TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Software Engineering Course

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Content

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Module - 1 [Overview of IT Industry]

What is Program

- Program- It is a set of Instructions

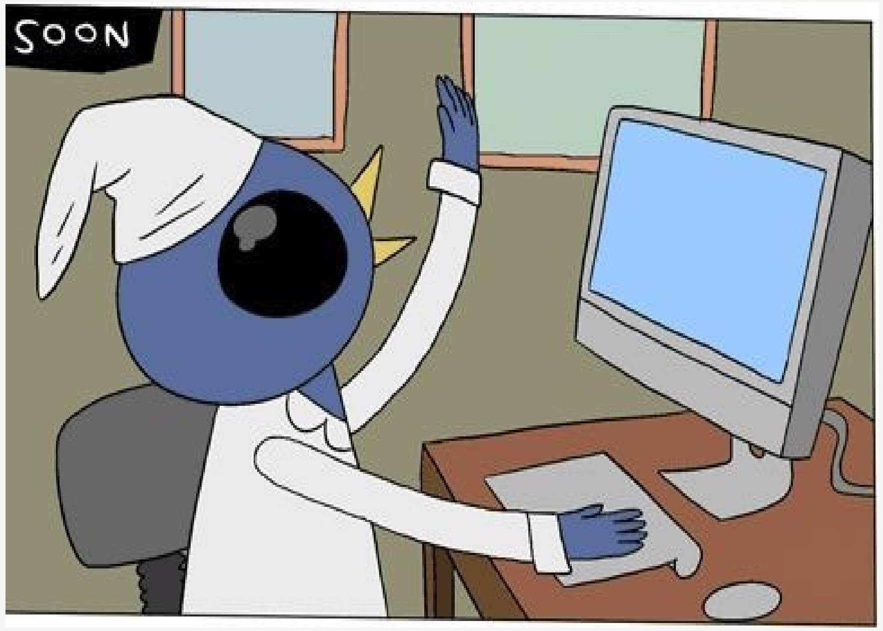
Ex. 1- Instructions to your Pet Ex. 2- Starting your Computer

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

What is Programming

- Programming- To create a Program.

Ex. 1- Using Keyboard & Mouse



Types of Programming Language

- Procedural Programming
  - Ex.- C Language

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Types of Programming Language

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Types of Programming Language

- Logical Programming
  - Ex.- Prolog Language

Types of Programming Language

- Functional Programming
  - Ex.- Python Language

World Wide Web - WWW

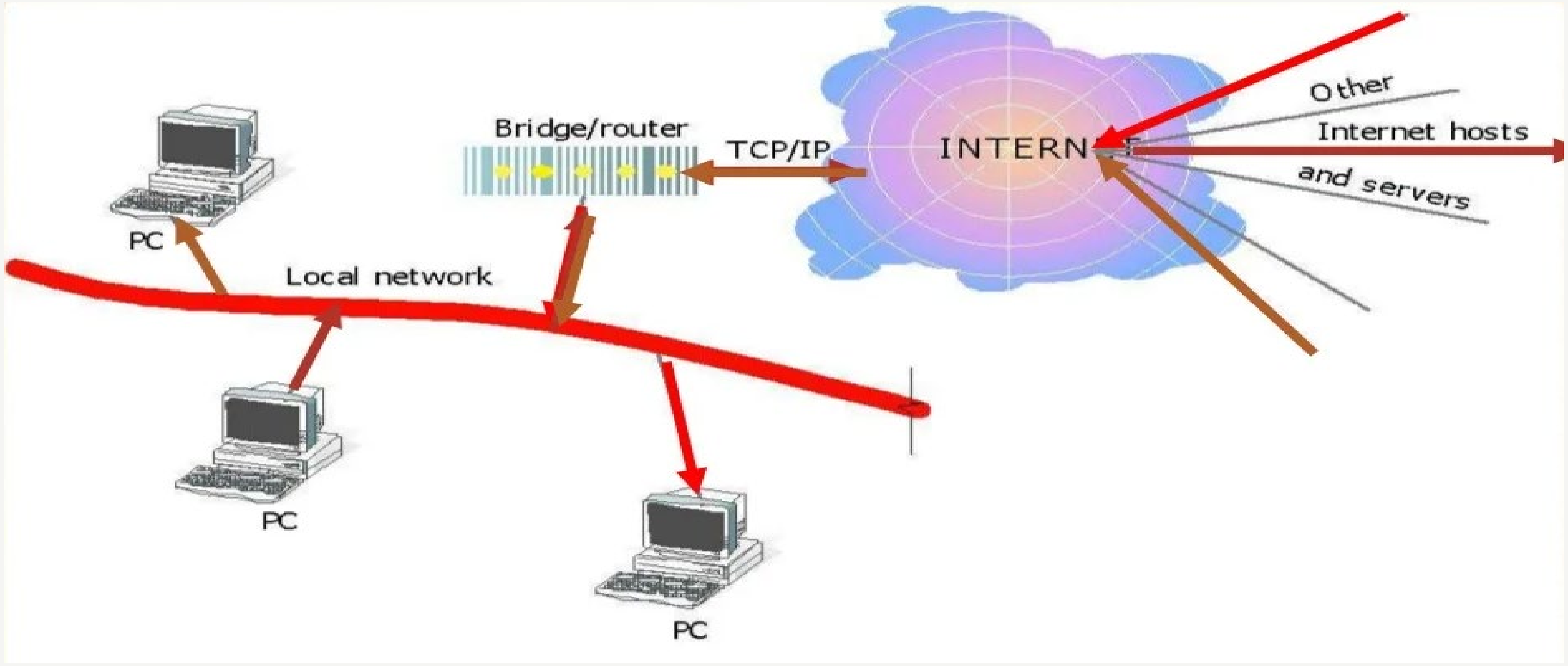
- known as a Web
- is a collection of websites or web pages stored in web servers
- connected to local computers through the internet
- These websites contain text pages, digital images, audios, videos, etc.
- Users can access the content of these sites from any part of the world over the internet using 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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

WWW – How Internet Works?

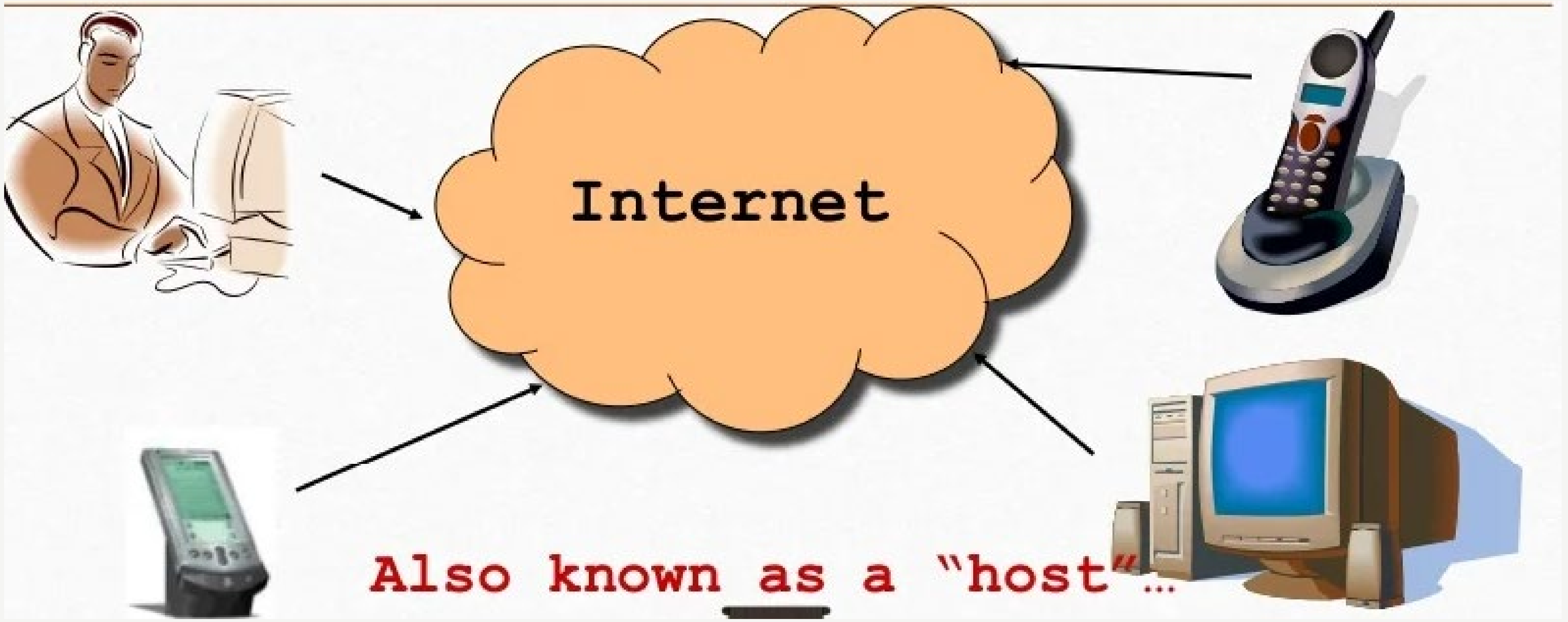
- A server is where websites are stored, and it works a lot like your computer's hard drive.
- Once the request arrives, the server retrieves the website and sends the correct data back to your computer.
- One of the best features of the Internet is the ability to communicate almost instantly with anyone 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?



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Network layers on Client & Server

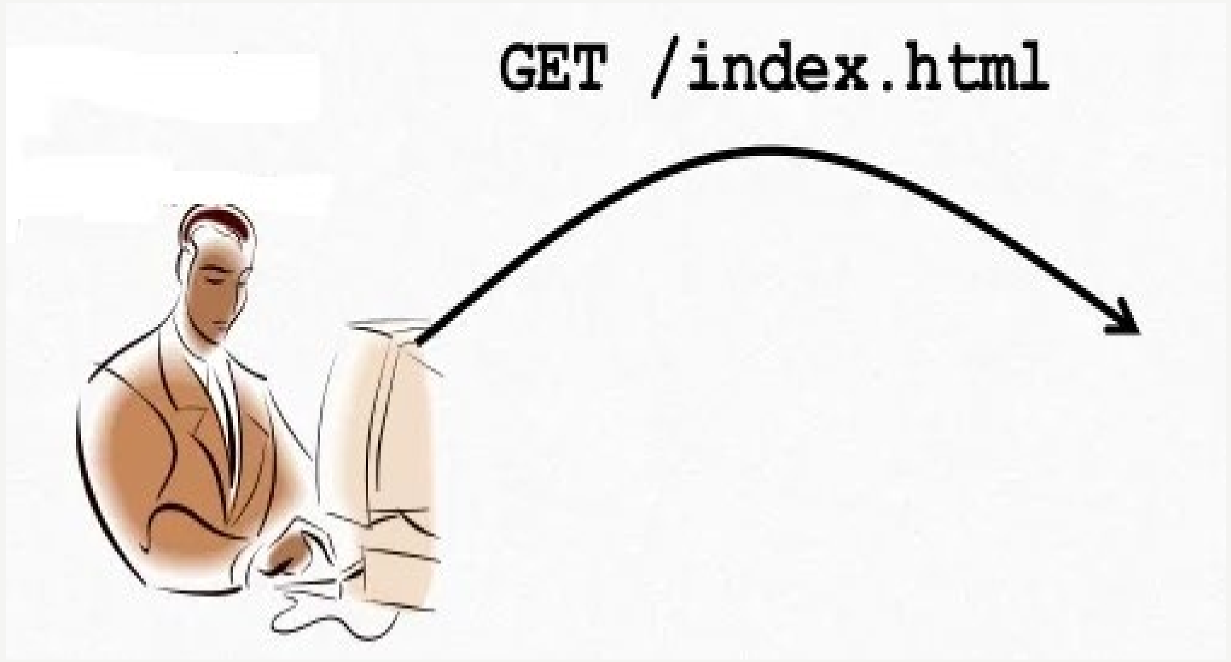


TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Client & Servers

1. **Client Program**

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



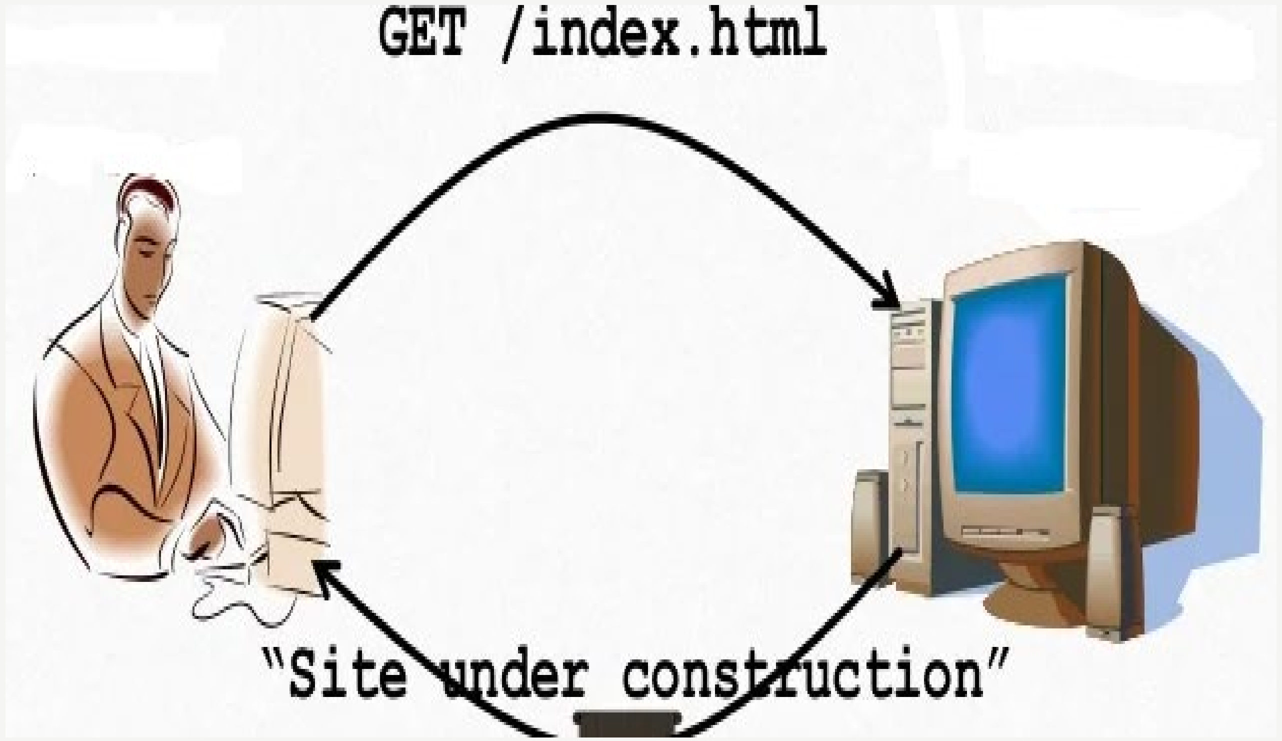
Client & Servers

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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.

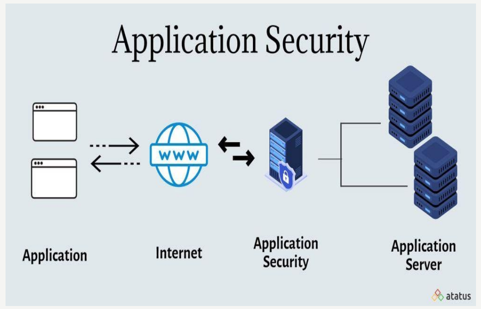
Types of Protocols

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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..



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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..

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

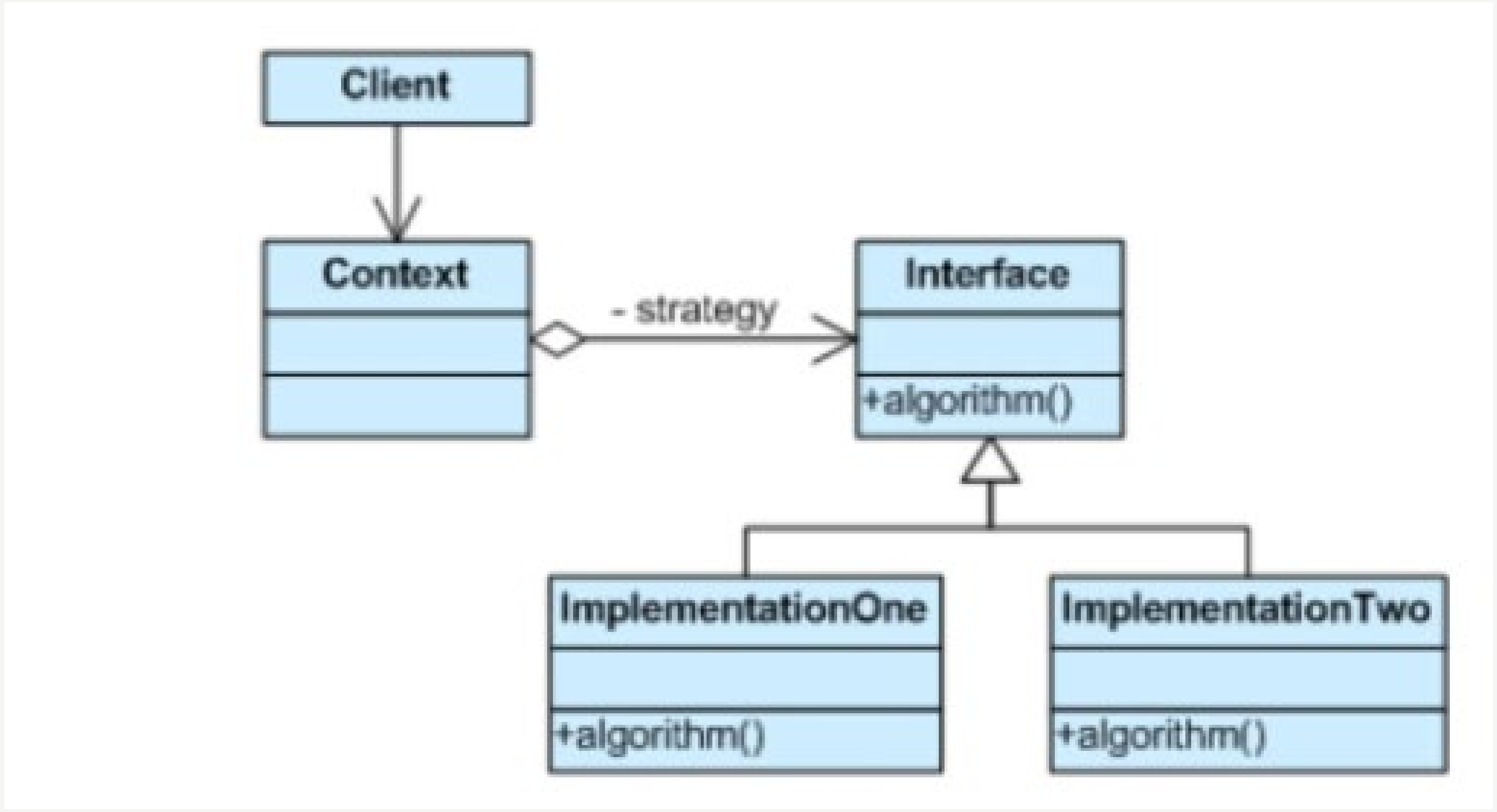
Programming Software

- 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..

