension, but closed for modification.

New discount strategy may be applied in future



Open closed principle – software entities should be open for extension, but closed for modification.


Shopping Mall
strategy
Discount Strategy
Discount algorithm
LowDiscountStrategy
Discount algorithm
HighDiscountStrategy
Discount algorithm
NoDiscountStrategy
Discount algorithm

Software Architecture

- Software architecture is the blueprint of building software. It shows the overall structure of the software, the collection of components in it, and how they interact with one another while hiding the implementation.

- This helps the software development team to clearly communicate how the software is going to be built as per the requirements of customers.
- There are various ways to organize the components in software architecture. And the different predefined organization of components in software architectures are known as software architecture patterns.

- A lot of patterns were tried and tested. Most of them have successfully solved various problems. In each pattern, the components are organized differently for solving a specific problem in software architectures.

- Well, I hope you don’t want to bore yourself by reading the endless types of software architecture patterns.

Layers in Software Architecture

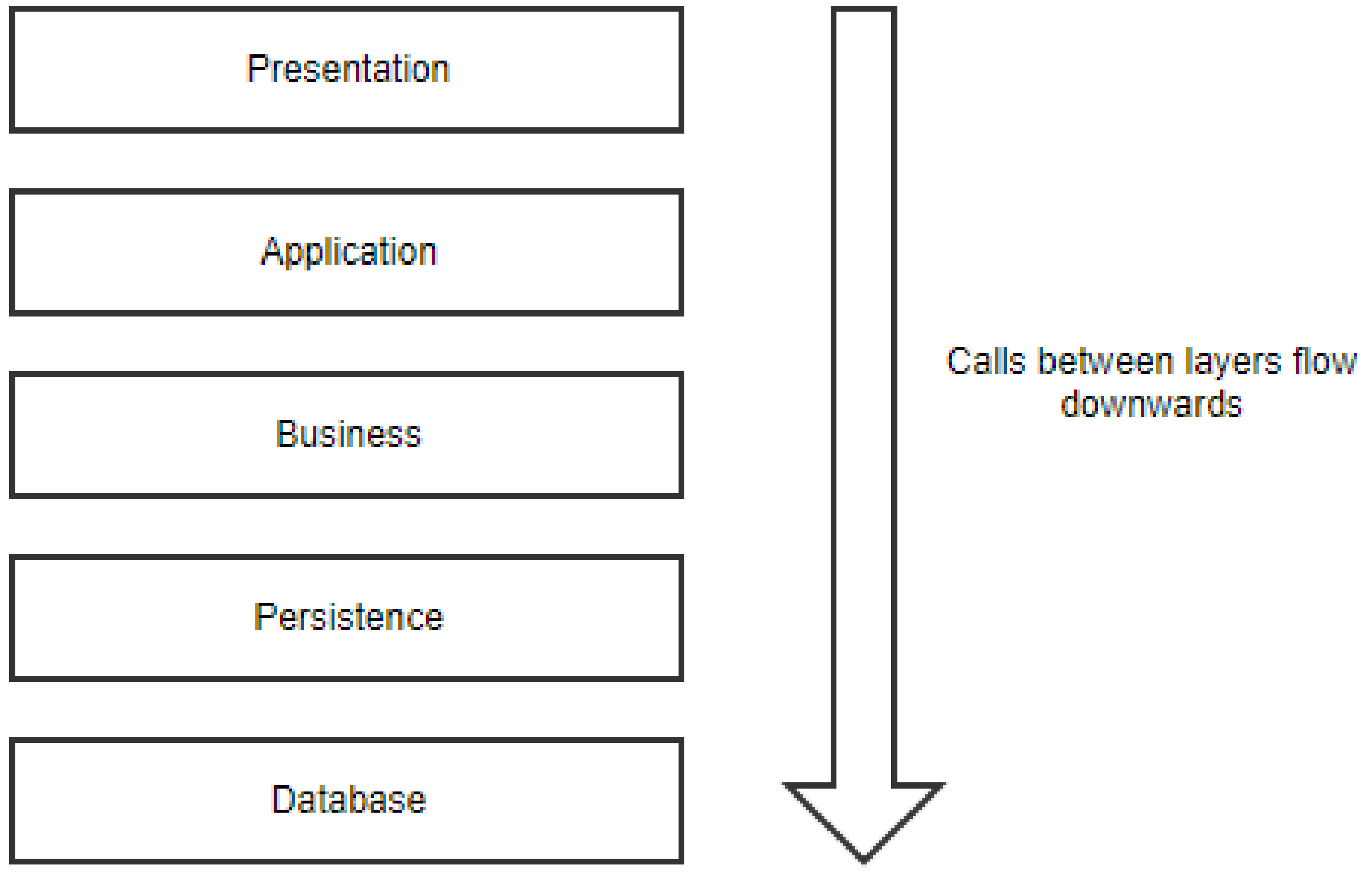
1. Presentation layer

2. Application layer

3. Business layer

4. Persistence layer

5. Database layer



Presentation

Application

Business

Persistence

Database

Calls between layers flow downwards

1. Presentation layer

The presentation layer, also called the UI layer, handles the interactions that users have with the software. It's the most visible layer and defines the application's overall look and presentation to the end-users. This is the tier that's most accessible, which anyone can use from their client device, like a desktop, laptop, mobile phone or tablet.