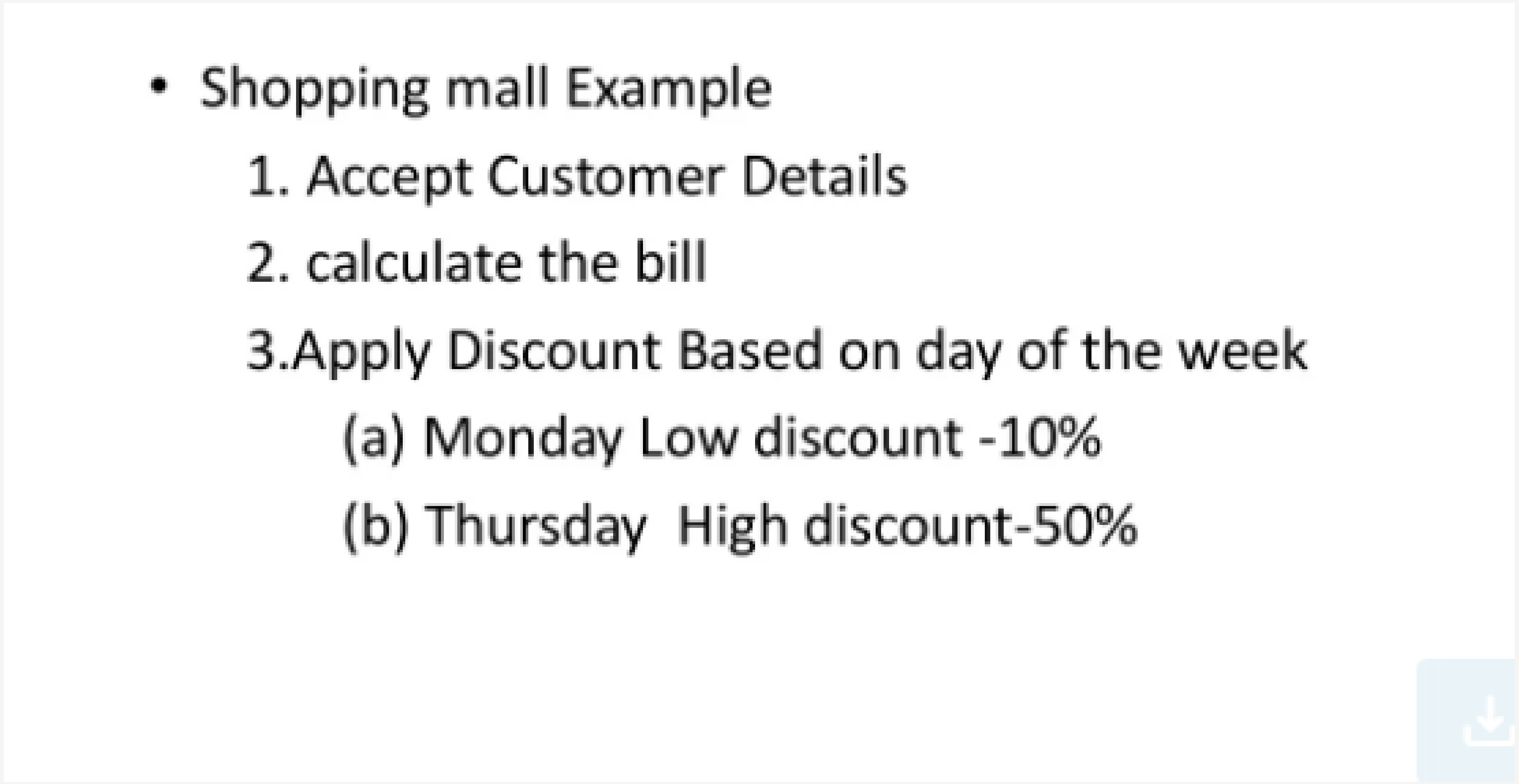
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

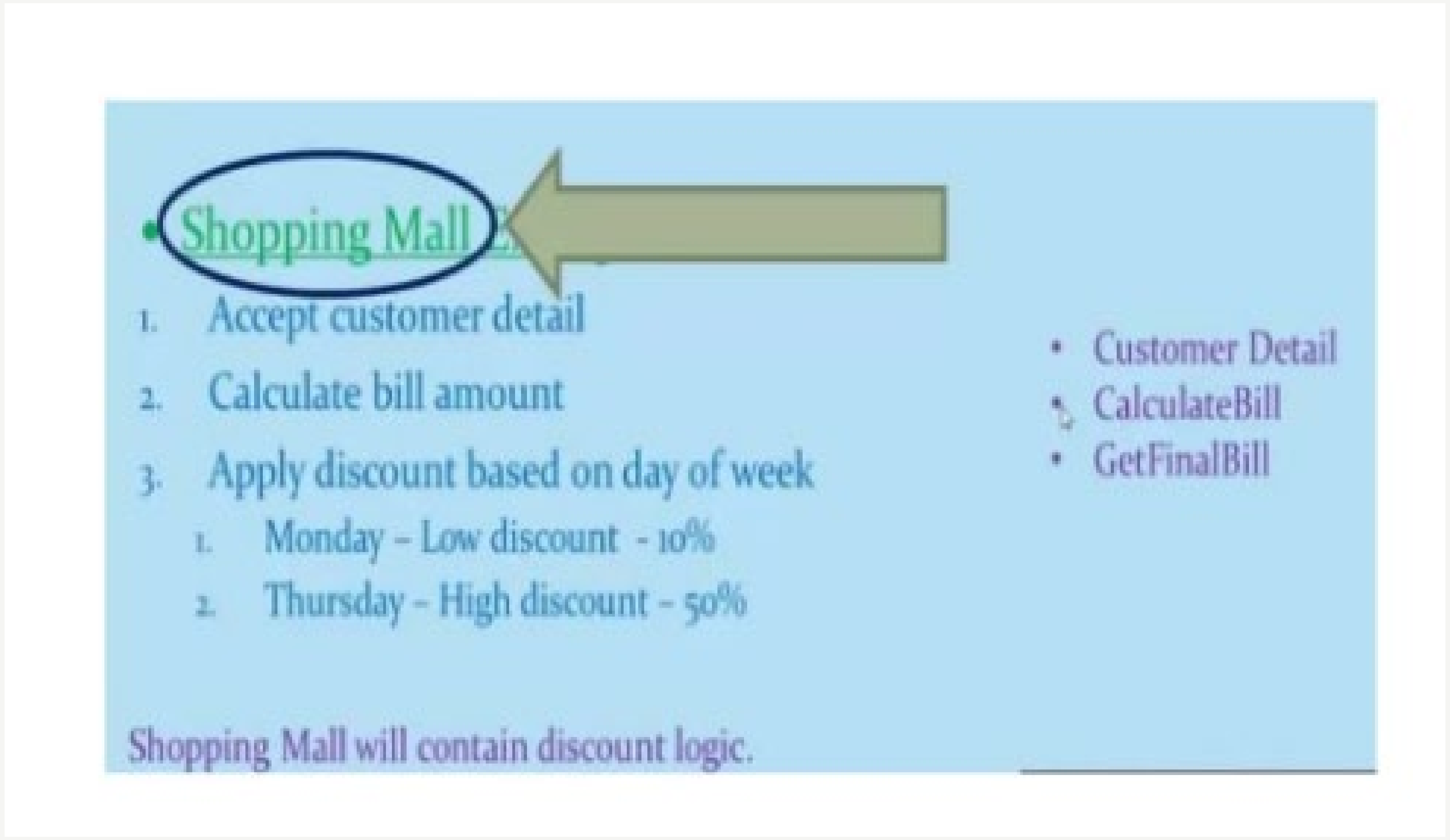
Architecture



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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%

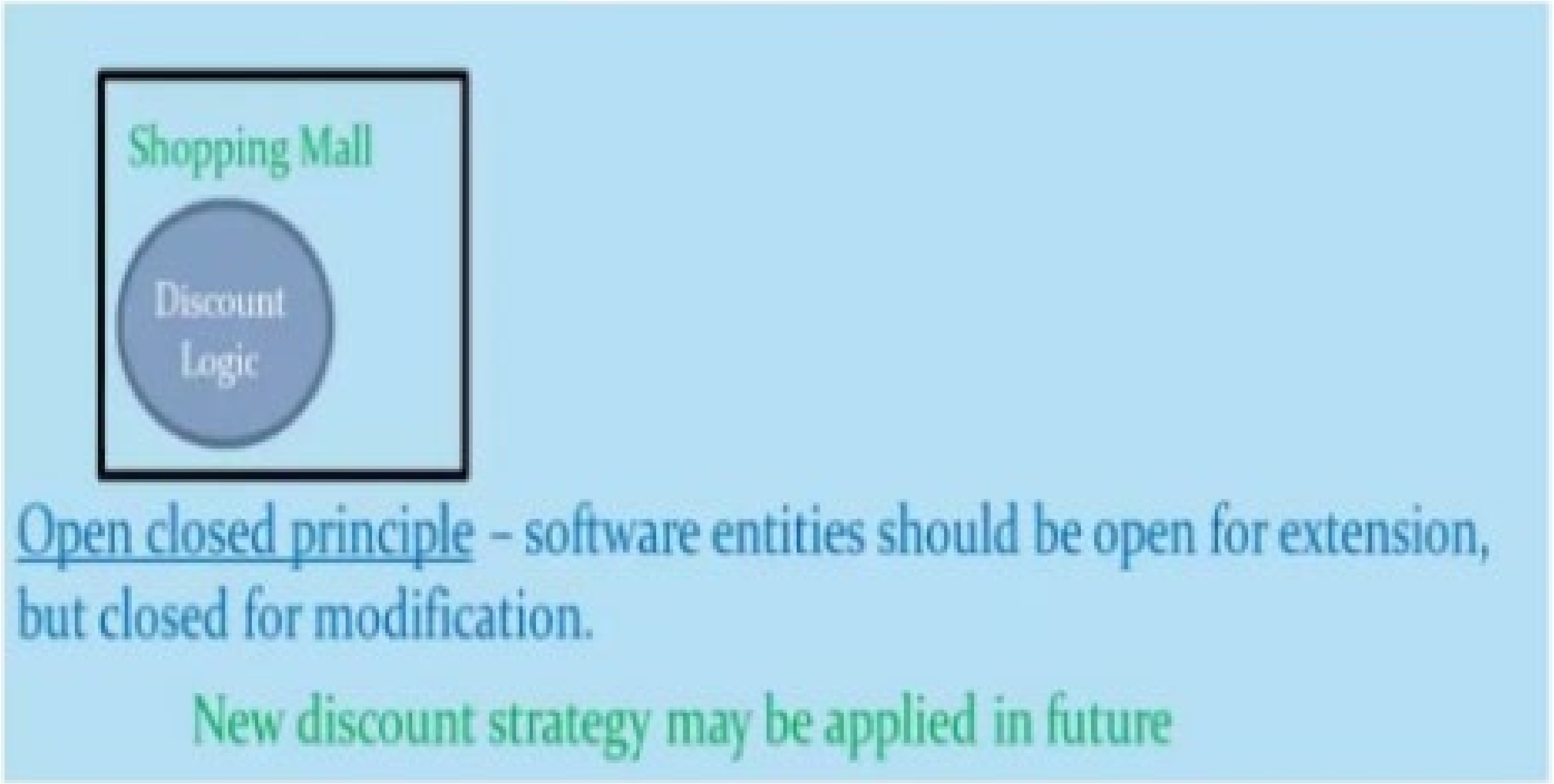


- Shopping Mall

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

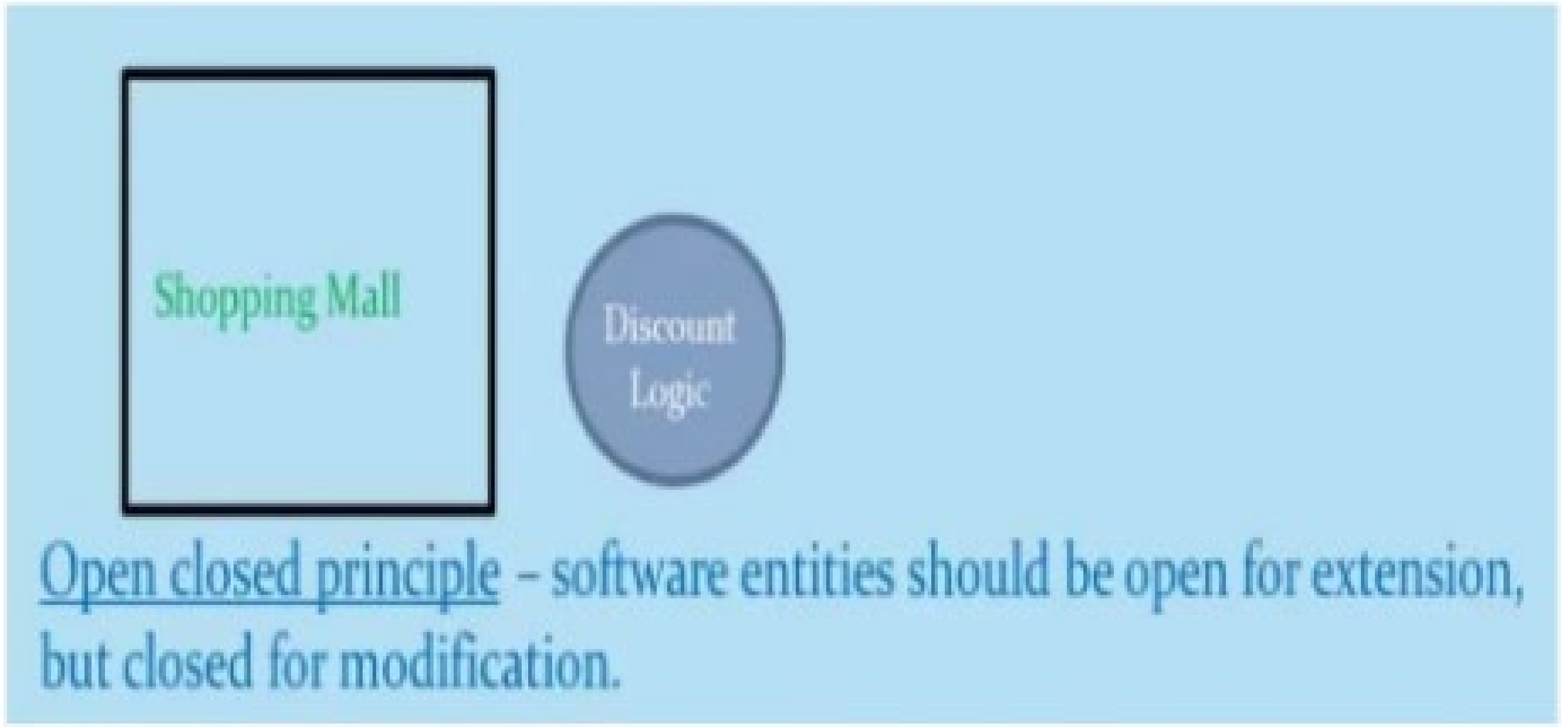
Shopping Mall will contain discount logic.



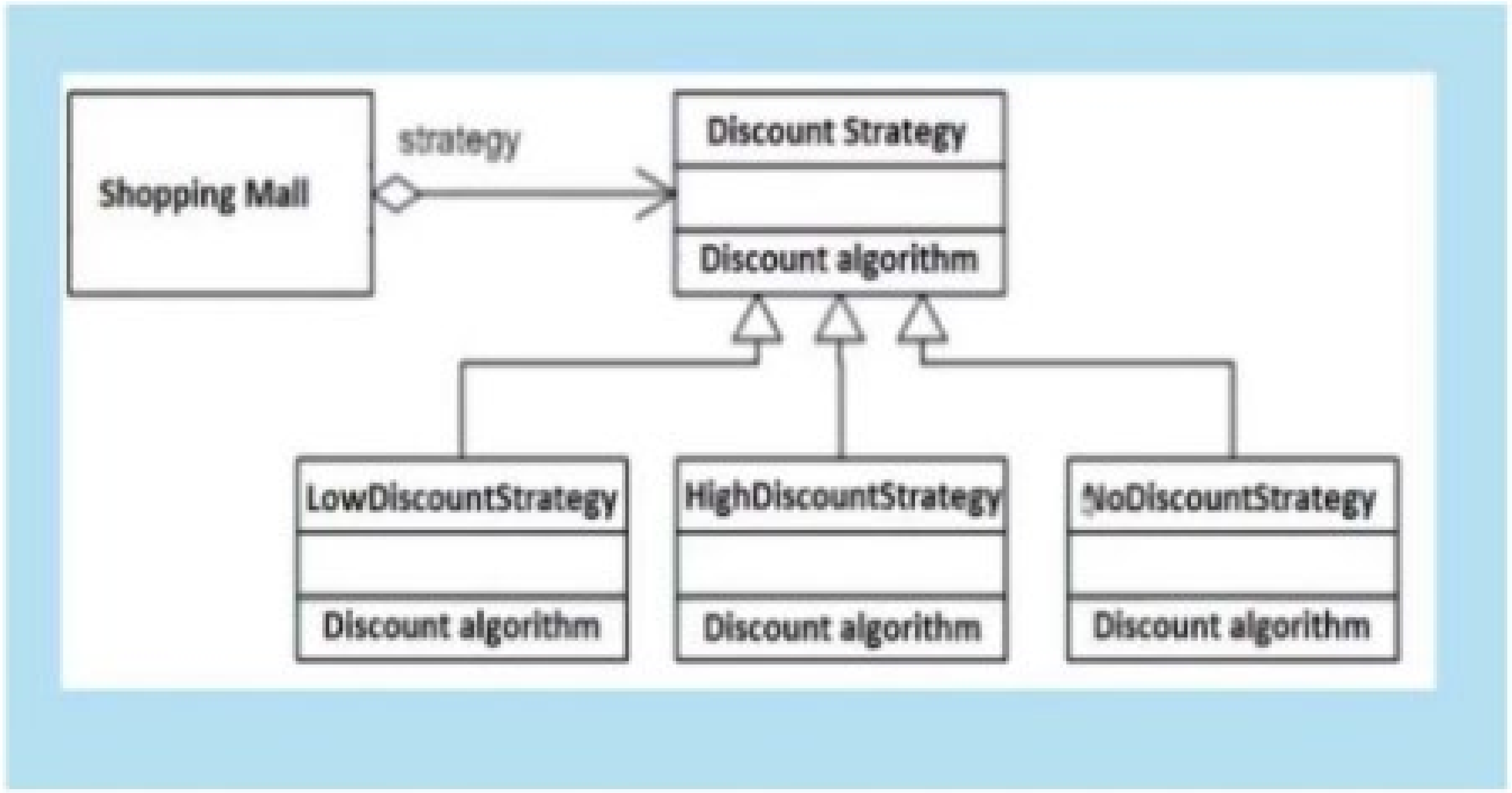
Shopping Mall
Discount Logic

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.

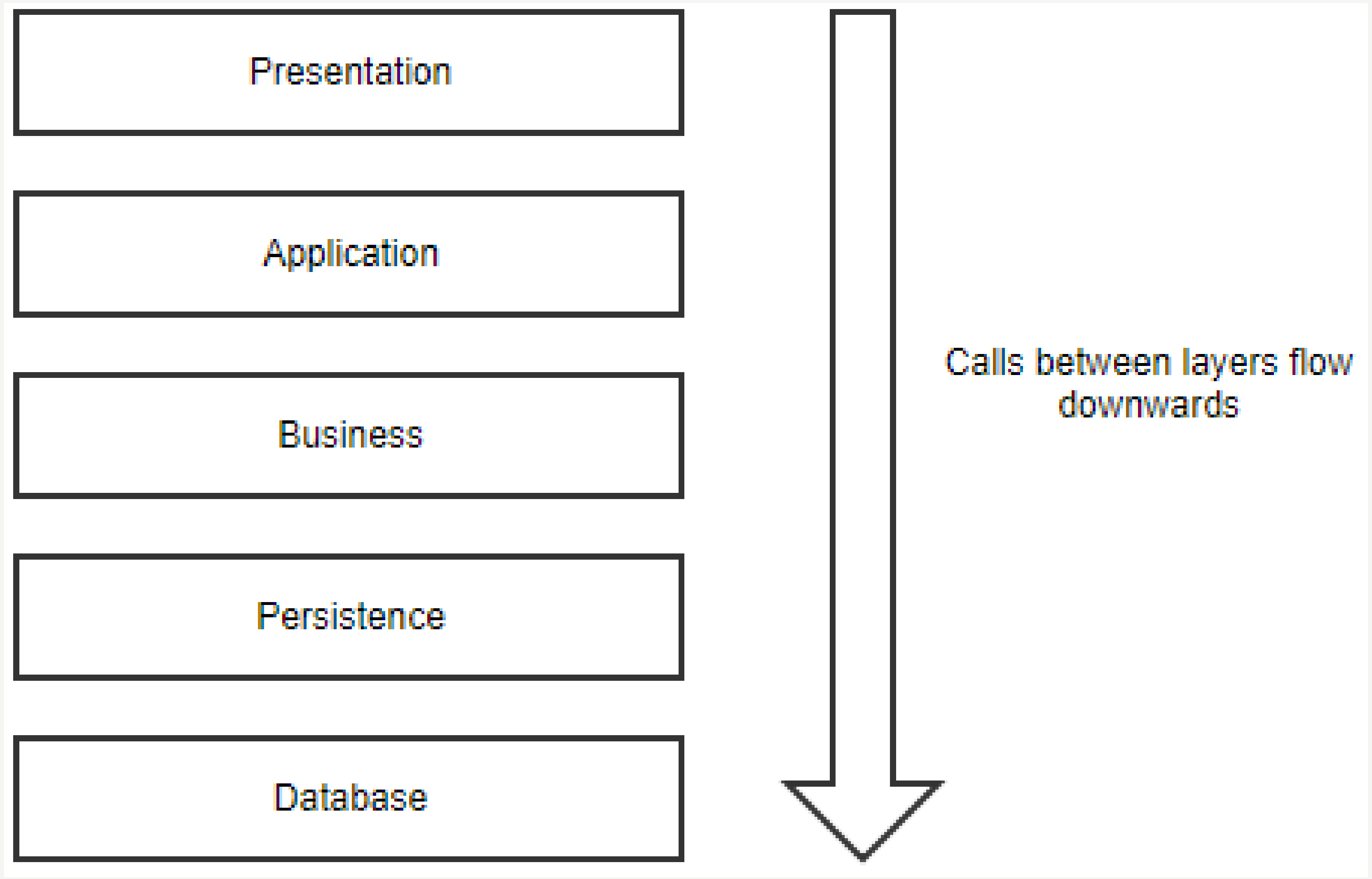
- A lot of patterns were tried and tested. Most of them have successfully solved various problems. In each pattern, the components are organized differently for solving a specific problem in software architectures.

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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

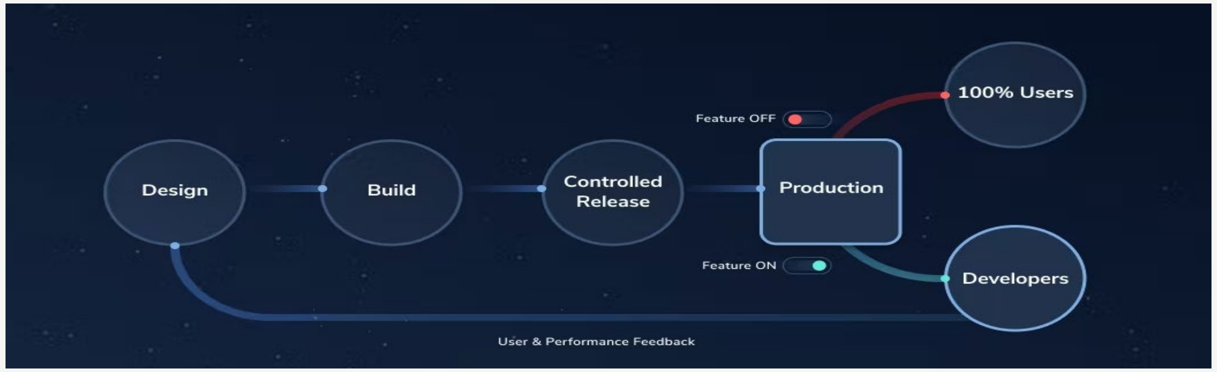
The database layer is where the system stores all the data. It's the lowest tier in the software architecture and houses not only data but indexes and tables as well. Search, insert, update and delete operations occur here frequently. Application data can store in a file server or database server, revealing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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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

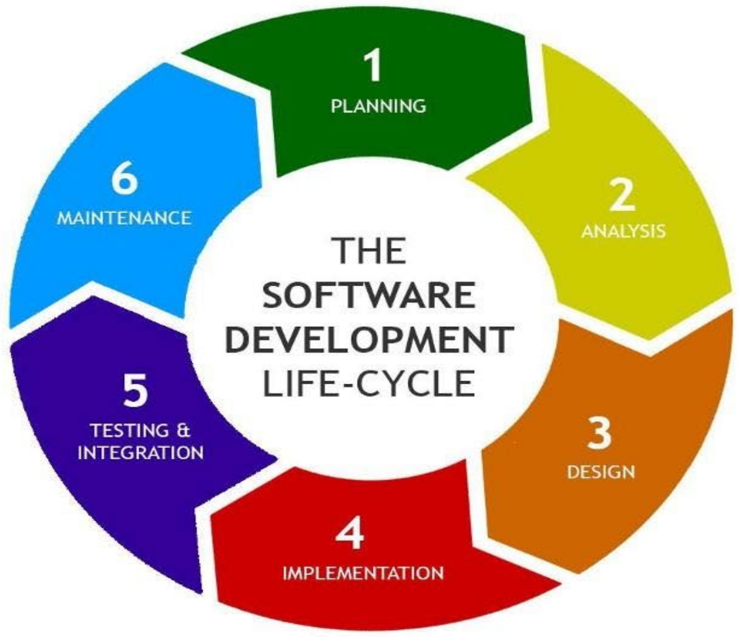
- The test environment is where testing teams evaluate the application/quality. program's This also allows computer programmers to find out and solve any defects that may interfere with the application's smooth operation or degrade the user experience.

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

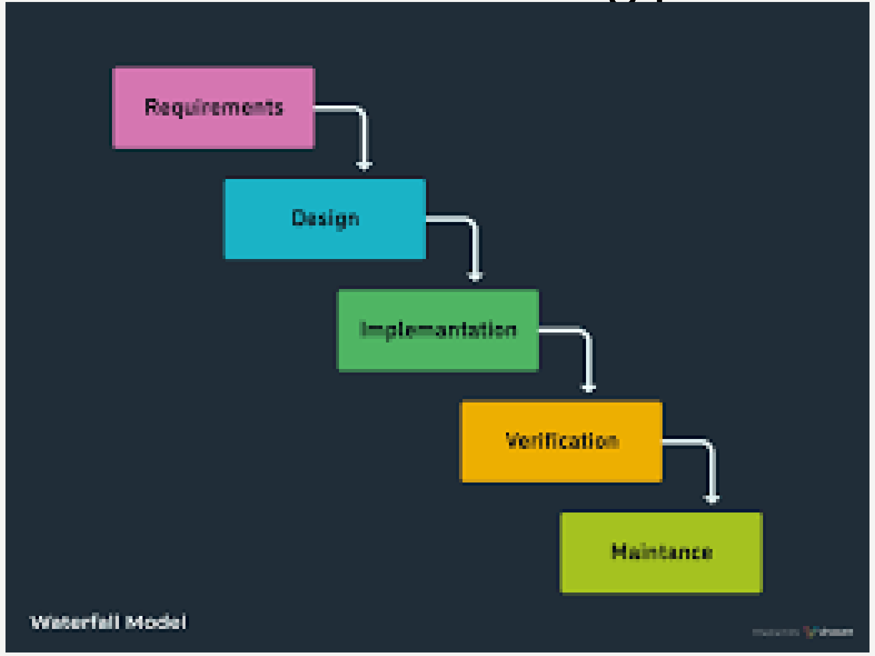
- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software.



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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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.

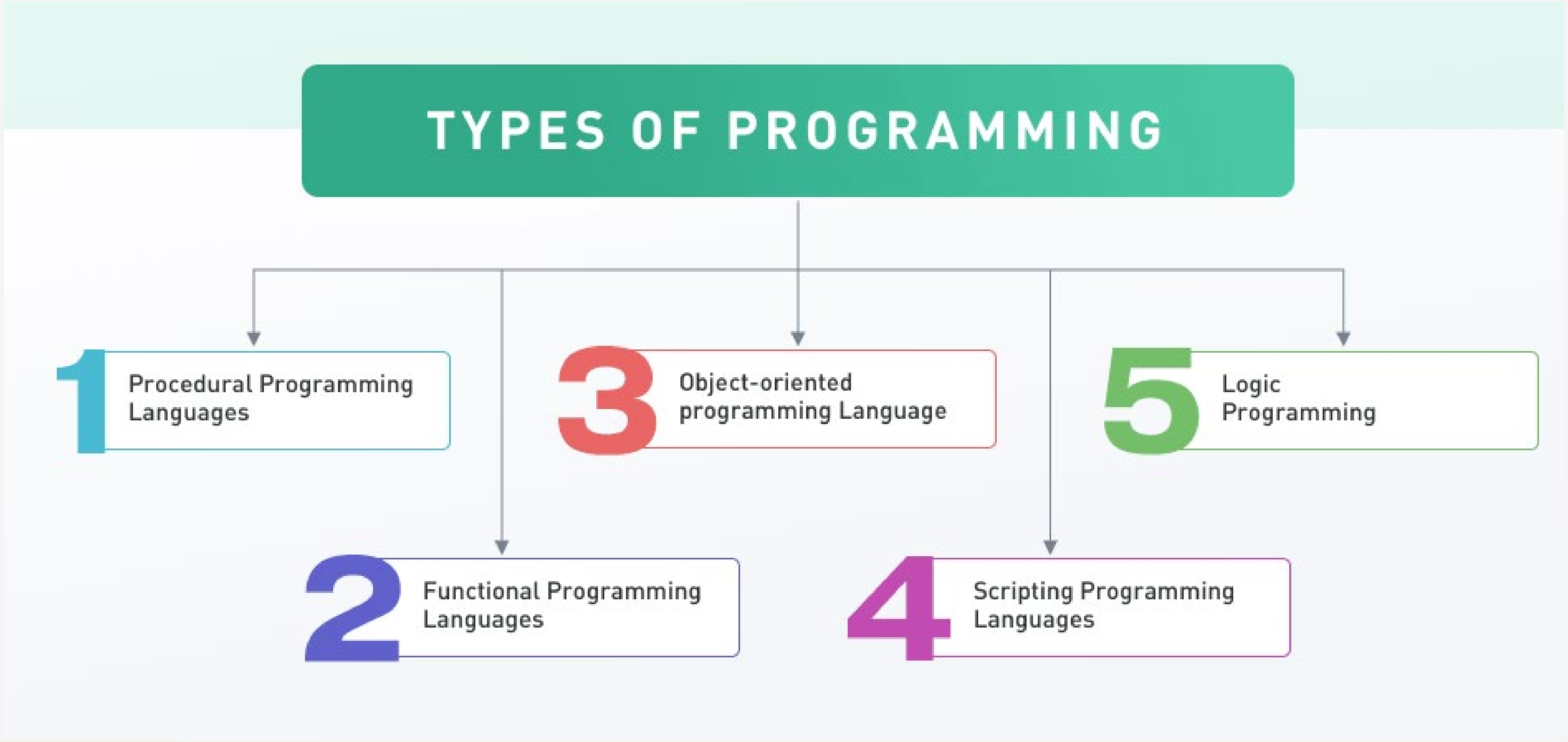
The possibilities are constrained only by your imagination. That’s why I love programming.Programming having a multiple Programming Languages.

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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Running Code

When you will run your code two things are Most Important

1.Compiler

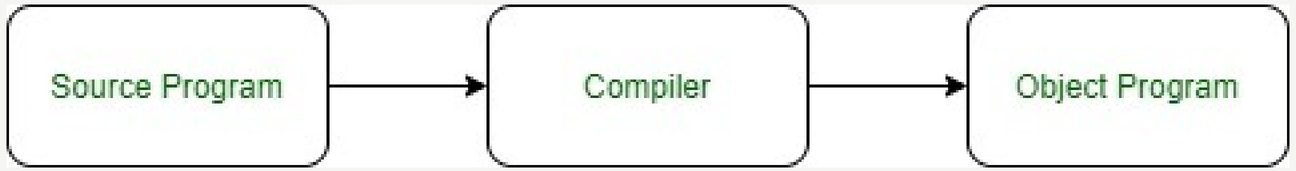
2.Interpreter

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.

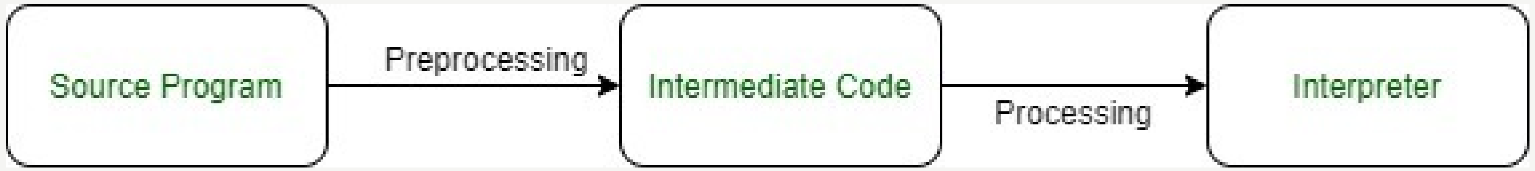
- But its program run time is more and occupies a larger part of memory. It has a slow speed because a compiler goes through the entire program and then translates the entire program into machine codes.



2.Interpreter :

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

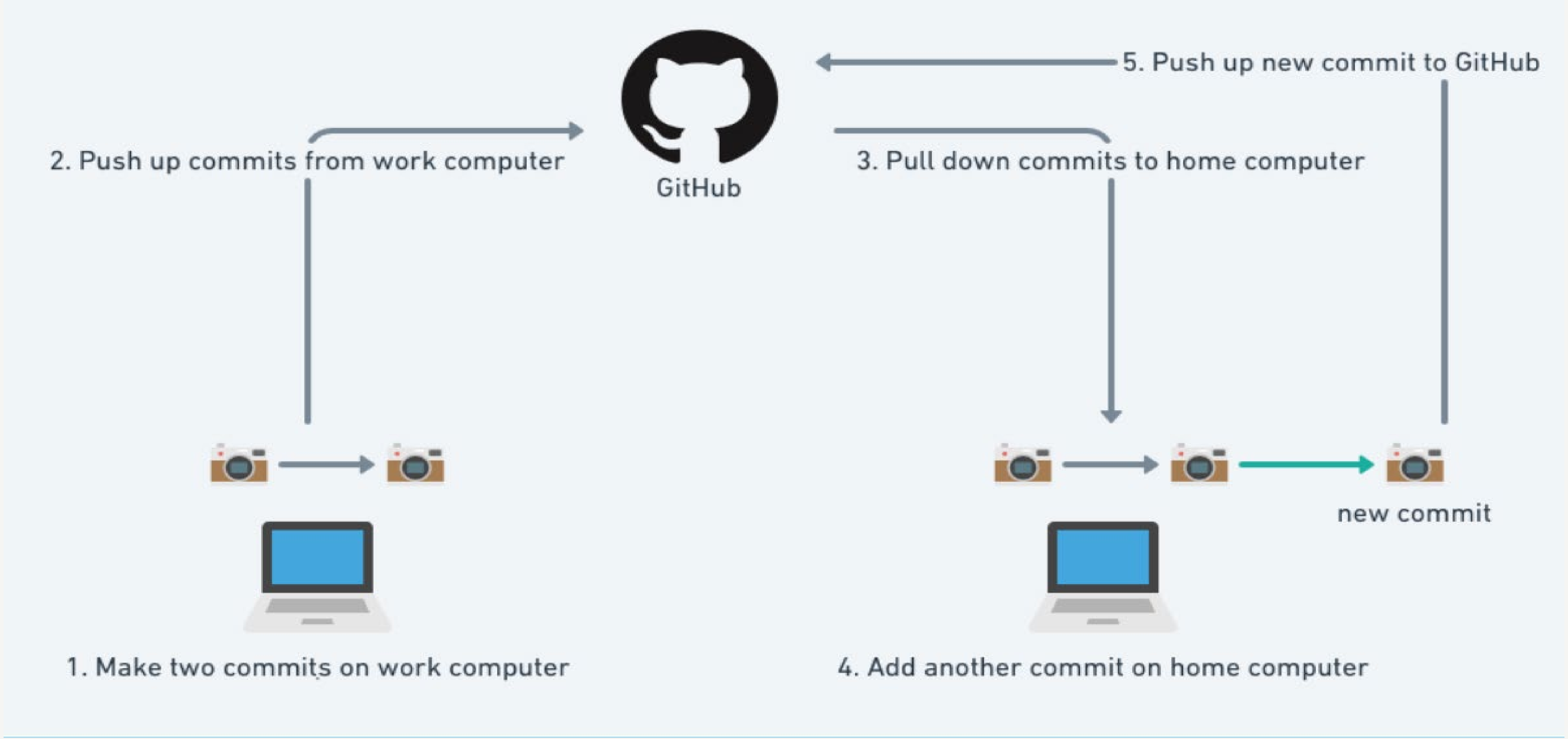
- It translates only one statement of the program at a time.
- Interpreters, more often than not are smaller than compilers.



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

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Module - 2
[Fundamentals of Programing]

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Basic Concepts of Programing Languages

1. Syntax
2. Data Structures
3. Variables
4. Operators
5. Control & Looping Structures
6. Functions
7. Arrays & Strings
8. File Handling

1.Basic Syntax

Syntax of Programing Languages

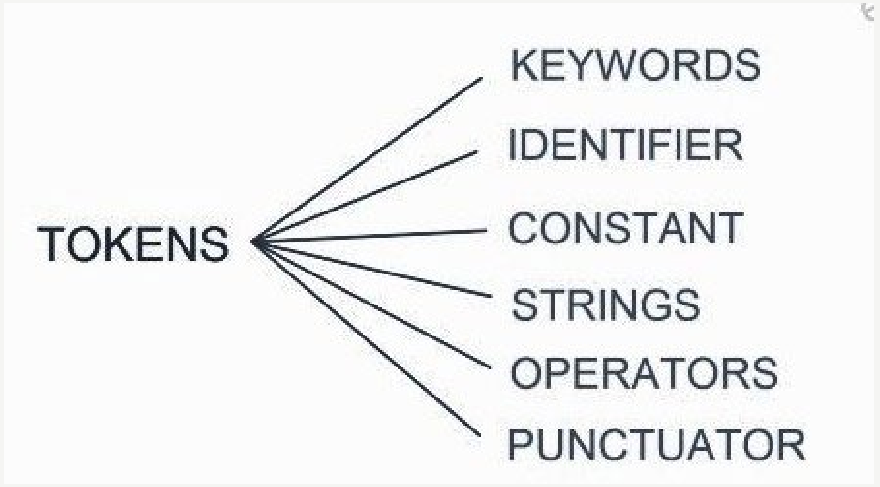
1. **Syntax**

- Syntax refers to the rules that define the structure of a language. Syntax in computer programming means the rules that control the structure of the symbols, punctuation, and words of a programming language.
- Without syntax, the meaning or semantics of a language is nearly impossible to understand.
- For example, a series of English words, such as — subject a need and does sentence a verb — has little meaning without syntax.
- If the syntax of a language is not followed, the code will not be understood by a compiler or interpreter.
- **Compilers** convert programming languages like Java or C++ into binary code that computers can understand. If the syntax is incorrect, the code will not compile.
- **Interpreters** execute programming languages such as JavaScript or Python at runtime. The incorrect syntax will cause the code to fail.

What are Tokens?

- C program consists of various tokens.

Ex. - Keyword, Identifier, Constant, Literal, Operators, or a Symbol.



Identifiers

- Identifier is a name used to identify variable, function, or any user-defined item.

Ex. - num1, getchar(), sum, ab_c etc.

```
num1 = 3;
```

Identifiers

- Rules for writing the names of Identifiers
  - An identifier starts with a letter A to Z, a to z,
  - or an underscore '_'. Followed by zero or more letters, underscores, and digits (0 to 9).

```
num1 = 3;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Keywords

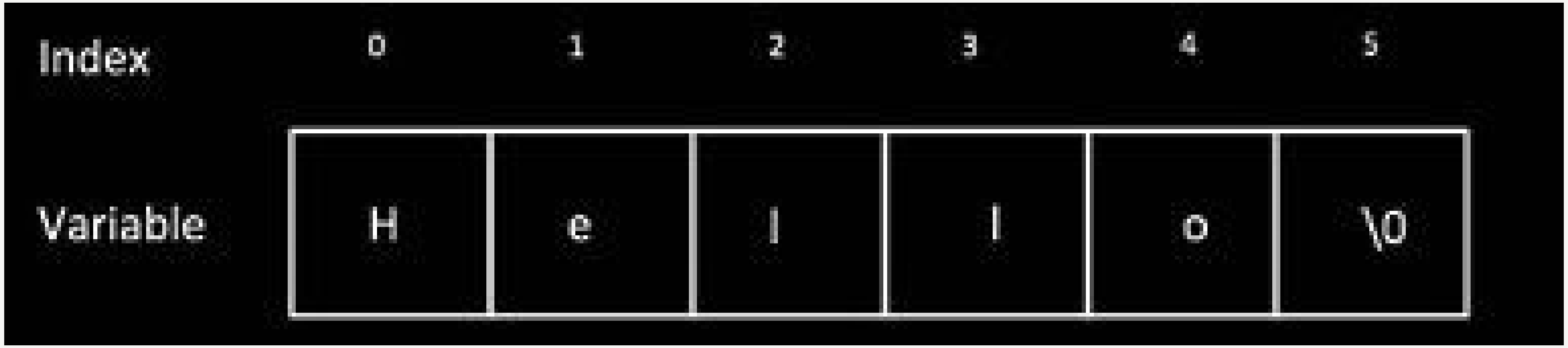
- Some Reserved keywords names given in the next slide cannot be used as identifier
  names.

Keyword

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Strings

- Strings in C are always represented as a set of characters having null character at the end of the
  string.



Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0

Operators

- An operator is **a symbol that tells the compiler to perform specific mathematical or logical functions**.

Ex. - +, -, *, /, ==, ++, --
etc

```
a= b + c;
```

Special Characters

- Various Punctuators are used as a part of Syntax

.

Ex. - [ ], ( ), { }, #, *
etc.