2. Application layer

The application layer handles the main programs of the architecture. It includes the code definitions and most basic functions of the developed application. This is the layer that programmers spend most of their time in when working on the software. You can use this layer to implement specific coordination logic that doesn't align exactly with either the presentation or business layer.

3. Business layer

The business layer, also called the domain layer, is where the application's business logic operates. Business logic is a collection of rules that tell the system how to run an application, based on the organization's guidelines. This layer essentially determines the behavior of the entire application. After one action finishes, it tells the application what to do next.

4. Persistence layer

The persistence layer, also called the data access layer, acts as a protective layer. It contains the code that's necessary to access the database layer. This layer also holds the set of codes that allow you to manipulate various aspects of the database, such as connection details and SQL statements.

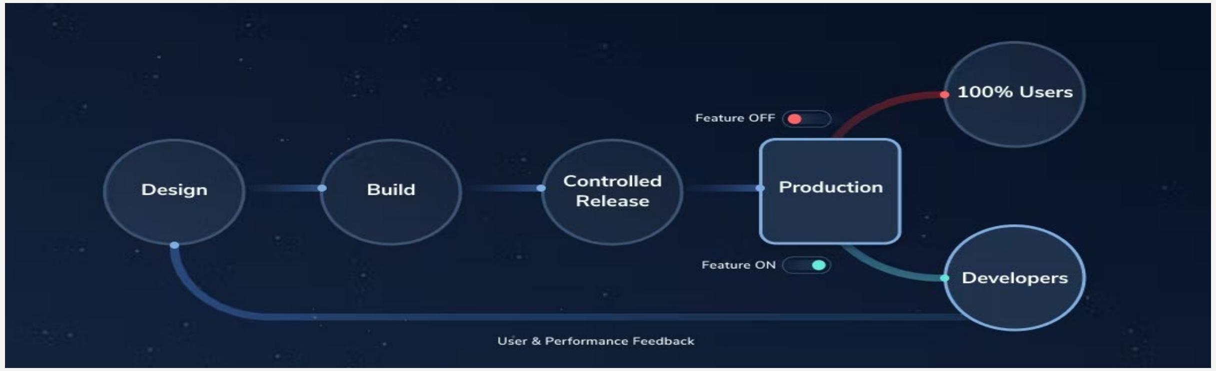
5. Database layer

The database layer is where the system stores all the data. It's the lowest tier in the software architecture and houses not only data but indexes and tables as well. Search, insert, update and delete operations occur here frequently. Application data can store in a file server or database server, revealing crucial data but keeping data storage and retrieval procedures hidden.

Environments in industry

There are different Types of environments in Industry.