```
return 0;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Constants

- A constant is a value assigned to variable which will remain the throughout the same program

```
const int abc = 5;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

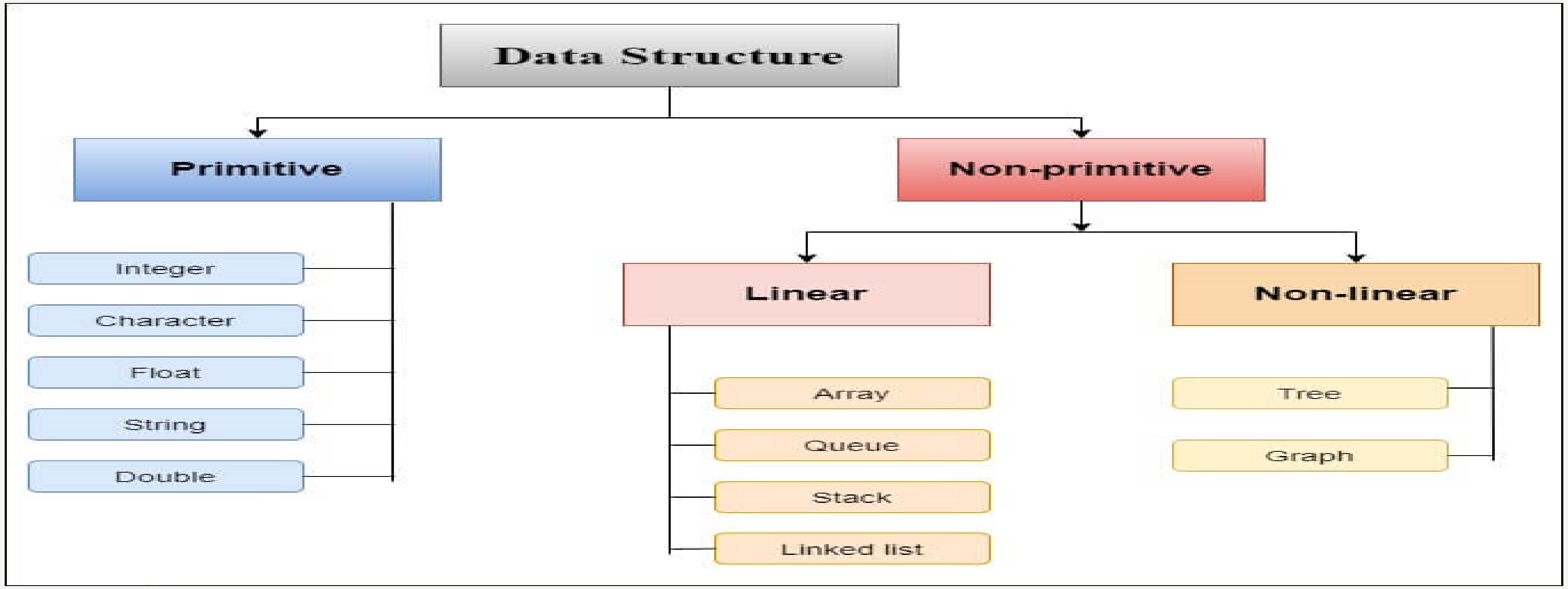
2.Data Structures

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Data Structures

2. Data structures

- A **data structure** is a particular way of organizing data in a computer so that it can be used effectively.



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

What are Data Types?

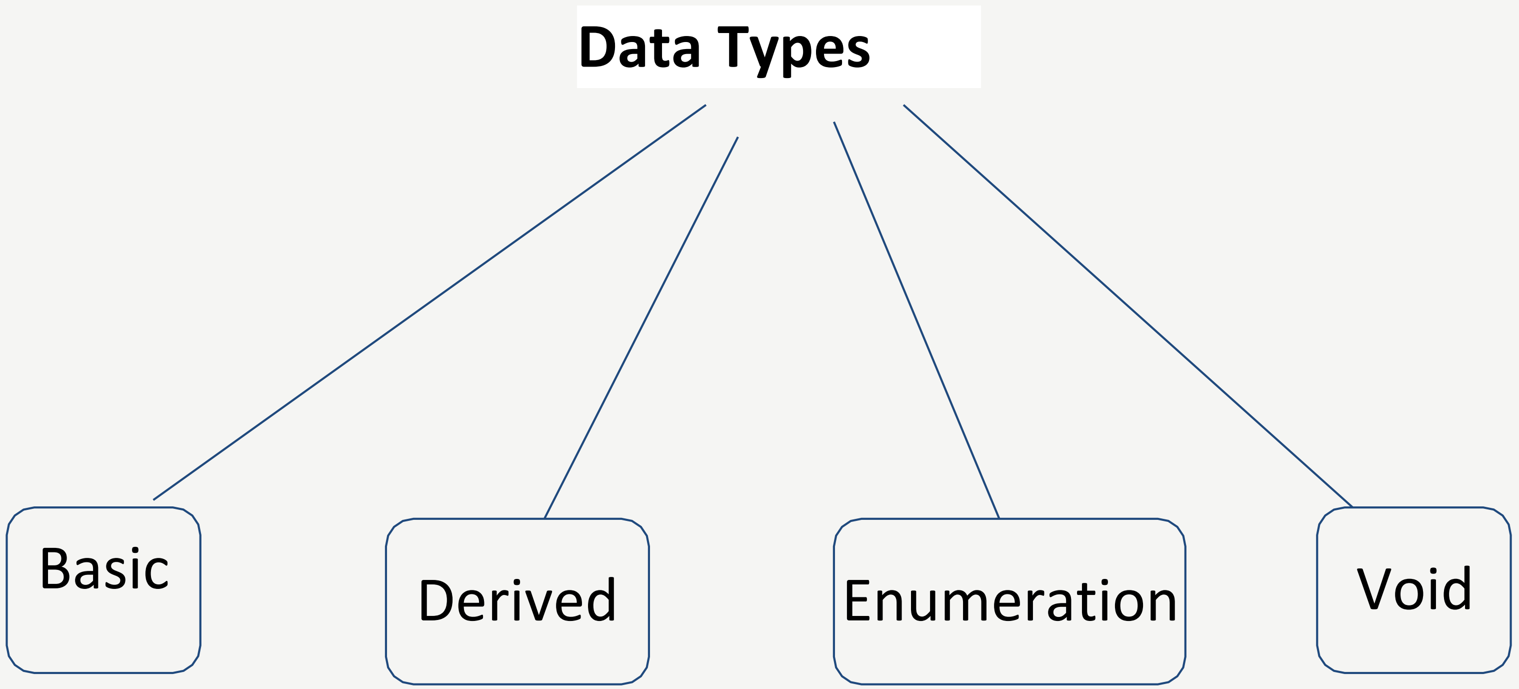
- A data type specifies what type of data a variable can store such as integer, floating, character, etc.

Ex. - int, float, char etc.

```
int res =
5;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Types of Data Types



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Basic Data Types

- The basic data types are integer-and floating-point based.

```
int res = 5;
```

```
char b = 'G';
```

```
float zab1 = 5.34;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

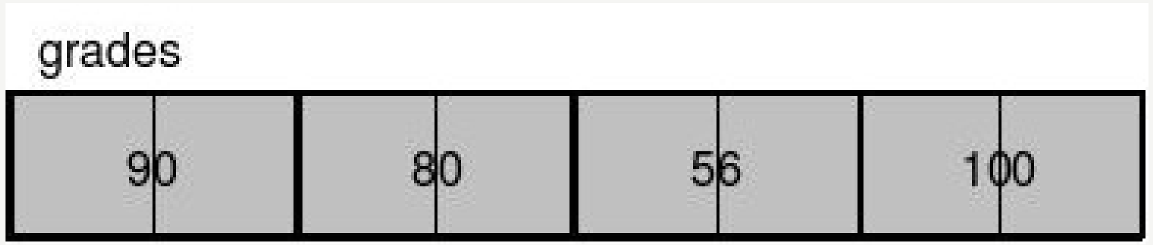
Basic Data Types

Data Types	Memory Size	Range
short int	2 byte	−32,768 to 32,767
signed short int	2 byte	−32,768 to 32,767
unsigned short int	2 byte	0 to 65,535
long int	4 byte	-2,147,483,648 to 2,147,483,647
signed long int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned long int	4 byte	0 to 4,294,967,295
float	4 byte	
double	8 byte	
long double	10 byte	

Derived Data Types

- Array
- s
- Pointers
- Union

  Structure



```
int * p = &n;
```

Void

- Void is an empty data type that has no value.

```
void num();
```

Refer This Example :

https://github.com/TopsCode/Software-Engineering/blob/master/C/dataType.c

https://github.com/TopsCode/Software-Engineering/blob/master/C/inputOutput1.c

https://github.com/TopsCode/Software-Engineering/blob/master/C/simpleProgram.c

https://github.com/TopsCode/Software-Engineering/blob/master/C/inputOutput.c

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

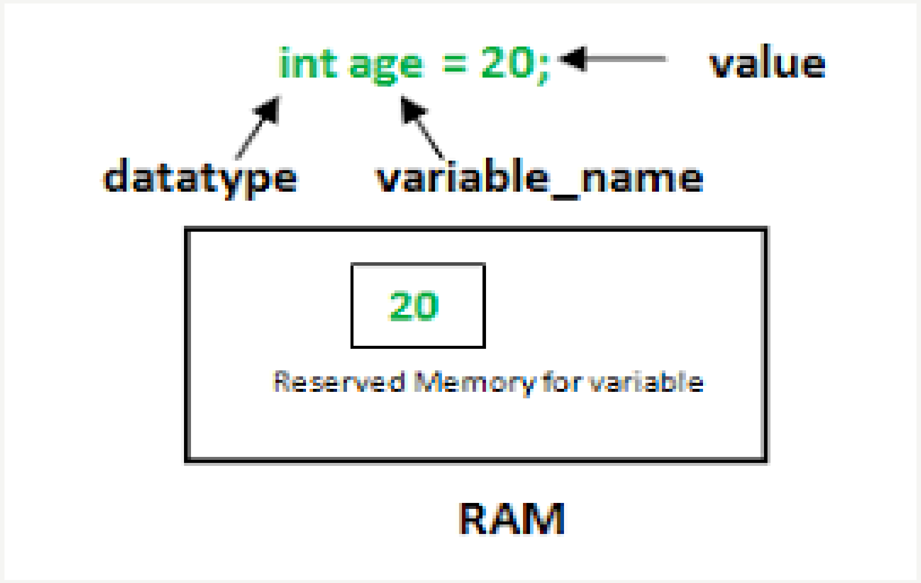
3. Variables

Variables

3. Variables

- Variables are the names you give to computer memory locations which are used to store values in a computer program.
- For example, assume you want to store two values 10 and 20 in your program and at a later stage, you want to use these two values. Let's see how you will do it. Here are the following three simple steps –

i. Create variables with appropriate names.
ii. Store your values in those two variables.
iii. Retrieve and use the stored values from the variables.



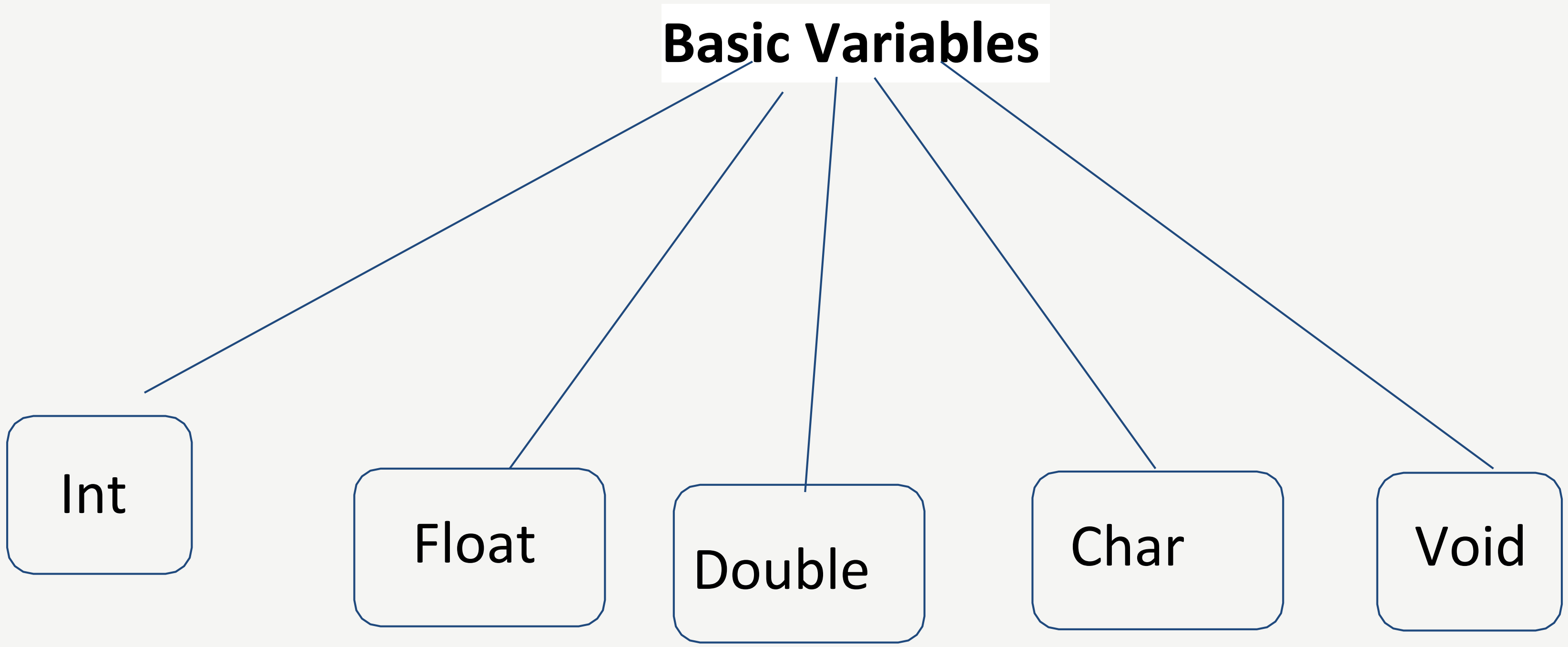
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Deep Dive in Variables

- A variable is a name given to a storage area our programs can use to store values.
- The name of a variable can be composed letters, digits, and the underscore character.

```
int res =
5;
```

Basic Types of Variables



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Variable Declaration

- A variable declaration tells the where and how much storage to for the variable.

```
float   f, salary;
double d;
```

Variable Definition

- A variable definition assigns a value in variable

  .

```
i = 43;
```

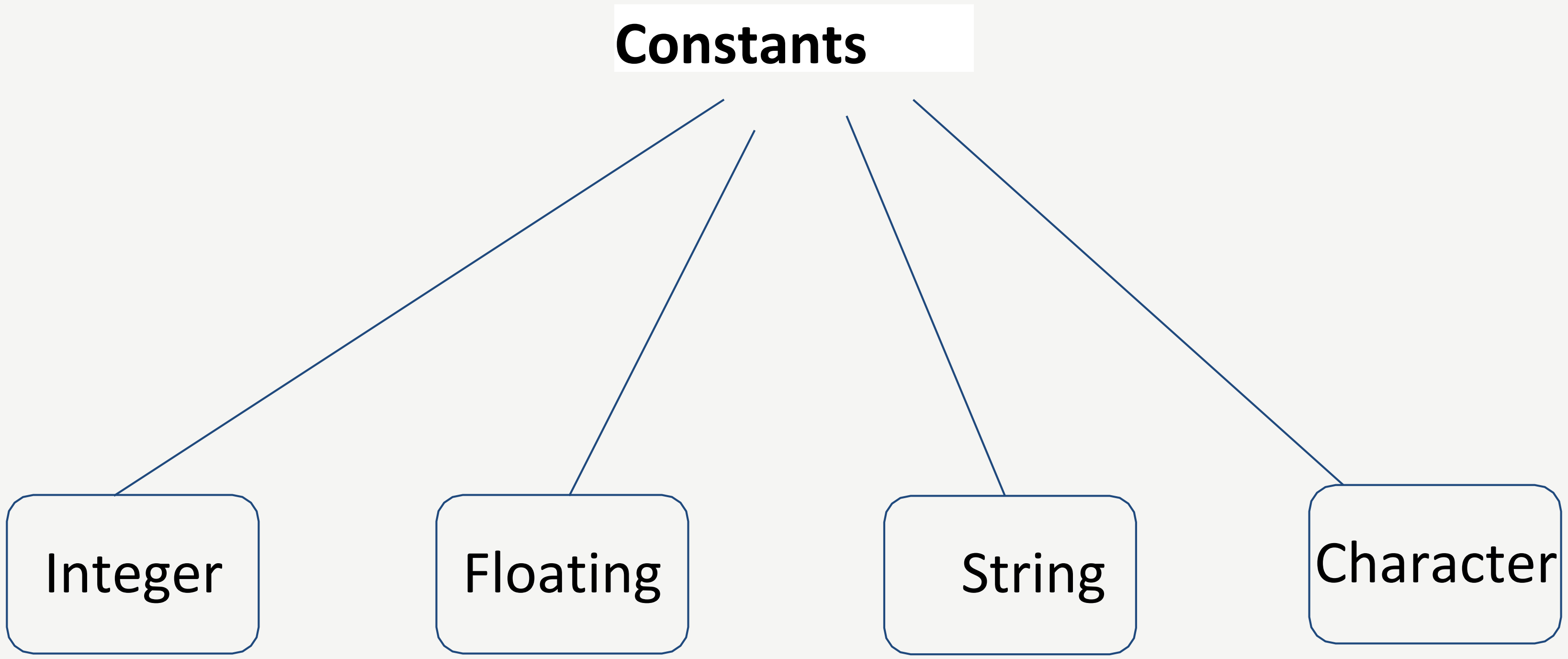
What are Constants?

- Constants are the fixed value in program . Which means that we cannot change it’s value.

```
const int res = 5;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Types of Constants



const keyword

- Using const keyword
- Syntax : const data_type variable_name = value ;

Ex. - `const int a = 10;`

Refer This Example :

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/variable/variables.cpp

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/variable/globalVariable.cpp

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/variable/localVariable.cpp

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/variable/typeCasting.cpp

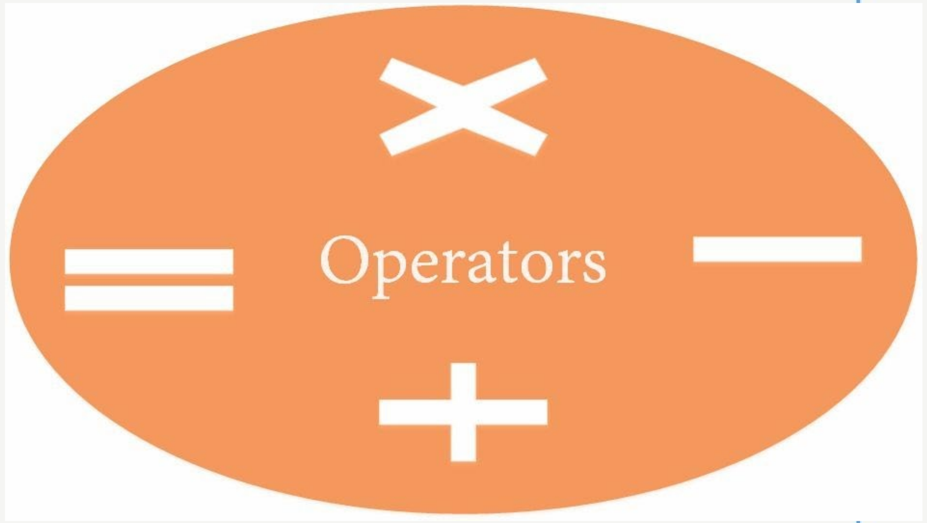
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

4.Operators

What are Operators?

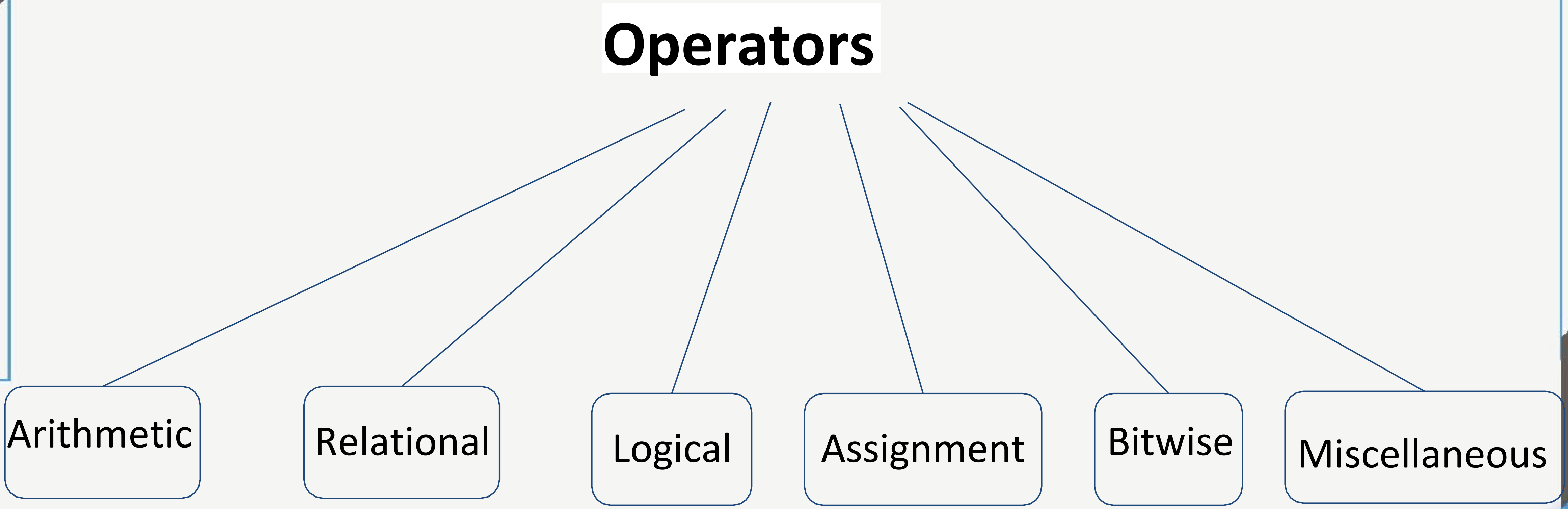
- A symbol that takes one or more operands such as variables, expressions or values and operates on them to give an output.

Ex. - =, +, -, /, *, ==, ++, --, %, etc.



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Types of Operators



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Arithmetic Operators

Operator	Function	Example
+	Addition	var=a+b
–	Subtraction	var=a-b
*	Multiplication	var=a*b
/	Division	var=a/b
%	Modulo	var=a%b
++	Increment	var++
- -	Decrement	var–

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Relational Operators

Operator	Meaning of Operator	Example
==	Equal to	5 == 3 is evaluated to 0
>	Greater than	5 > 3 is evaluated to 1
<	Less than	5 < 3 is evaluated to 0
!=	Not equal to	5 != 3 is evaluated to 1
>=	Greater than or equal to	5 >= 3 is evaluated to 1
<=	Less than or equal to	5 <= 3 is evaluated to 0

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Logical Operators

Operator	Meaning	Example
&&	Logical AND. True only if all operands are true	If c = 5 and d = 2 then, expression ((c==5) && (d>5)) equals to 0.
||	Logical OR. True only if either one operand is true	If c = 5 and d = 2 then, expression ((c==5) || (d>5)) equals to 1.
!	Logical NOT. True only if the operand is 0	If c = 5 then, expression !(c==5) equals to 0.

Assignment Operators

- The Basic type of Assignment operator is ‘=’.
- There are other derived operators

Ex. - *=, -=, /=, += etc.

$$\begin{tikzcd} \texttt{c += 15;} \arrow[d] \\ \texttt{c = c + 15;} \end{tikzcd}$$

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Bitwise Operators

Operators	Meaning of operators
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Miscellaneous Operators

Operator	Description	Example
sizeof()	Returns the size of a variable.	sizeof(a), where a is integer, will return 4.
&	Returns the address of a variable.	&a; returns the actual address of the variable.
*	Pointer to a variable.	*a;
? :	Conditional Expression.	If Condition is true ? then value X : otherwise value Y

Refer this Example:

- **Assignment Operator**
  https://github.com/TopsCode/Software-Engineering/blob/master/C/AssignmentOperator.c

- **Arithmetic Operator Example**
  https://github.com/TopsCode/Software-Engineering/blob/master/C/ArthmeticOperator.c

- **Logical Operator Example**
  https://github.com/TopsCode/Software-Engineering/blob/master/C/logicalOperator.c

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

5. Control & Looping Structures

What is Control Structure?

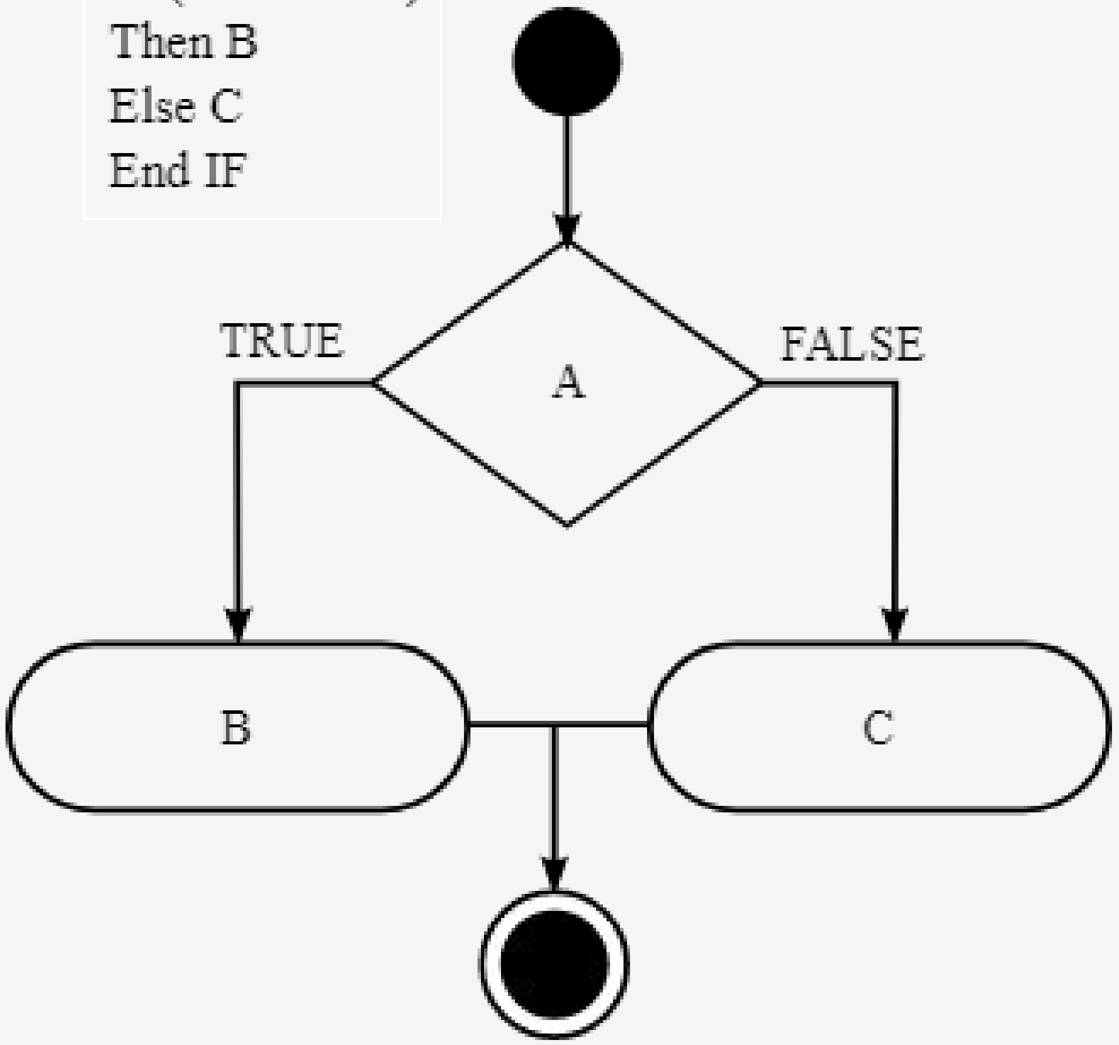
- Real life situations where we have to take condition based decisions by asking ‘if’ questions.

Ex. - if age is above 18, I am allowed drive vehicles or else not allowed.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Flow of Control Structure

```
IF (A = TRUE)
Then B
Else C
End IF
```



Control structure Statements

- Following types of Statements:
  - if
  - if else
  - Nested if
  - Switch
  - Nested switch

If Statements

- if statement is the basic decision making statement
- Used to decide whether a certain statement or block of statements will be executed or not

If Statements

- **Syntax :**

```
if( condition )
{
      statement_1 ;  // true block
      statements
}
statement x ;
```

Example :
https://github.com/TopsCode/Software-Engineering/blob/master/C/if.c

If else Statements

- if else statement allows selecting any one of the two available options depending upon the output of the test condition

If else Statements

- **Syntax :**

```
 if (condition )
{
  statements;   // true
}               statement
 else
{
  statements ;  // false statement
}
```

Example :
https://github.com/TopsCode/Software-Engineering/blob/master/C/ifElse.c

Nested If Statements

- Nested if statement is simply an if statement embedded with an another if statement

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Nested If Statements

- **Syntax**

:

```
if (condition1)
{
  statements ;        // executes when condition1 is
  if ( condition2)    true
  {
  statements ;              // executes when condition2 is
  }                         true
}
```

Example :
https://github.com/TopsCode/Software-Engineering/blob/master/C/ifElseLadder.c

Refer this Example:

- **If statement:**
https://github.com/TopsCode/Software-Engineering/blob/master/C/if.c

- **If-else:**
https://github.com/TopsCode/Software-Engineering/blob/master/C/ifElse.c

- **If else with compound Relational Test:**
https://github.com/TopsCode/Software-Engineering/blob/master/C/compoundRelationalTest.c

- **Nested if-else**
https://github.com/TopsCode/Software-Engineering/blob/master/C/NestedIfElse.c

- **If-else Ladder:**
https://github.com/TopsCode/Software-Engineering/blob/master/C/ifElseLadder.c

Switch Statements

- Switch case statements are a substitute for long if statements that compare a variable to several integer values

Switch Statements

- **Syntax :**

```
switch ( n)
{
case 1 :  break;      // executed when n = 1
case 2 :  break ;     // executed when n = 2
default :             // executed when n doesn't match any case
}
```

Example : https://github.com/TopsCode/Software-Engineering/blob/master/C/switchCase.c

Nested Switch Statements

- Nested Switch Statements occurs when a switch statement is defined inside another switch statement.

Nested Switch Statements

- **Syntax :**

```
switch(ch1)
{
  case ‘A’: printf (“\n This A is part of outer switch “ );
  switch (ch2)
  {
  case ‘A ‘: printf(“\n This A is part of inner switch “);
  break;
   case’ B’ :
           }
  break ;
  case’ B’ :
}
```

Looping Structures

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

What are Loops?

- A loop statement allows us to execute a statement or group of statements multiple times based on a condition

Ex. - printing 1 to 100 on the output screen

Types of Loops

- **Entry Controlled** A condition is checked before executing the loop. It also called as a pre-checking loop.
- **Exit Controlled** A condition is checked after executing the loop. It is called as a post-checking loop.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Entry controlled Loops

1. **For Loops-** It is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

2. **While Loops-** It repeatedly executes a target statement as long as the given condition is true.

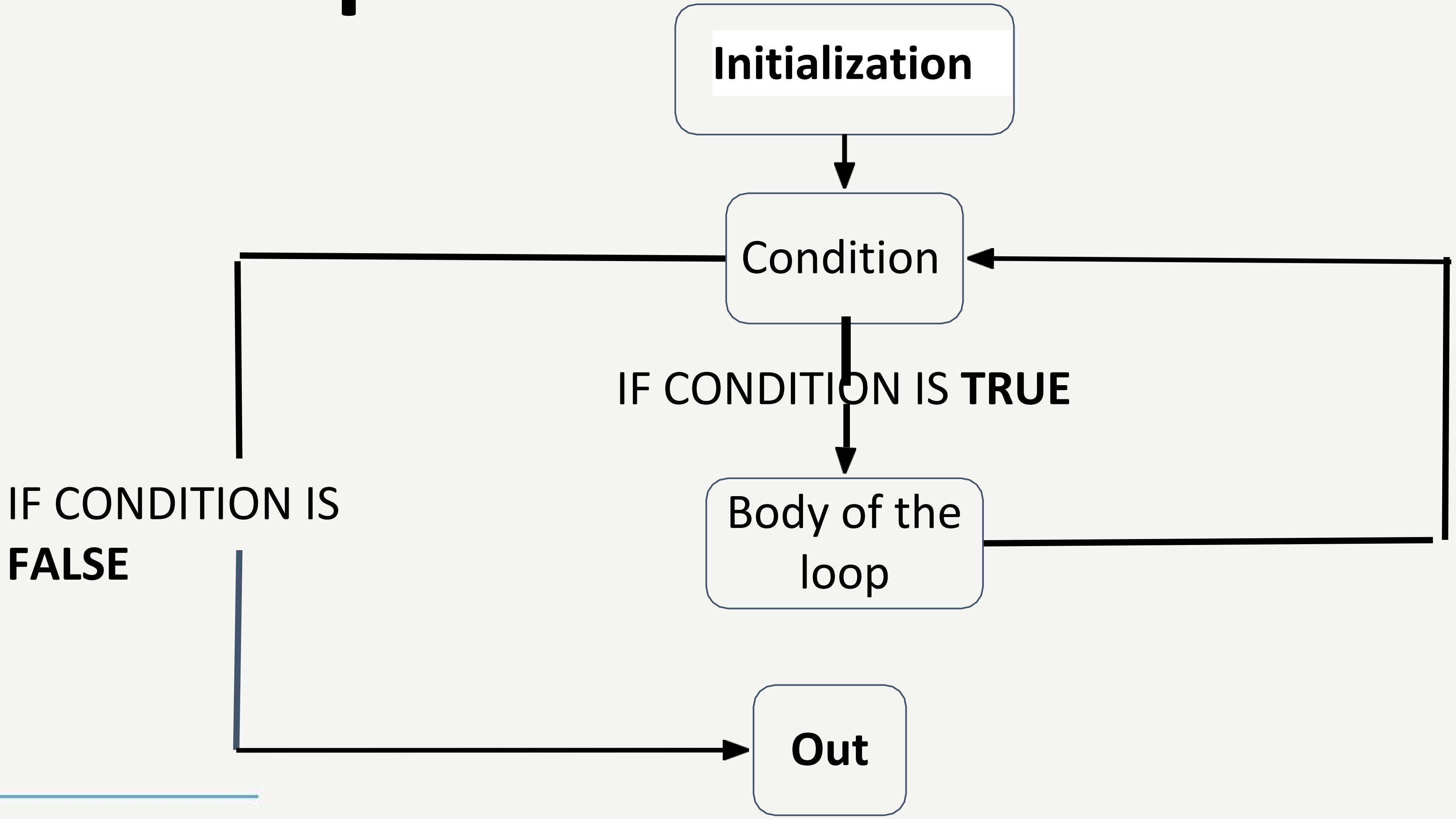
For Loops

- Used to efficiently write a loop that needs to execute a specific number of times.

Syntax :

```
for ( initialization ;test condition ;          increment )
{
      Body of loop
}
```

For Loop Flow Chart



For Loop

- All for loops are executed in following sequence

  : Step 1 : It executes initialization statements

  Step 2 : It checks the condition;

  if true then go to Step

  3 other wise Step 4

  Step 3: Executes loop and go to Step

  2 Step 4 : Out of the Loop

Example:
https://github.com/TopsCode/Software-Engineering/blob/master/C/ForLoop.c

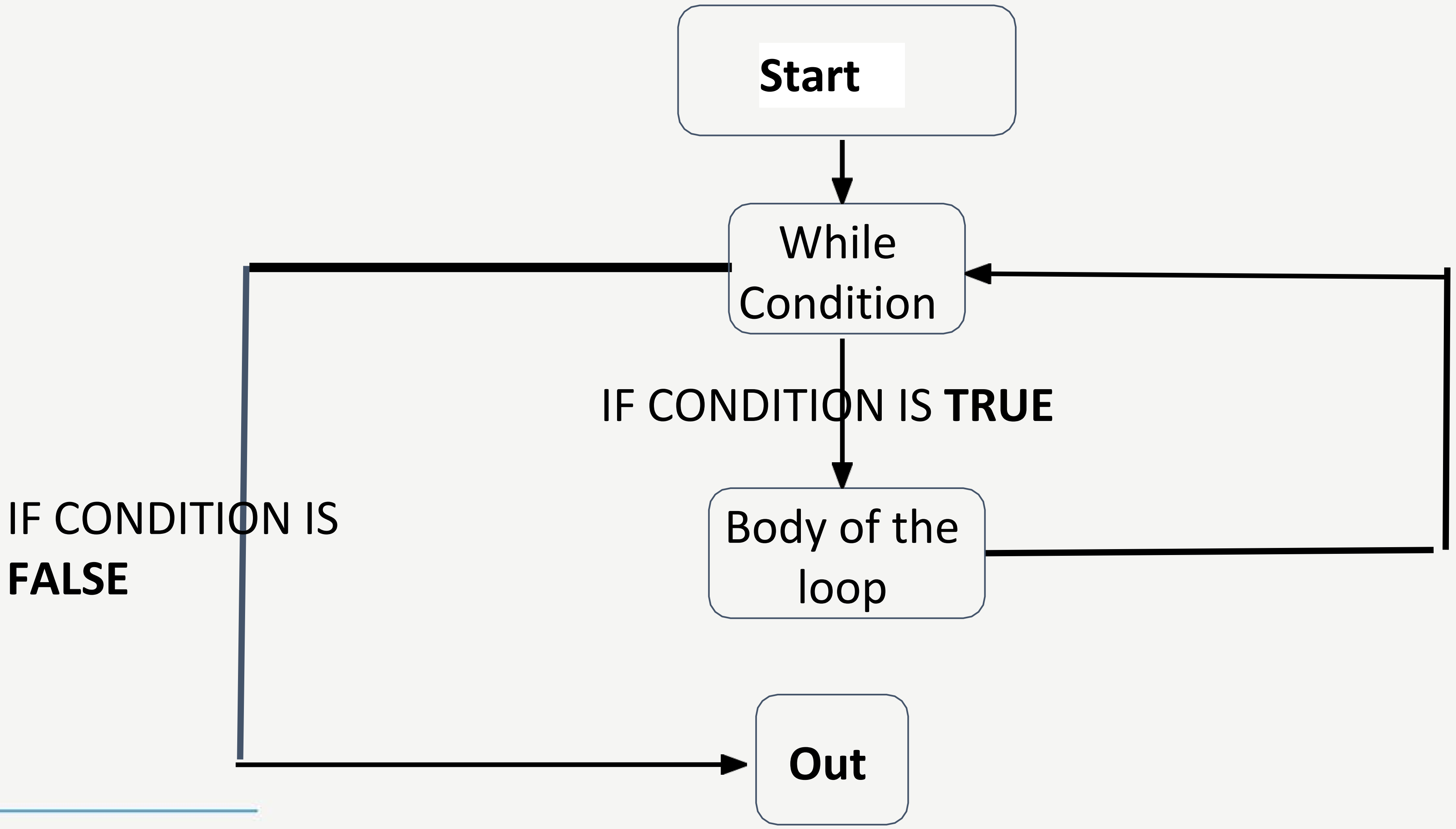
While Loops

- It repeatedly executes a target statement as long as the given condition is true .

Syntax :

```
while (condition )
{
    Body of loop
}
```

While Loop Flow Chart



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

While Loops

- All while loops are executed in following sequence :

  Step 1 : It checks the test
  condition if true then go to
  Step 2 other wise to Step 3

  Step 2 : Executes body of loop and go to Step

  1. Step 3 : Other statements of program.

Example :
https://github.com/TopsCode/Software-Engineering/blob/master/C/while.c

Exit controlled Loops

1. **Do-while Loops-** Do-while loop is similar to while loop , except the fact that it will execute once even if condition is false.