1. The analysis and design environment
2. The development environment
3. The common build environment
4. The testing environment
5. The production environment



Environments in industry

1. Analysis & Design Environment

- The analysis and design environment is aligned to the **planning and analysis phases** of the SDLC. In this environment, the main processes that take place include carrying out an in-depth examination of the current system and the proposed system. The system architecture is also defined and includes developing the design of the hardware, software, and network requirements for the system. Within this environment, systems and business analysts work closely with software engineers.

2. The development environment

- The development environment can also be a physical space where development takes place and where software engineers interact. Another example of the development environment is the **integrated development environment** (IDE). The IDE provides a platform where tools and development processes are coordinated in order to provide software engineers a convenient way of accessing the resources they require during the development process.

Environments in industry

3. The common build environment

- The **common build environment** is closely aligned to the development phase of the SDLC. In this environment, software engineers merge the work done in the development environment. Within this environment, software engineers build systems. These are used to automate the process of software compilation.

4. The testing environment

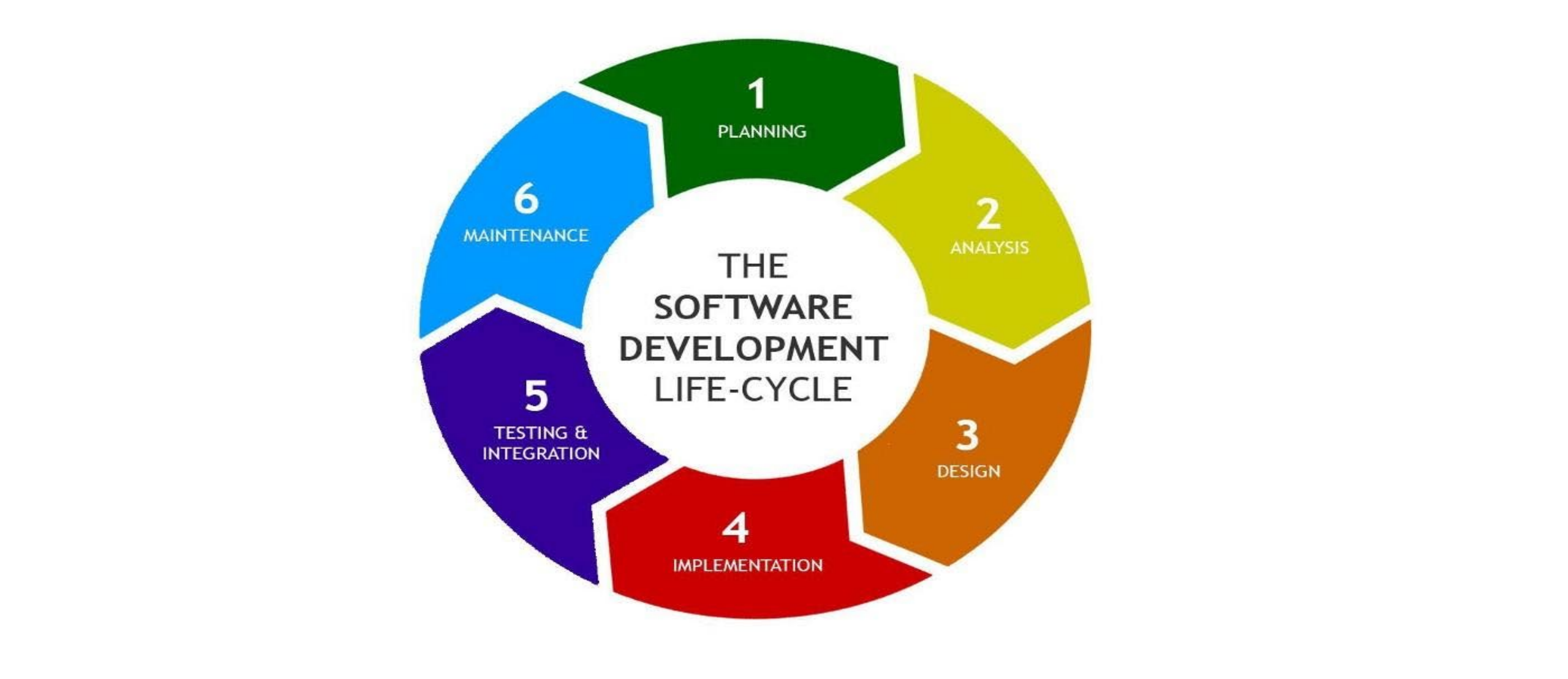
- The test environment is where testing teams evaluate the application/quality. program’s This also allows computer programmers to find out and solve any defects that may interfere with the application’s smooth operation or degrade the user experience.