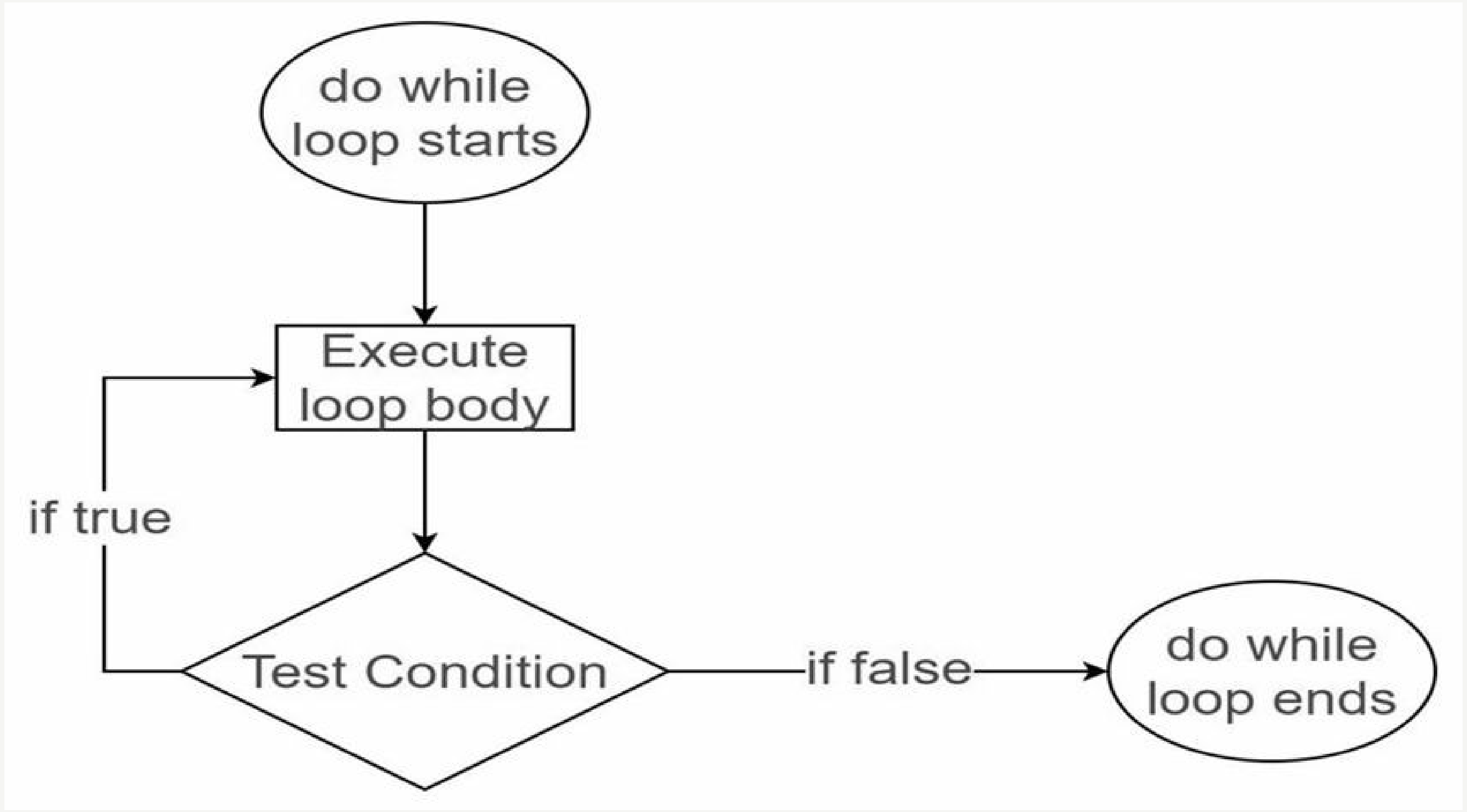
Do-while Loops

- **Syntax :**

```
do
      {
body of loop
      } while (condition ) ;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Do-while Loops



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Do-while Loops

- All do while loops are executed in following sequence :

  Step 1 : Executes the body of loop and go to Step 2

  .

  Step 2 : It checks the test

  condition if true then go to

  Step 1 otherwise go to

  Step 3.

  Step 3: Other statements of the program .

Example:
https://github.com/TopsCode/Software-Engineering/blob/master/C/DoWhile.c

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Loops

Jumping Statements:

1. Goto Statement.
2. Break Statement.
3. Countinue Statement

Loops

The GOTO statement:

- By using this goto statements we can transfer the control from current location to anywhere in the program.
-
- To do all this we have to specify a label with goto and the control will transfer to the location where the label is specified.

```
goto label;
      ----------
      ----------
label:
      ----------
      ----------
```

Refer this Example:
https://github.com/TopsCode/Software-Engineering/blob/master/C/got.c

Loops

The Break Statement:

- The break statement is used inside loop or switch statement.

- When compiler finds the break statement inside a loop, compiler will abort the loop and continue to execute statements followed by loop.

Syntax: break ;

Refer this Example:
https://github.com/TopsCode/Software-Engineering/blob/master/C/Break.c

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Loops

Continue statement:

- The continue statement is also used inside loop.

- When compiler finds the continue statement inside a loop, compiler will skip all the following statements in the loop and resume the next loop iteration.

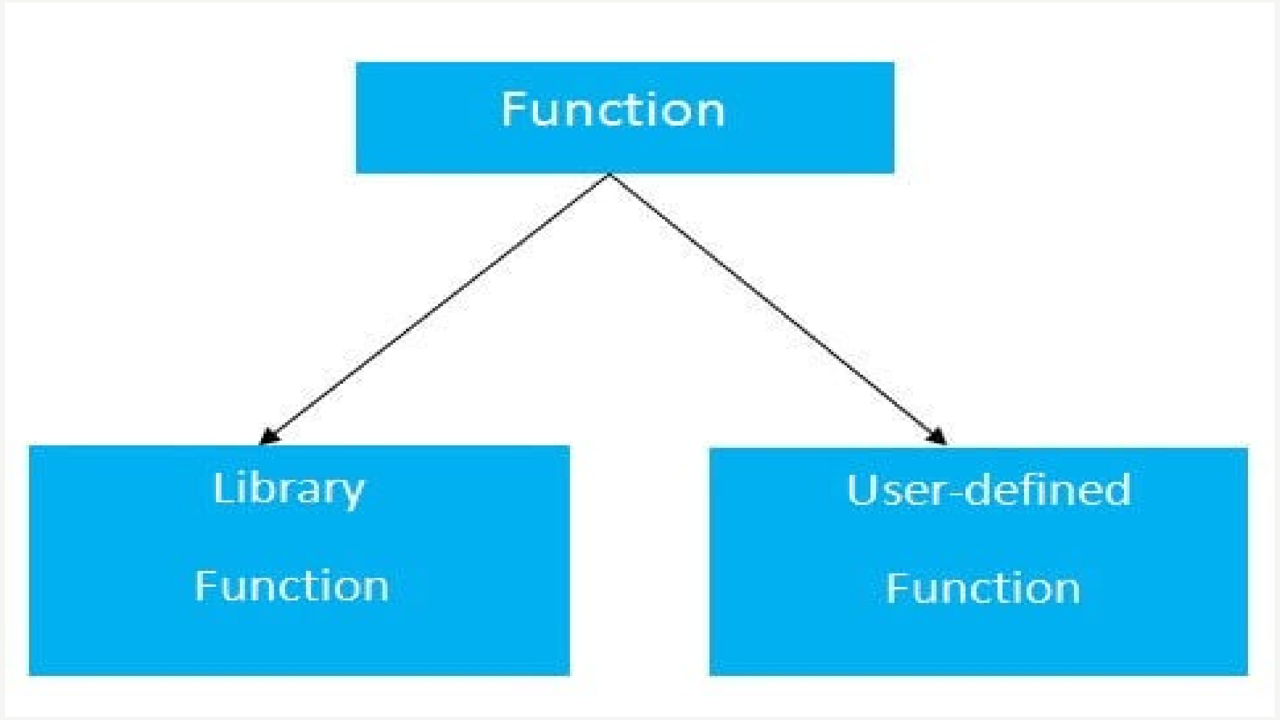
Syntax: continue ;

Refer this Example:
https://github.com/TopsCode/Software-Engineering/blob/master/C/continue.c

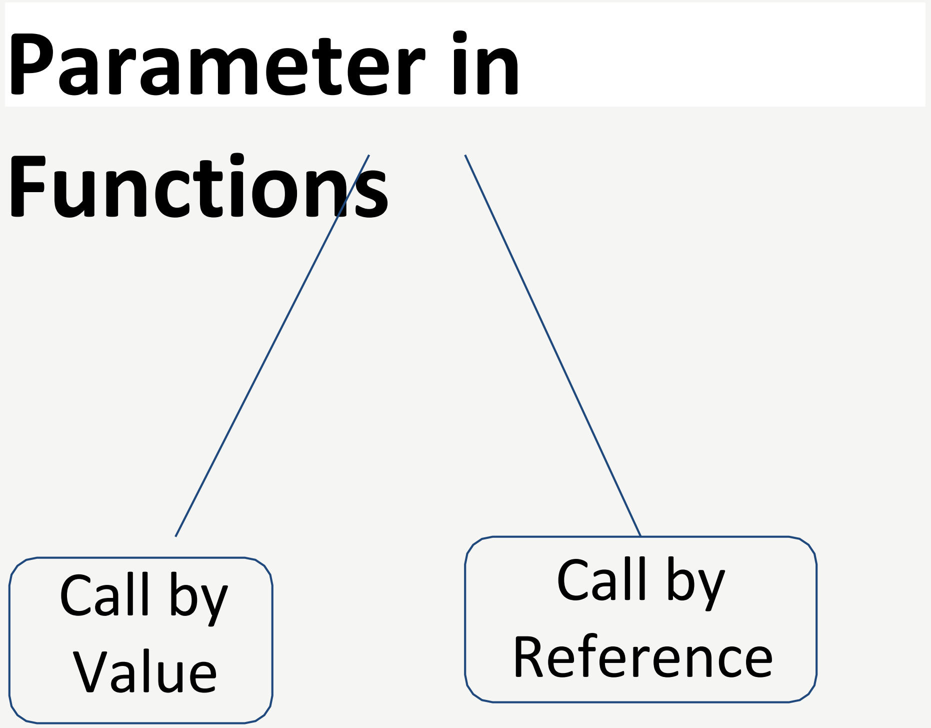
6. Functions

6. Functions

- A function is a block of organized code that is used to perform a single task. They provide better modularity for your application and reuse-ability. Depending on the programming language, a function may be called a subroutine, a procedure, a routine, a method, or a subprogram.



Types of Parameters



Call by Value

- A method of passing parameters, where it copies the actual value into formal parameter

- Changes made to the parameter inside the function have no effect on the actual parameters

Call by Reference

- A method of passing arguments which copies the address of an argument into formal parameter.
- Changes made to the parameter affect the passed argument.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

What are Functions?

- A function is a set of statements that take inputs, do some specific computation and produces output

  Ex. - main( ), sum( ), swap( ) etc.

```
int num1( );
```

Example:
https://github.com/TopsCode/Software-Engineering/blob/master/C/Function1.c

Parts of Functions

- FUNCTIONS-
    - Function Definition
    - Function Call
    - Function Declaration

```
int num1(
);
```

Functions Definition

- Syntax of Function Definition-

```
return_typefunction_name( parameter list )
{
    body of the function
}
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Components in Definition

Return_type : It is data type of value which function will return.If function does not return any value data type will be void().

```
return_type    function_name( parameter list )
{
    body of the function
}
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Components in Definition

Function_name : It is the name given to the function by the programmer.

```
return_type    function_name( parameter list )
{
    body of the function
}
```

Components in Definition

Parameter list : This list refers to type, order and number of parameters of the function. A function can have no parameters also.

```
return_type    function_name( parameter list
)
{
    body of the function
}
```

Components in Definition

Body of function : It is collection of statements that define the working of the function.

```
return_type    function_name( parameter list )
{
   body of the function
}
```

Function Declaration

- It tells the compiler about the function name and how to call the function.

- Syntax :

  return_type function_name (parameter_list) ;

Function Declaration

- Only type is required in function declaration , we can skip the parameter name

- Example:

```
int add ( int , int );
```

Function Call

- To use a function, we have to call that function to perform the given task.

Function Call

- When a program calls a function, the compiler gets redirected towards the function definition

- Function Call simply pass the required parameters along with the function name

What are Function Parameters

- Parameters are the variables that are taken as input to perform the function

```
int max(int num1, int
num2);
```

Scope of a Variable

What is Scope?

- Scope is the part of the program where a defined variable can have its existence and beyond that it cannot exist

Three Scopes

- **Three places where variables can be declared-**
  - Local Variables : Declared inside a block
  - Global Variables : Declared outside all functions
  - Formal Parameter : Declared in function definition

Local Variables

```
#include<stdio.h
    >  void
    {    main() int x , y , z              //        local
                        ;                  variable
              x=10 ;
              y =20 ;
              z =x + y;      %d       %d ” , x, y,
    }         printf(“ %d   z ) ;
```

Global Variables

```
#include<stdio.h
>  int z;                                    // global variable

void  main ()
     {          int x , y ;                  //   local variable
                x=10 ;
                y =20 ;
                z =x + y   ;
                printf(“ %d   %d   %d ” , x, y, z ) ;
     }
```

Formal Parameters

```
void swap(int a, int b)
{
        int  temp;
        temp = a;
        a = b;
        b = temp;
}
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Refer This Example

https://github.com/TopsCode/Software-Engineering/blob/master/C/functions/pass_by_reference.c

https://github.com/TopsCode/Software-Engineering/blob/master/C/functions/pass_by_value.c

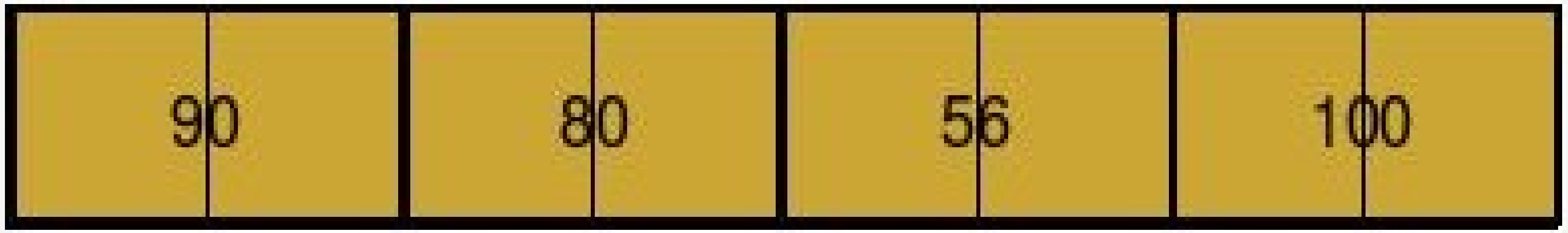
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

7. Arrays & Strings

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

What are Arrays?

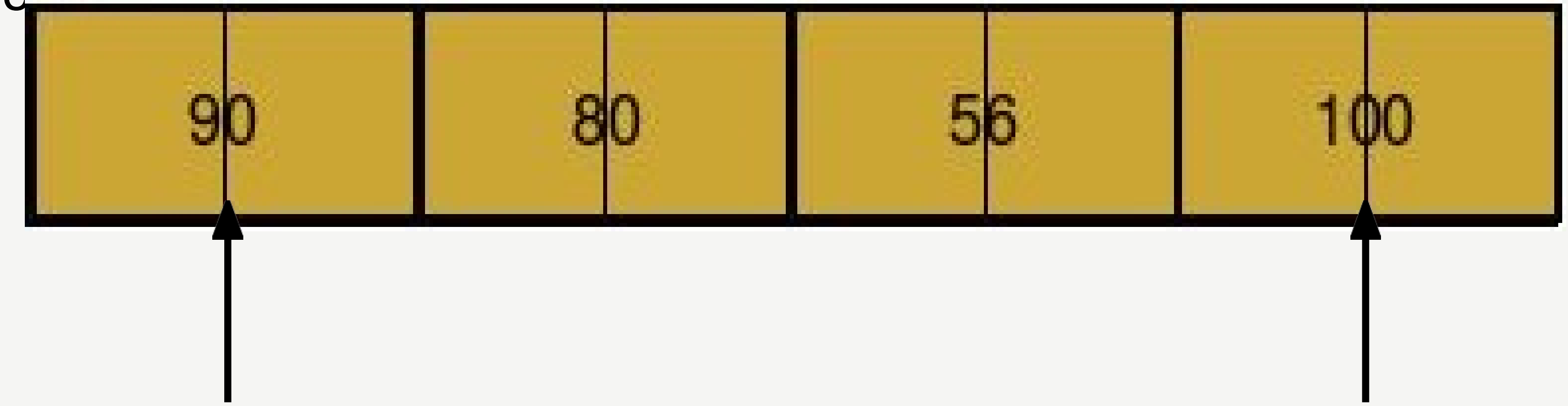
- An array is used to store a collection of data, and it is often used as a collection variables of the same type.



0 1 2 3

What are Arrays?

- All arrays consist of contiguous memory locatio



First Element

Last Element

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Declaring Arrays

- In declaration we specify the type of element and size of the array element

- **Syntax :**

  data_type array_name [size] ;

Ex. - int roll[20] ;

Initializing Arrays

- We can initialize an array in C either one by one or using single statement

Ex. - double balance [ ] = { 1000.0, 2.0, 3.4, 7.0, 50.0};

OR

balance[4] = 50.0 ;

Accessing Arrays

Accessing Array Elements

- An element is accessed by placing the index of the element within the square brackets after the name of the array

Ex. - double income = balance[9];

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Accessing Struct Members

- The individual members of structure can be accessed using the member access operator (.)
- It connects the structure variable and the structure member.

- **Syntax:**

  structure_variable.structure_member

https://github.com/TopsCode/SoftwareEngineering/blob/master/c%2B%2B/Array/1dArray.cpp

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Types of Arrays

Types of Arrays

- Single/ One Dimensional Array
- Multi Dimensional Array

Single Dimensional

```
int roll[20] ;
```

Example :
https://github.com/TopsCode/Software-Engineering/blob/master/C/Array/example1.c

Multi Dimensional

```
int roll [20][12]...;
```

Multi Dimensional

- C programming language provides us multi-dimensional array.
- Out of which we will discuss Two Dimensional Array

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Multidimensional Arrays

What are they?

- Multidimensional arrays are arrays of arrays
- **General Form-**

```
data_type
        array_name[size1][size2]....[sizeN];
```

Example

```
int x[2][3][4] =
 {
     { {0,1,2,3}, {4,5,6,7}, {8,9,10,11} },
     { {12,13,14,15}, {16,17,18,19}, {20,21,22,23} }
 };
```

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/Array/2DArray.cpp

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Strings

What are Strings?

- A string is an array of characters stored in a consecutive memory locations

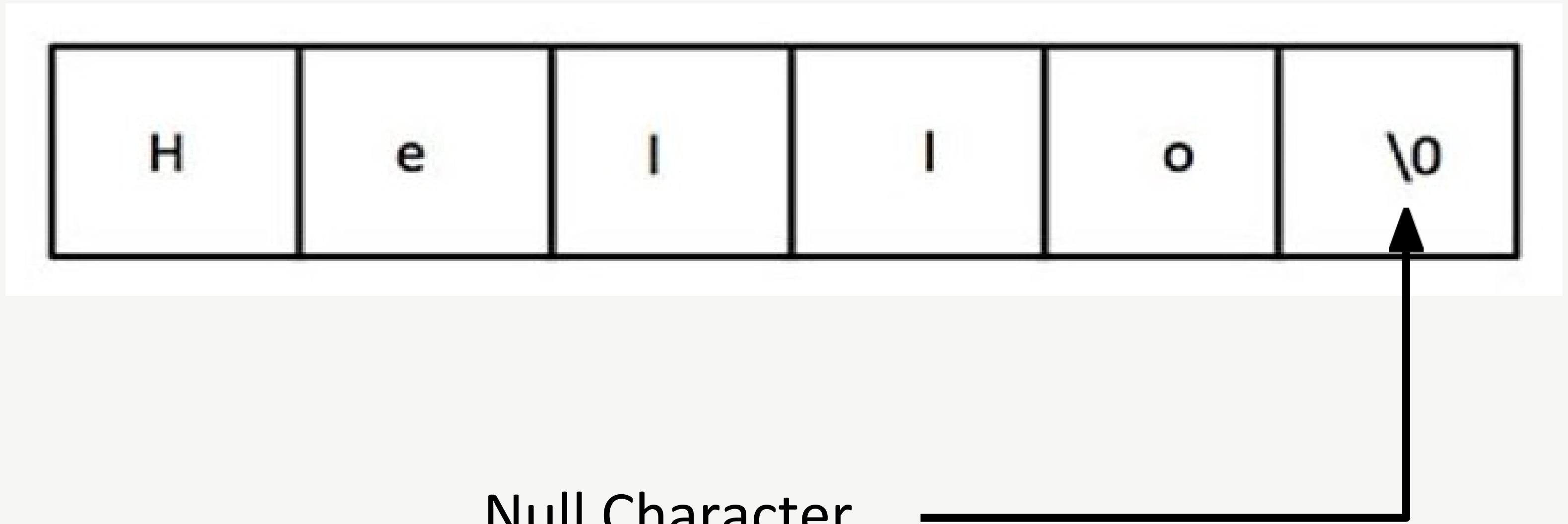
- The ending character is always the null character ‘\0’. It acts as string terminator.

- **Syntax :**

```
char     string_name [ length ] ;
```

Strings

- The compiler automatically places the ‘\0’ at the end of the string when we initializes the array



Null Character

String Functions

Function	What It Does
strcpy()	Copies one string to another.
strlen()	Returns the length of a string, not counting the NULL character

String Functions

Function	What It Does
strcmp()	Compares two strings. If the strings match, the function returns 0.
strcat()	Appends one string to another, creating a single string out of two.

String Functions

Refer this Example:

https://github.com/TopsCode/Software-Engineering/blob/master/C/strlen.c

https://github.com/TopsCode/Software-Engineering/blob/master/C/stringFunctin.c

Module 3
[OOP Concepts]

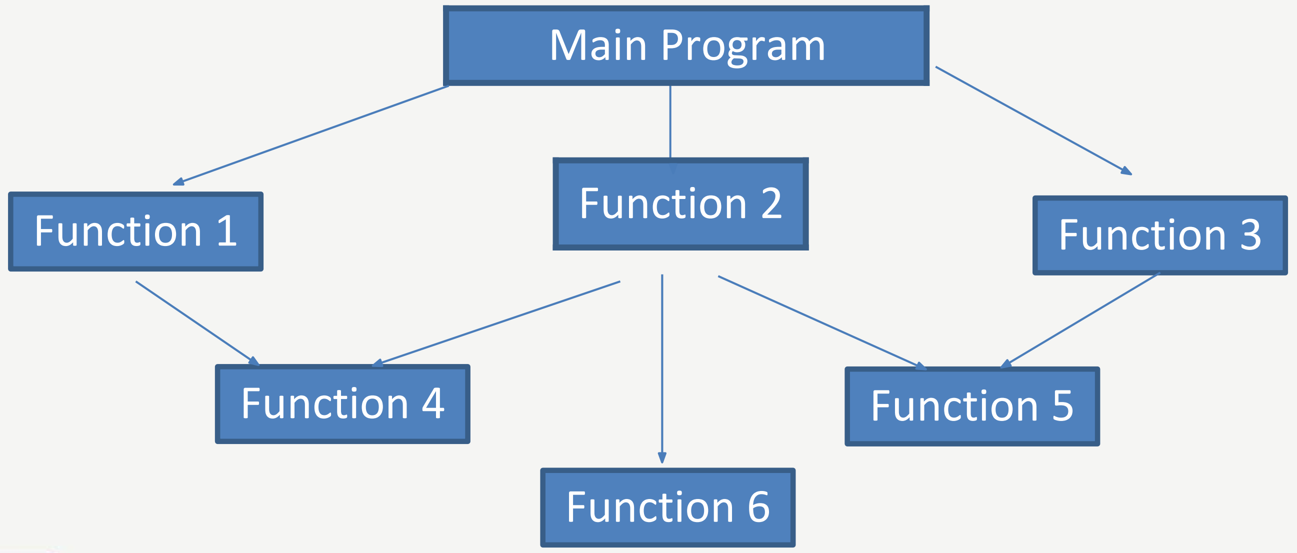
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Procedure Oriented Programming &
Object Oriented Programming

Procedure Oriented Programming

In Procedure Oriented programming , the problem is viewed as sequence of things to be done such as reading, calculating and printing.
The number of functions are return to accomplish such task, i.e. the focus is on functions.

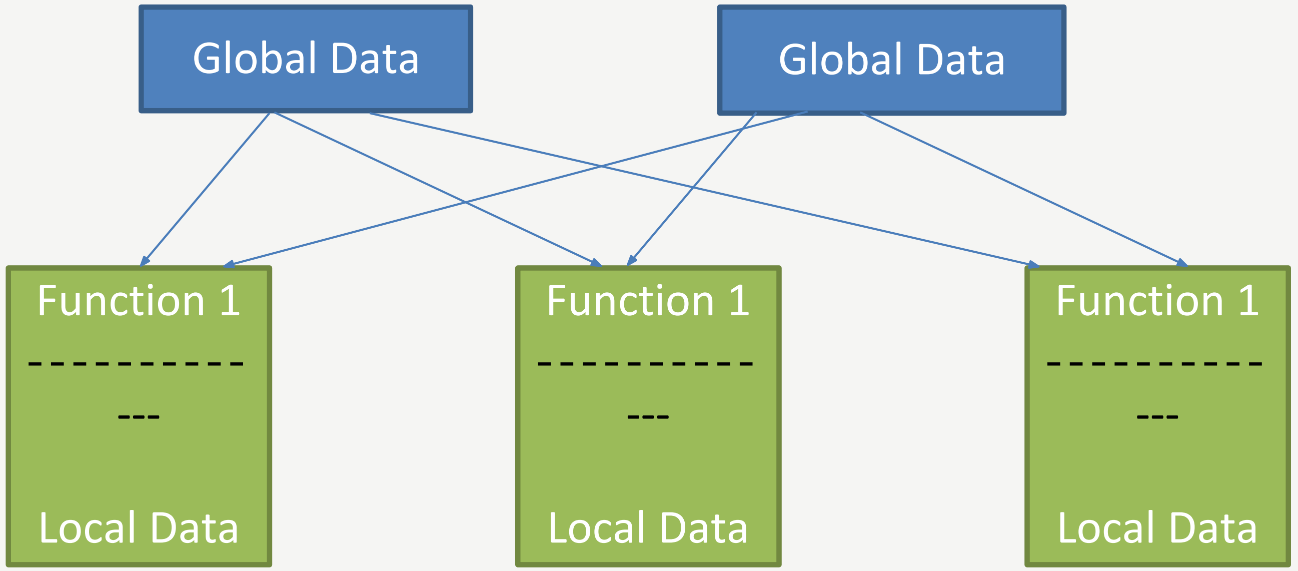
The typical program structure for procedure programming is shown below :



Procedure Oriented Programming

Characteristics of Procedure Oriented Programming.

- ❖ Emphasis is on doing things
- ❖ Large programs are broken into small known as functions
- ❖ Most of the function share global data.
- ❖ Data move openly around the system from function to function.
- ❖ Follows Top Down approach in program design


Global Data
Global Data
Function 1

Local Data
Function 1

Local Data
Function 1

Local Data

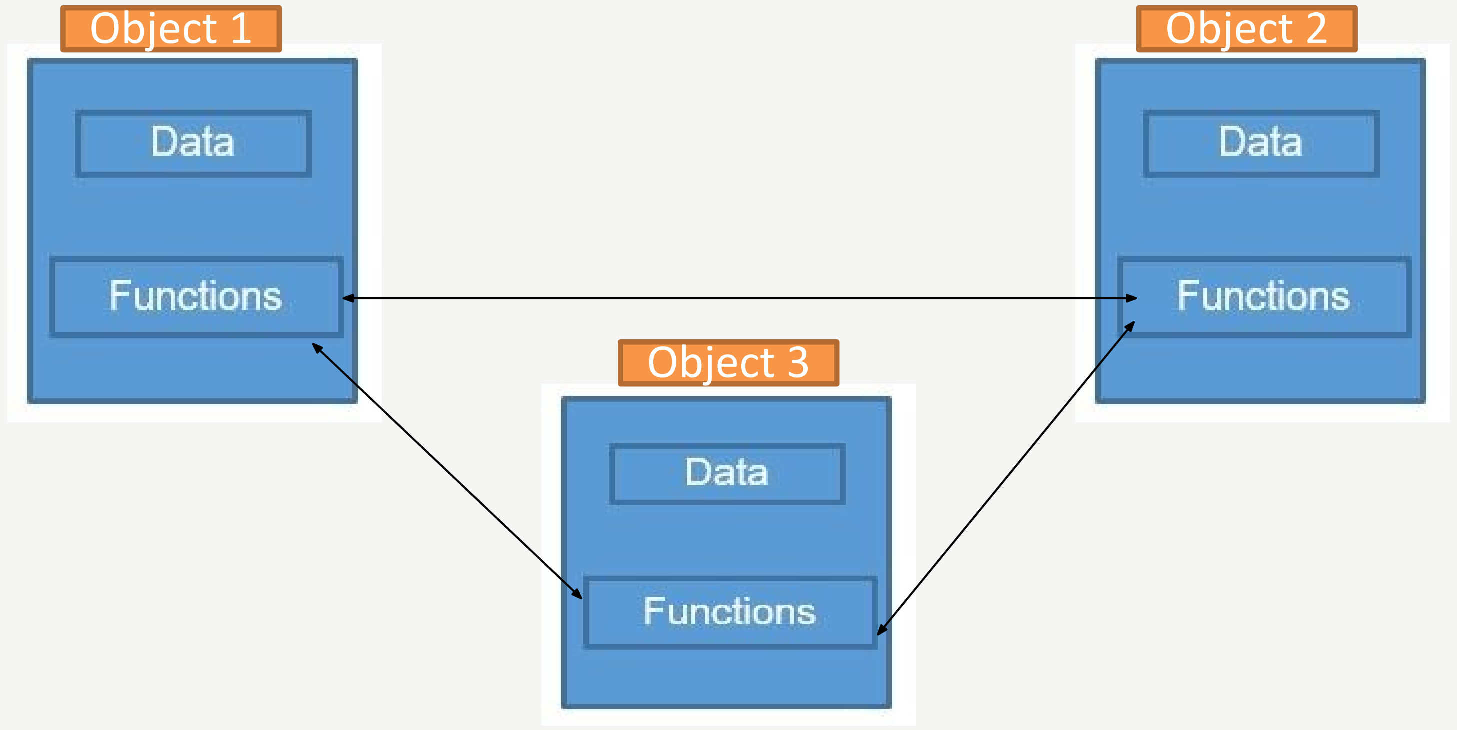
Object Oriented Programming

In order to remove some of the flaws of POP, OOP came into existence.

OOP treats data as critical element in program development and does not allow it to flow freely around the system.

It ties the data more closely to the function that operates on it.

Object Oriented Programming allows decomposition of program into a number of entities called objects and then builds data and function around these objects.


Object 1
Data
Functions
Object 2
Data
Functions
Object 3
Data
Functions

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Object Oriented Programming

Characteristics of Object Oriented Programming:

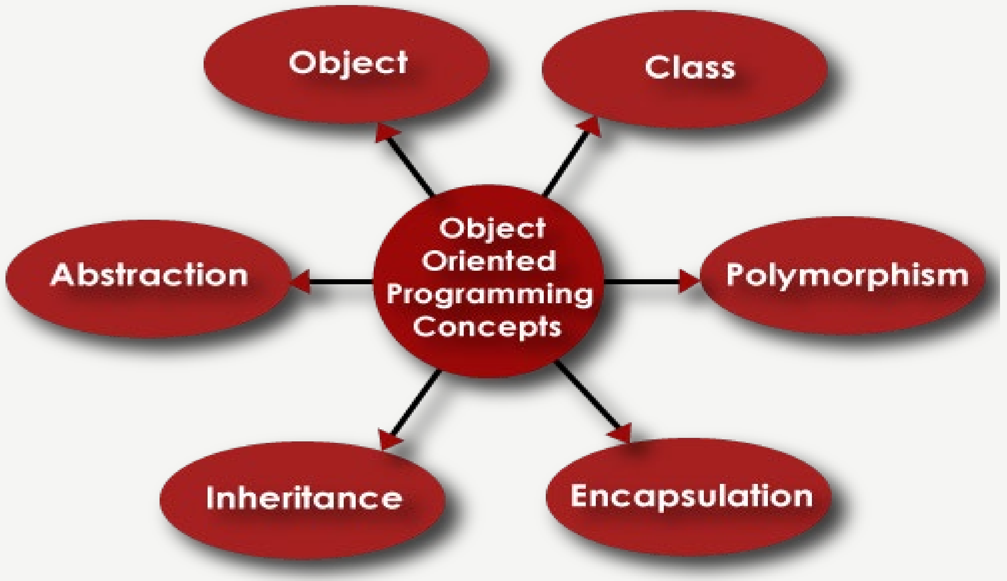
- ❖ Emphasis on data rather than procedure.
- ❖ Program are divided into objects.
- ❖ Data is hidden and cannot be accessed by external functions.
- ❖ Objects may communicate with each other through functions.
- ❖ Follows bottom up approach in program design.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Basic Concepts of OOP

Some of the basic concepts of object oriented programming are:

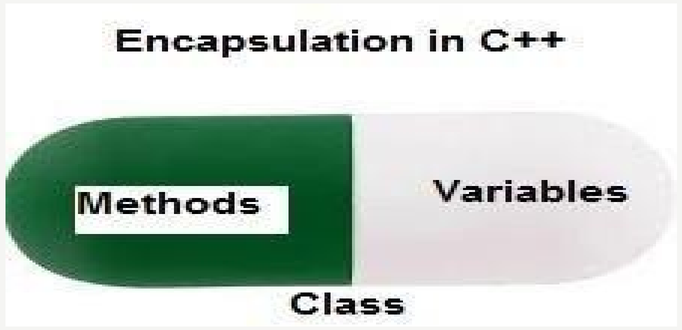
- ❖ Objects
- ❖ Classes
- ❖ Data abstraction and encapsulation
- ❖ Polymorphism
- ❖ Inheritance
- ❖ Dynamic Binding



Some of the basic concepts of object oriented programming are:

❖ **Encapsulation:**

In normal terms, Encapsulation is defined as wrapping up of data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them.



❖ **Abstraction:**

Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

❖ Polymorphism:

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.
C++ supports operator overloading and function overloading.

❖ Inheritance:

- The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programing.
- Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

❖ Classes

- These contain data and functions bundled together under a unit. In other words class is a collection of similar objects. When we define a class it just creates template or Skelton. So no memory is created when class is created. Memory is occupied only by object.for eg. :

Fruit is class of apple.

❖ Objects

- In other words object is an instance of a class.

NOTE: In other words classes acts as data types for objects.

❖ Member functions

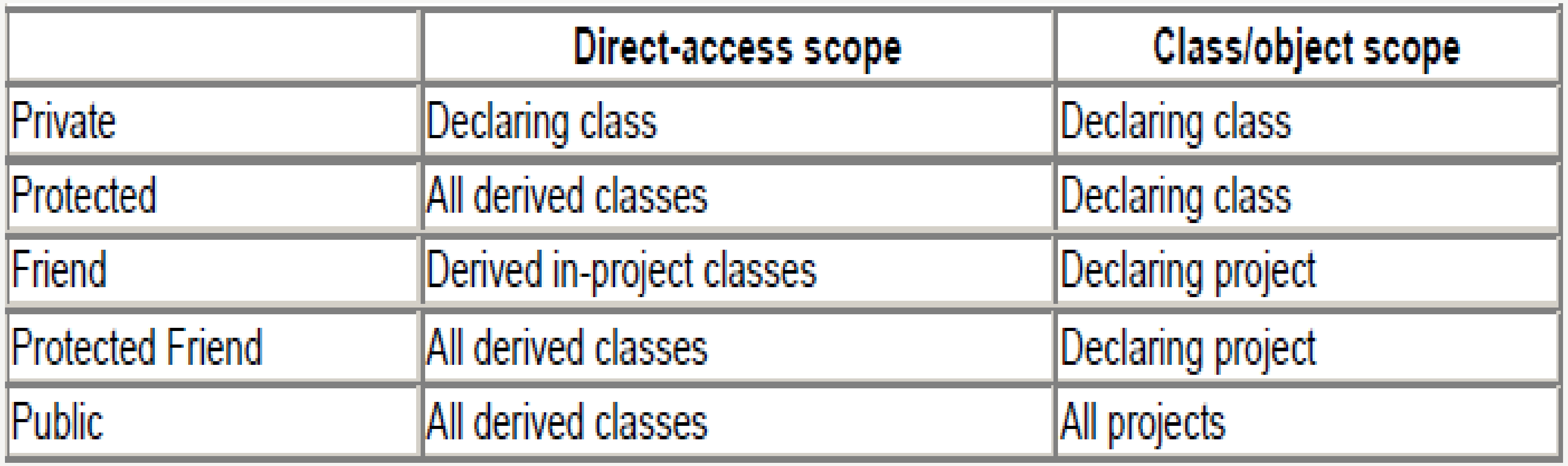
- The functions defined inside the class as above are called member functions.

Example:

```
class classname
{
   variable declarations;
   Data Functions;
};
main ( )
{
   classname
objectname1,objectname2,..;
}
```

Types of accesscifiers

Private	Public	Protected
Only for that class can access	Other class can also access.	Only immediate inheritance class can access.



	Direct-access scope	Class/object scope
Private	Declaring class	Declaring class
Protected	All derived classes	Declaring class
Friend	Derived in-project classes	Declaring project
Protected Friend	All derived classes	Declaring project
Public	All derived classes	All projects

Example of accesscifiers:

```
Class classname
{
      private:
      datatype data;

      public:
      Member functions
 };
 main ( )
 {
     classname objectname1,objectname2,..;
 }
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

- **How to Access Class Members**

It is possible to access the class members after a class is defined and objects are
created.
General syntax to access class member:

```
Object_name.function_name (arguments);
```

```
class exforsys
{
  int a, b;
  public:
  void sum(int,int);
} e1;
e1.sum(5,6);
```

• Scope Resolution Operator

Member functions can be defined within the class definition or separately using **scope resolution operator, ::.** Defining a member function within the class definition declares the function inline, even if you do not use the inline specifier. So either you can define Volume() function as below:

```
class Box
{
        public:

        double length; // Length of a box
        double breadth; // Breadth of a box
        double height; // Height of a box