5. The production environment

- When the end-user use a web/mobile application, the program is operating on a production server. It’s been created in the production environment.

SDLC

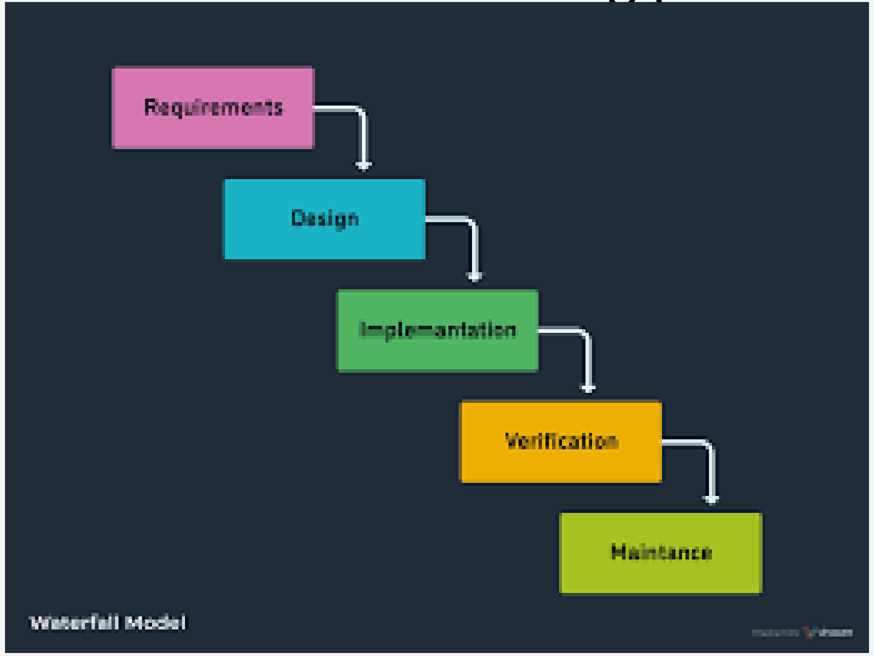
- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software.



SDLC Methodology

- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of software development:

1. Requirement Gathering
2. Analysis
3. Designing
4. Implementation
5. Testing
6. Maintenance



Programming

There are countless definitions of what computer programming is, but here is Ours.

“Programming is how *you* get computers to solve problems.”

There are two key phrases here that are important:

- **You**: without the programmer (you), the computer is useless. It does what **you** tell it to do.
- **Solve problems**: computers are tools. They are complex tools, admittedly, but they are not mysterious or magical: they exist to make tasks easier.

Programming

Computer programs make computers work

Computer programs (or software) are what makes computers work. Without software, modern computers are just complicated machines for turning electricity into heat. It’s software on your computer that runs your operating system, browser, email, games, movie player – just about everything.

Programming

Programming is creative:

Programming is a creative task: there is no right or wrong way to solve a problem, in the same way, that there is no right or wrong way to paint a picture.

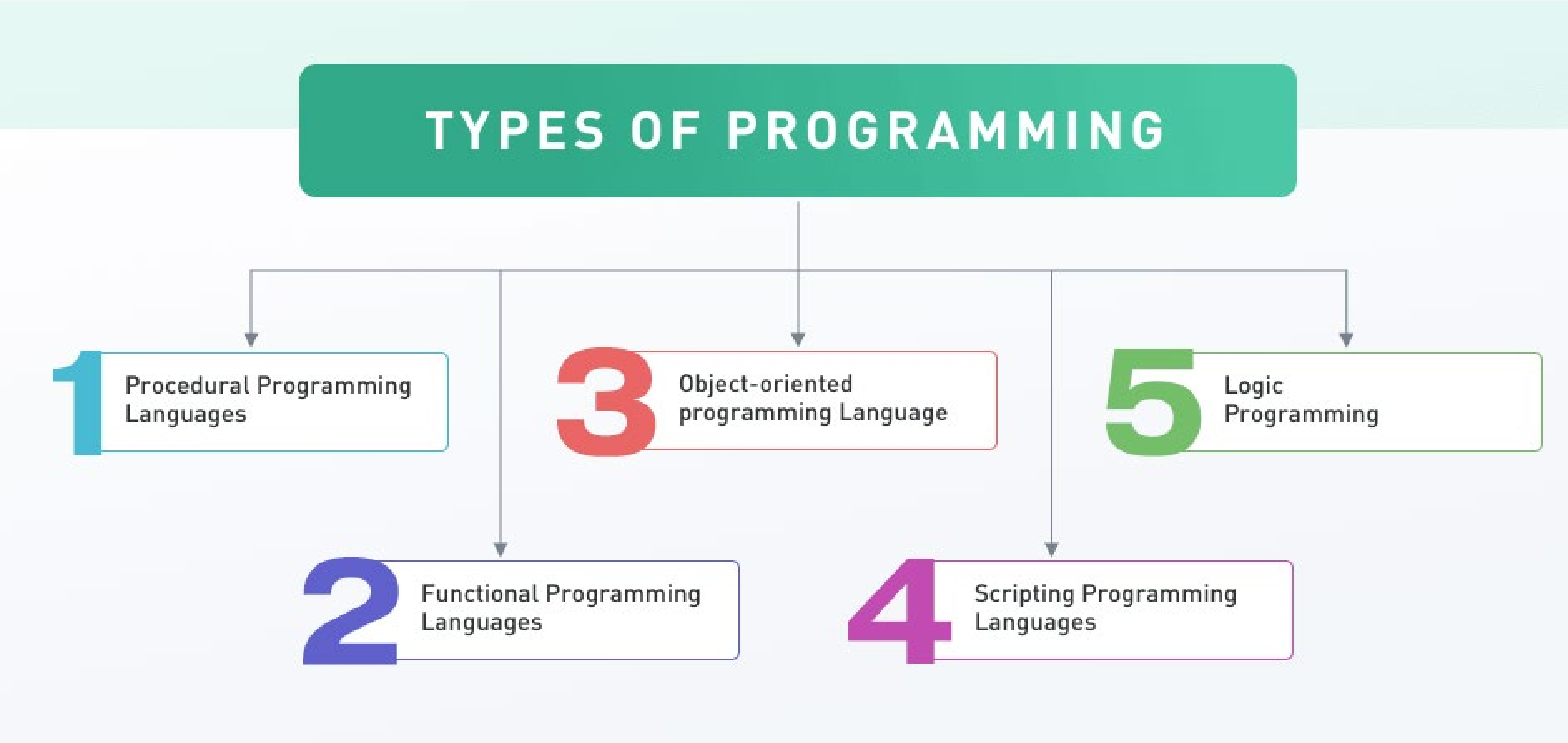
There are choices to be made, and one way may seem better than another, but that doesn’t mean the other is wrong! With the right skills and experience, a programmer can craft software to solve an unlimited number of problems – from telling you when your next train will arrive at playing your favourite music.

The possibilities are constrained only by your imagination. That’s why I love programming.Programming having a multiple Programming Languages.

Programing Languages

- A **programming language** is a computer language programmers use to develop software programs, scripts, or other sets of instructions for computers to execute.
- Although many languages share similarities, each has its own syntax. Once a programmer learns the languages rules, syntax, and structure, they write the source code in a text editor or IDE. Then, the programmer often compiles the code into machine language that can be understood by the computer. Scripting languages, which do not require a compiler, use an interpreter to execute the script.
- A programming language is a set of instructions that can be used to interact with and control a computer. These languages are used to design websites, create apps, develop operating systems, control spacecraft, and analyze data. Programming languages are necessary because computers can't understand English. Programming languages bridge this gap by helping programmers translate their commands into something that the computer can understand and execute.

Types of Programing Languages



TYPES OF PROGRAMMING

1 Procedural Programming Languages

2 Functional Programming Languages

3 Object-oriented programming Language

4 Scripting Programming Languages

5 Logic Programming

Types Of Programming Languages

1. Low-level programming language

Low-level language is **machine-dependent (0s and 1s)** programming language. The processor runs low- level programs directly without the need of a compiler or interpreter, so the programs written in low-level language can be run very fast.

Low-level language is further divided into two parts -

i. Machine Language

Machine language is a type of low-level programming language. It is also called as **machine code or object code**. Machine language is easier to read because it is normally displayed in binary or hexadecimal form (base 16) form. It does not require a translator to convert the programs because computers directly understand the machine language programs.

The advantage of machine language is that it helps the programmer to execute the programs faster than the high-level programming language.

ii. Assembly Language

Assembly language (ASM) is also a type of low-level programming language that is designed for specific processors. It represents the set of instructions in a **symbolic and human-understandable form**. It uses an assembler to convert the assembly language to machine language.

The advantage of assembly language is that it requires less memory and less execution time to execute a program.

2. High-level programming language

High-level programming language (HLL) is designed for **developing user-friendly software programs and websites**. This programming language requires a compiler or interpreter to translate the program into machine language (execute the program).

The main advantage of a high-level language is that it is **easy to read, write, and maintain**.

High-level programming language includes **Python, Java, JavaScript, PHP, C#, C++, Objective C, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift programming language**.

A high-level language is further divided into three parts -

i. Procedural Oriented programming language

Procedural Oriented Programming (POP) language is derived from structured programming and based upon the procedure call concept. It divides a program into small procedures called **routines or functions**.

Procedural Oriented programming language is used by a software programmer to create a program that can be accomplished by using a programming editor like IDE, Adobe Dreamweaver, or Microsoft Visual Studio.

The advantage of POP language is that it helps programmers to easily track the program flow and code can be reused in different parts of the program.

Example: C, FORTRAN, Basic, Pascal, etc.

ii. Object-Oriented Programming language

Object-Oriented Programming (OOP) language is **based upon the objects**. In this **programming language, programs are divided into small parts called objects**. It is used to implement real-world entities like inheritance, polymorphism, abstraction, etc in the program to makes the program resusable, efficient, and easy-to-use.

The main advantage of object-oriented programming is that OOP is faster and easier to execute, maintain, modify, as well as debug.

Example: C++, Java, Python, C#, etc.

iii. Natural language

Natural language is a **part of human languages** such as English, Russian, German, and Japanese. It is used by machines to understand, manipulate, and interpret human's language. It is used by developers to **perform tasks such as translation, automatic summarization, Named Entity Recognition (NER), relationship extraction, and topic segmentation**.

The main advantage of natural language is that it helps users to ask questions in any subject and directly respond within seconds.

3. Middle-level programming language

Middle-level programming language **lies between the low-level programming language and high-level programming language**. It is also known as the intermediate programming language and pseudo-language.

A middle-level programming language's advantages are that it supports the features of high-level programming, it is a user-friendly language, and closely related to machine language and human language.

Example: C, C++, language

Important Programming Language

- **Development is Mainly Divided into two Parts**

Front end Development

HTML

CSS

JAVASCRIPT

React

Angular

Important Programming Language

Backend Development

In Backend you have a many languages you can choose anyone language

If you are Choosing web development platform then you have a Mainly Three Languages

PHP

JAVA

PYTHON

Dot Net

Important Programming Language

HTML:

HTML—“HyperText Markup Language”—is **the language used to tell your web browser what each part of a website is**. So, using HTML, you can define headers, paragraphs, links, images, and more, so your browser knows how to structure the web page you're looking at.

CSS:

CSS makes the front-end of a website shine and it creates a great user experience. Without CSS, websites would be less pleasing to the eye and likely much harder to navigate. In addition to layout and format, CSS is responsible for font color and more.

Important Programming Language

JAVASCRIPT:

JavaScript has evolved over the past 25 years **to become a versatile and accessible programming language for working with web browsers**. Developers use JavaScript to build complex interactive websites and browser games, and to connect servers to websites and web applications

Important Programming Language

Programming Language in Backend:

The backend (or “server-side”) is the portion of the website you don't see. It's **responsible for storing and organizing data, and ensuring everything on the client-side actually works**. The backend communicates with the frontend, sending and receiving information to be displayed as a web page.

Writing Source Code

What is Source Code?

Source code is the source of a computer program. It contains declarations, instructions, functions, loops and other statements, which act as instructions for the program on how to function. Programs may contain one or more source code text files, which can be stored on a computer's hard disk, in a database, or be printed in books of code snippets.

Important Notes For Source Code

- Coding Must Have Comments
- Coding Must Have a Proper Architecture
- Coding Must Have a
- Declaring Coding Versions
- Divide your code in packages

Running Code

When you will run your code two things are Most Important

1.Compiler

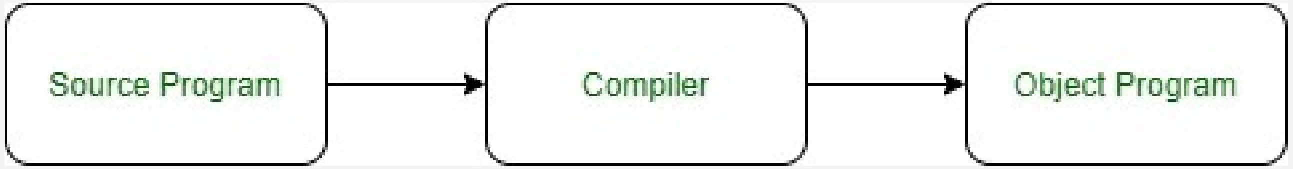
2.Interpreter

1.Compiler:

It is a translator which takes input i.e., High-Level Language, and produces an output of low-level language i.e. machine or assembly language.

- A compiler is more intelligent than an assembler it checks all kinds of limits, ranges, errors, etc.

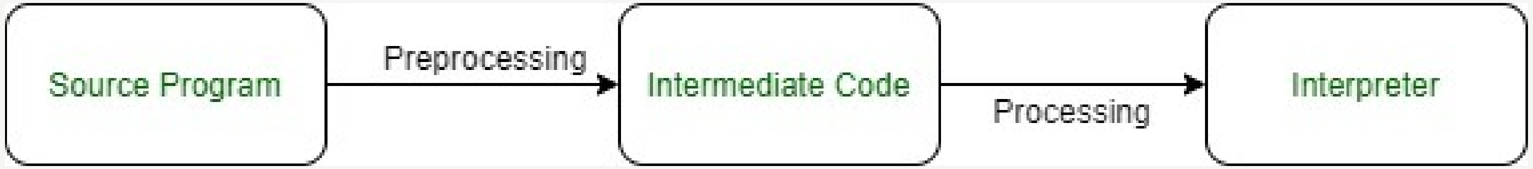
- But its program run time is more and occupies a larger part of memory. It has a slow speed because a compiler goes through the entire program and then translates the entire program into machine codes.



2.Interpreter :

An interpreter is a program that translates a programming language into a comprehensible language. –

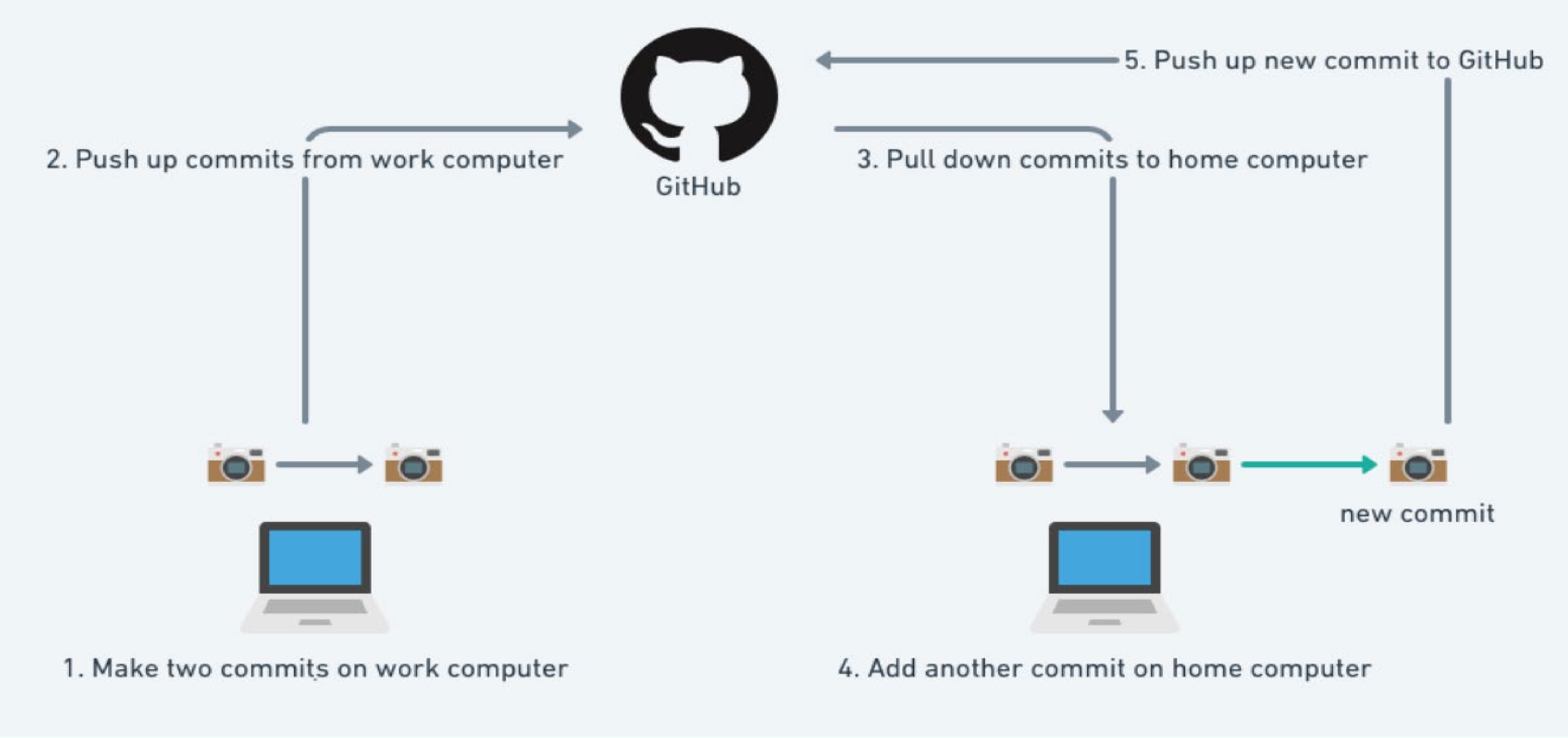
- It translates only one statement of the program at a time.
- Interpreters, more often than not are smaller than compilers.



GitHub

- GitHub is one of the most popular resources for developers to share code and work on projects together. It’s free, easy to use, and has become central in the movement toward open-source software.
- Git is used for managing the changes to a project over time. A project might be just a single file, a handful of files, or thousands of files. Those files can be anything from plain text to images or videos.
- Because Git is focused on managing changes, it is often used as a collaboration tool allowing people to work on the same project at the same time. By tracking their individual changes, Git can bring everything together to the final version.

Working with GitHub


5. Push up new commit to GitHub
2. Push up commits from work computer
GitHub
3. Pull down commits to home computer
new commit
1. Make two commits on work computer
4. Add another commit on home computer

THANK YOU