        double getVolume(void)
        {
                return length * breadth * height;
        }
};
```

Refer This Example:

Simple Class:

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/class/class1.cpp

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/class/class2.cpp

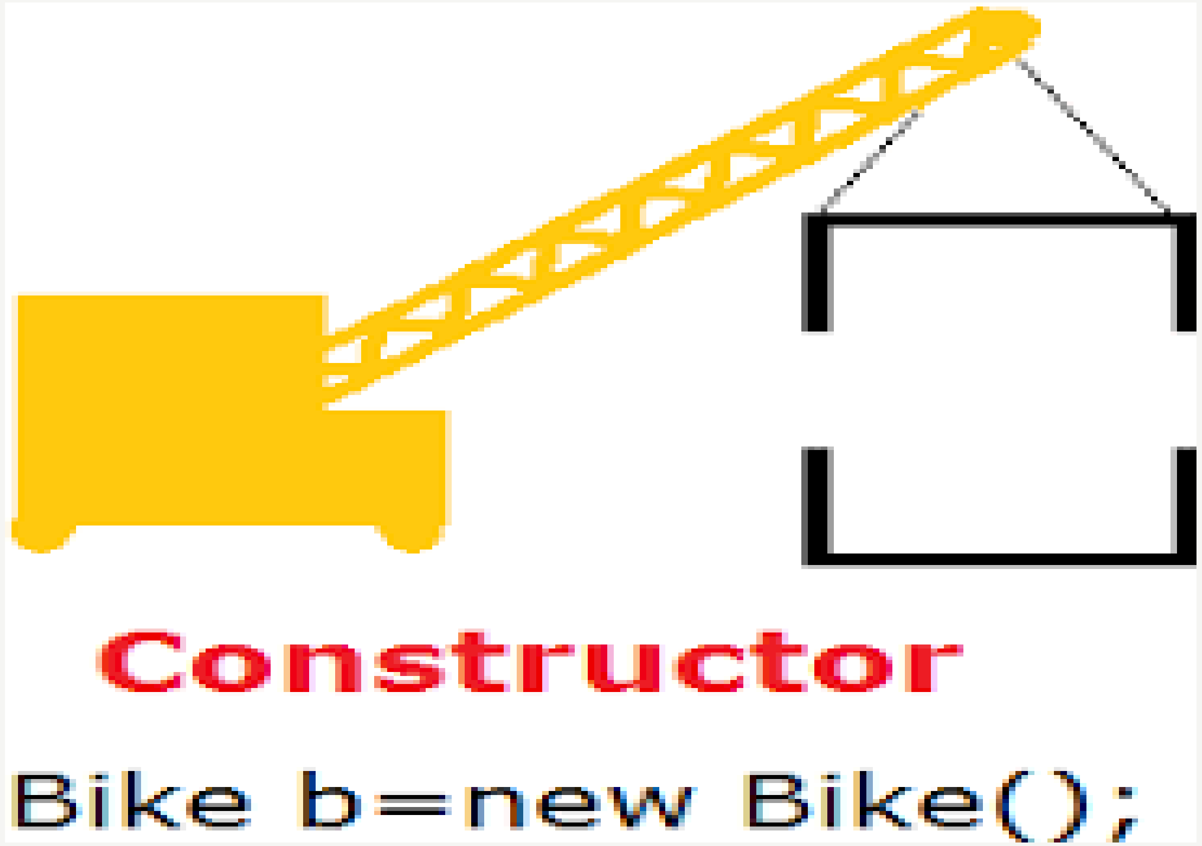
Constructor

- It is a member function which initializes a class.
- A constructor has:
  - (i)the same name as the class itself
  - (ii)no return type

- A constructor is called **automatically** invoked whenever a new instance of a class is created.
- You must supply the arguments to the constructor when a new instance is created.
- If you do not specify a constructor, the compiler generates a default constructor for you (expects no parameters and has an empty body).

Types of Constructor

- Simple(Default) Constructor
- Parameterized Constructor
- Copy Constructor



Refer this Example:

Default Constructor:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Constructor/DefaultConstructor.cpp

Parameterised Constructor:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Constructor/parameterisedConstructor.cpp

Copy Constructor:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Constructor/copyConstructoor.cpp

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Dynamic Constructors

- The constructor can also be used to allocate memory while creating objects. This will enable the system to allocate the right amount for each object when the objects are not of the same size, thus resulting in the saving of memory.
- Allocation of memory to objects at the time of their construction is known as dynamic constructor of objects. The memory is allocated with the help of the new operator.

Refer This Example:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Constructor/DynamicConstr uctor.cpp

What is a Destructor?

- It is a member function which deletes an object.

- A destructor function is called automatically when the object goes out of scope:

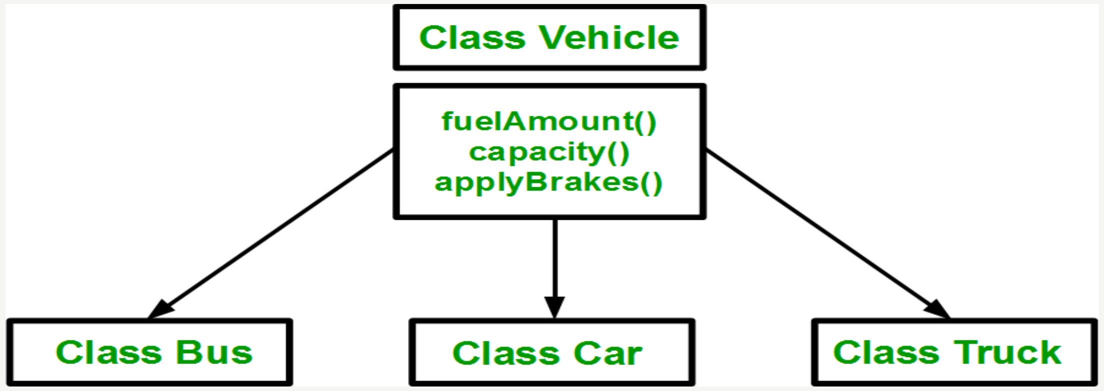
1. the function ends
1. the program ends
2. a block containing temporary variables ends
3. a delete operator is called

A destructor has:

1. the same name as the class but is preceded by a tilde (~)
2. no arguments and return no values

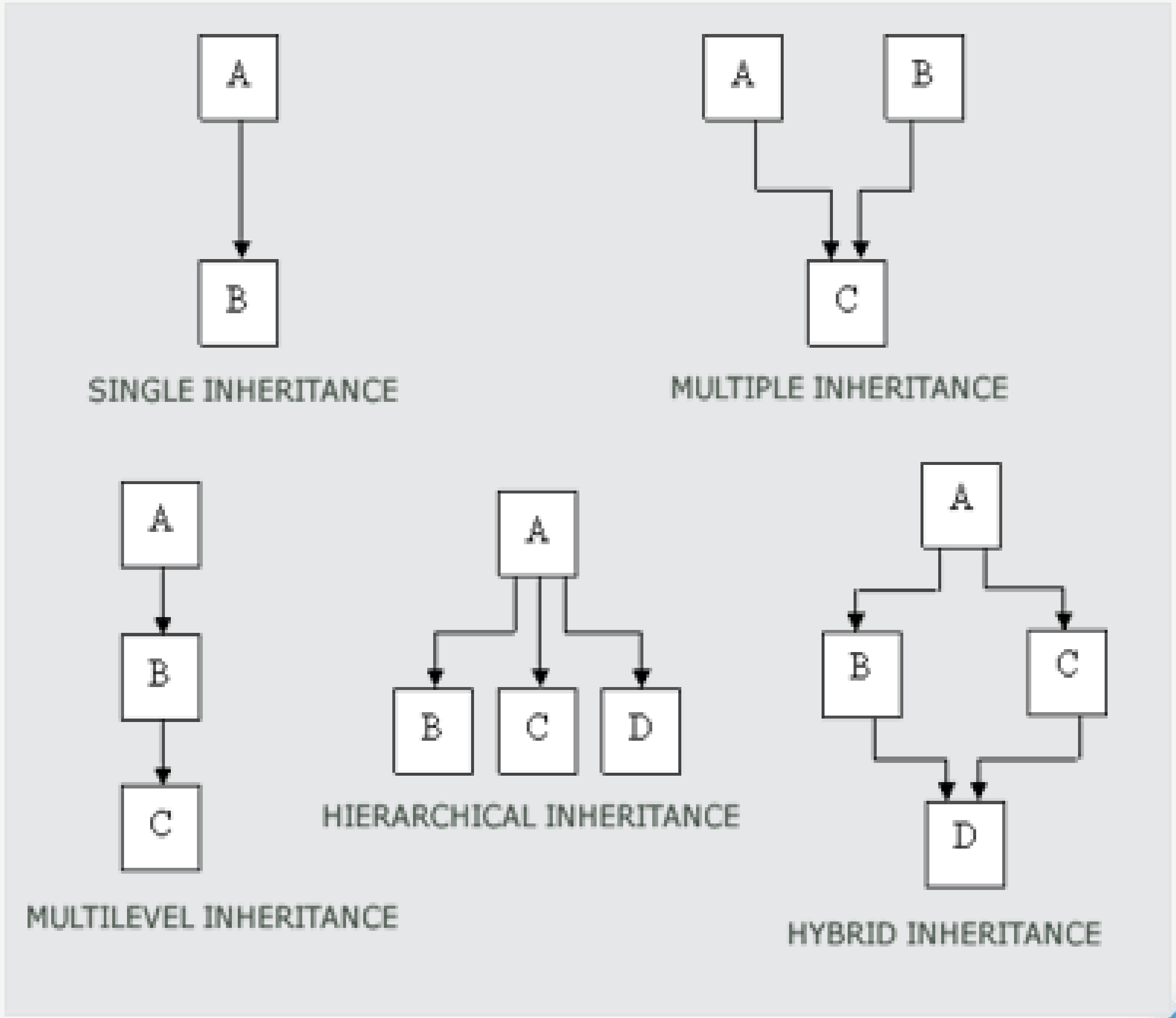
Inheritance

- Inheritance is a mechanism of reusing and extending existing classes without modifying them, thus producing hierarchical relationships between them.
- When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base** class, and the new class is referred to as the **derived** class.



Types of Inheritance

- ❑ Single Inheritance
- ❑ Multiple Inheritance
- ❑ Hierarchical Inheritance
- ❑ Multilevel Inheritance
- ❑ Hybrid Inheritance



Refer This Examples:

Single Inheritance:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Inheritance/single.cpp

Multiple Inheritance:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Inheritance/multiple.cpp

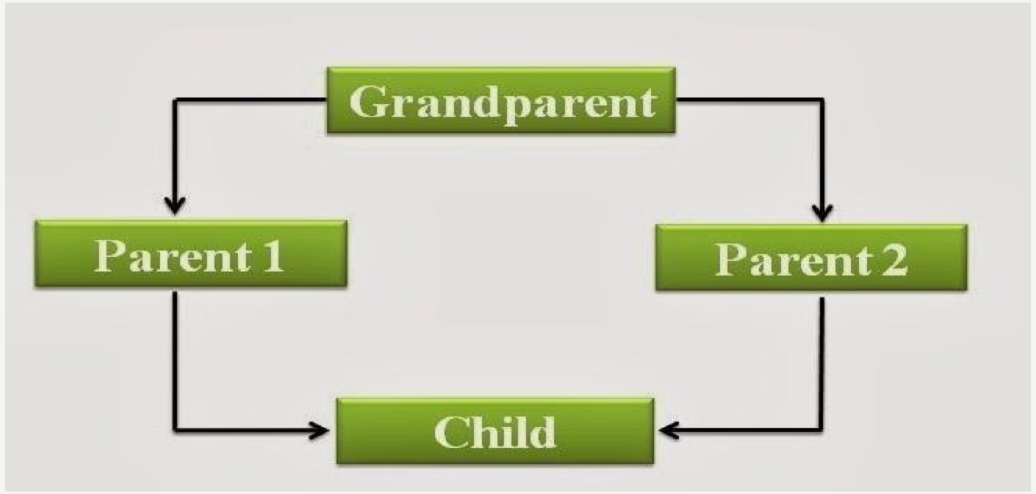
Multilevel Inheritance:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Inheritance/multilevel.cpp

Hierarchical Inheritance
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Inheritance/hierarchical.cpp

Hybrid Inheritance
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Inheritance/hybrid.cpp

Virtual Base Classes

- Virtual base class is used in situation where a derived have multiple copies of base class.
- When two or more objects are derived from a common base class, we can prevent multiple copies of the base class being present in an object derived from those objects by declaring the base class as virtual when it is being inherited. Such a base class is known as virtual base class.



- Refer This Example:

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/class/virtualClass.cpp

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Constructor in Derived Class

- Base class constructors are automatically called for you if they have no argument.
- If you want to call a superclass constructor with an argument, you must use the subclass's constructor initialization list.
- Unlike Java, **C++ supports multiple inheritance** (for better or worse), so the base class must be referred to by name, rather than "super()".

Refer This Example:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Constructor/constructorInheriitance.cpp

Polymorphism and Overloading

- Poly refers many.
- "single interface having multiple implementations."
- That is making a function or operator to act in different forms depending on the place they are present is called Polymorphism. Overloading is a kind of polymorphism.

- 2 Types of polymorphism:

1. **Static** : compile time
2. **Dynamic** : run time

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Polymorphism and Overloading

2 Types:

- **Function overloading** which is the process of using the same name for two or more functions.

Refer This Example: https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/FunctionOverloading/examp l e1.cpp

- **Operator overloading** which is the process of using the same operator for two or more operands.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Polymorphism and Overloading

Unary Operator Overloading:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/OperatorOverloadiing/Unary.cpp

Binary Operator Overloading:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/OperatorOverloadiing/binary.c

Polymorphism and Overloading

This refers to the entity which changes its form depending on circumstances at runtime.

Virtual Function:
A virtual function can be defined as the member function within a base class which you expect to redefine in derived classes.
For creating a virtual function, you have to precede your function's declaration within the base class with a **virtual** keyword.
Refer This Example:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Function/virtualfunction.cpp

What is Pure Virtual Function

Pure Virtual Function is a Virtual function with no body.

```
class classname //This denotes the base class of C++ virtual function
{
    public:
     virtual void virtualfunctioname() = 0        //This denotes the pure
                                                   virtual function in C++

};
```

Refer This Example:
https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Function/pureVirtualFunction.cpp

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Abstract classes

- An abstract class is a class that is designed to be specifically used as a base class.
- An abstract class contains at least one pure virtual function.
- You declare a pure virtual function by using a pure specifier (= 0) in the declaration of a virtual member function in the class declaration.
- You cannot use an abstract class as a parameter type, a function return type, or the type of an explicit conversion, nor can you declare an object of an abstract class.

Refer This Example:

https://github.com/TopsCode/Software-Engineering/blob/master/c%2B%2B/OOPS/Abstract%20Class/abstract.cpp

Static class

- We can define class members static using **static** keyword. When we declare a member of a class as static it means no matter how many objects of the class are created, there is only **one copy of the static member**.
- A static member is **shared by all objects of the class**. All static data is initialized to zero when the first object is created, if no other initialization is present. We can't put it in the class definition but it can be initialized outside the class as done in the following example by redeclaring the static variable, using the scope resolution operator :: to identify which class it belongs to.

Refer This Example:

https://github.com/TopsCode/Software-Engineering/blob/master/c++/OOPS/static/static.c

Static Function Members:

- By declaring a function member as static, you make it independent of any particular object of the class. A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator ::.
- A static member function can only access static data member, other static member functions and any other functions from outside the class.
- Static member functions have a class scope and they do not have access to the this pointer of the class. You could use a static member function to determine whether some objects of the class have been created or not.

Refer This Example:

https://github.com/TopsCode/Software-Engineering/blob/master/c++/OOPS/static/staticFunction.c

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

[Module 4]
HTML And CSS

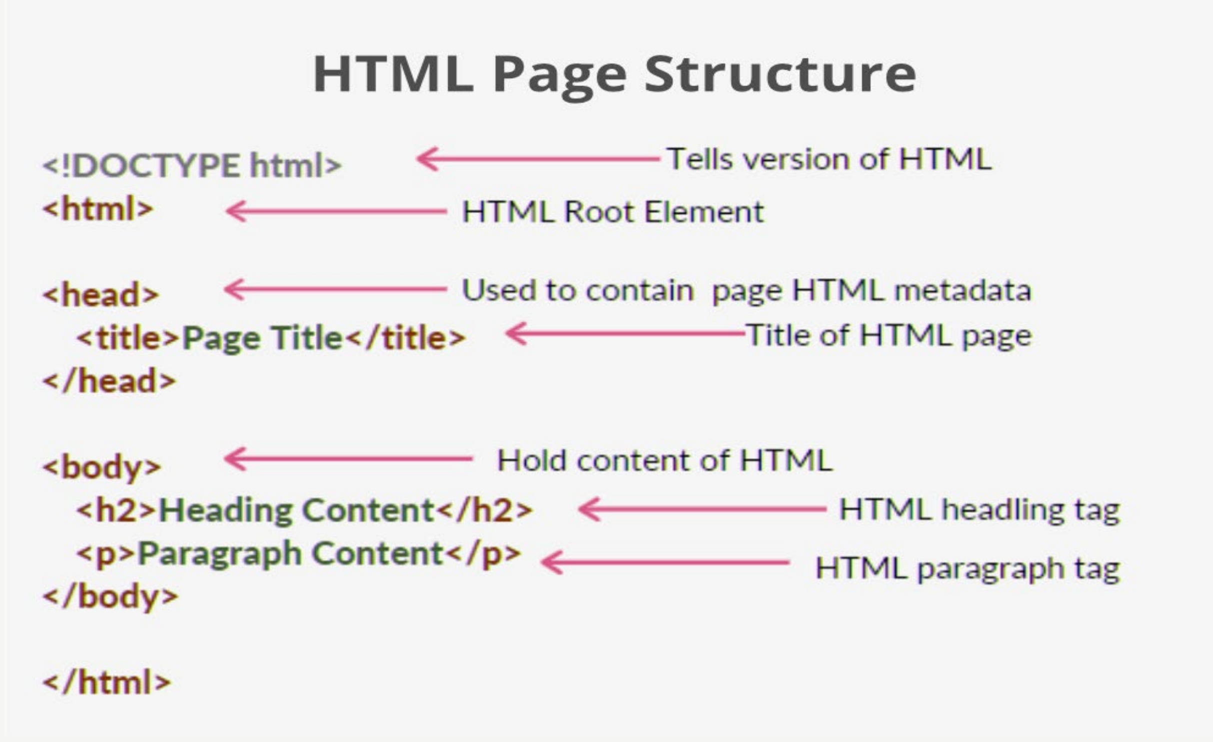
HTML

- HTML (Hypertext Markup Language) is the language used to create web page documents.
- The updated version, XHTML (extensible HTML) is essentially the same language with stricter syntax rules.
- (X)HTML is not a programming language; it is a markup language, which means it is a system for identifying and describing the various components of a document such as headings, paragraphs, and lists.
- You don’t need programming skills—only patience and common sense— to write (X)HTML.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

HTML Structure

- HTML : HTML stands for Hypertext Markup Language. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes


HTML Page Structure

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h2>Heading Content</h2>
  <p>Paragraph Content</p>
</body>

</html>
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

HTML

HTML Elements:

- HTML documents are text files made up of HTML elements. HTML elements are defined using HTML tags.

HTML Tags:

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and > The surrounding characters are called angle brackets HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag The text between the start and end tags is the element content HTML tags are not case sensitive, <b> means the same as <B>
- The browser interprets the HTML tags in the source code and displays your content according to those tags.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Basic Elements of HTML

- HTML Basic Tags:

Tag	Description
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Defines information about the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to<h6>	Defines HTML headings
<p>	Defines a paragraph
<br>	Inserts a single line break
<hr>	Defines a thematic change in the content
<!--...-->	Defines a comment

HTML

Heading Tag Example:
https://github.com/TopsCode/Software-Engineering/blob/master/HTML/heading.html

Paragraph Tag Example:
https://github.com/TopsCode/Software-Engineering/blob/master/HTML/paragraph.html

Line Break Example:
https://github.com/TopsCode/Software-Engineering/blob/master/HTML/LineBreak.html

Horizontal Line Example:

https://github.com/TopsCode/Software-Engineering/blob/master/HTML/horizontTOaPSlTLECiHnNOeLOG.IEhS PtVTm. LTDl.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

HTML

- HTML : HTML stands for Hypertext Markup Language. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes
- **HTML Link Tags**

<a href="http://www.TOPS-int.com">Go to TOPS-int.com</a>

- **HTML Image**

https://github.com/TopsCode/Software-Engineering/blob/master/HTML/image1.html

- **HTML List**

https://github.com/TopsCode/Software-Engineering/blob/master/HTML/

- **HTML Table**

https://github.com/TopsCode/Software-Engineering/blob/master/HTML/Table/colspan.html

- **HTML Form**

https://github.com/TopsCode/Software-Engineering/blob/master/HTML/Form/example1.html

HTML

- **Formatting Text:**

1. Bold Text: <b> and <strong>
2. Italic Text: <i > and <em>
3. Marked Formatting: <mark>
4. Underlined Text: <u> and <ins>
5. Strike Text: <strike> and <del>
6. Monospaced Font: <tt>
7. Superscript Text: <sup>
8. Subscript Text: <sub>
9. Larger Text: <big>
10. Smaller Text: <small>

Basic Elements of HTML

- **HTML Table Elements:**

<table>
<tr>
<th>
<td>
<caption>
<tbody>
<thead>
<tfooter>

Attributes:

rowspan
colspan

Basic Elements of HTML

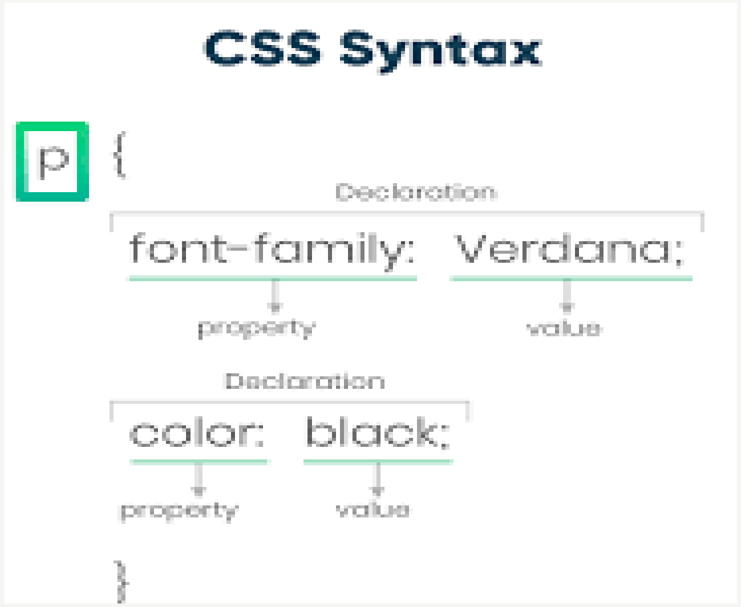
- **HTML FORM Elements:**

<form>
<input>
<textarea>
<label>
<fieldset>
<legend>

CSS

- CSS: CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colors, layout, and fonts, thus making our web pages presentable to the users.
- CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let’s try to break the acronym:

i. Cascading: Falling of Styles
ii. Style: Adding designs/Styling our HTML tags
iii. Sheets: Writing our style in different documents



CSS

- There are three ways to insert CSS in HTML documents.

1. Inline CSS

```
        exa: <h2 style="color:red;margin-left:40px;">Inline CSS is applied on this
heading.</h2>
```

2. Internal CSS

```
        exa: <style>
                body {
                background-color: linen;
                }
                h1 {
                color: red; margin-left: 80px;
                }
        </style>
```

CSS

3. External CSS

```
exa:<head>
        <link rel="stylesheet" type="text/css" href="mystyle.css">
    </head>
```

- Create Basic layout structure using div tag

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

CSS

Types of Selectors :

1. Universal Selector
2. HTML selector
3. Id Selector
4. Class Selector
5. Descendant Selector

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

CSS

Examples :

https://github.com/TopsCode/Software-Engineering/blob/master/CSS/htmlSelector.html

https://github.com/TopsCode/Software-Engineering/blob/master/CSS/universalSelector.html

https://github.com/TopsCode/Software-Engineering/blob/master/CSS/classSelector.html

https://github.com/TopsCode/Software-Engineering/blob/master/CSS/dependantClassSelector.html

https://github.com/TopsCode/Software- Engineering/blob/master/CSS/idSelector.html

https://github.com/TopsCode/Software-Engineering/blob/master/CSS/descendantSelector.css

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

CSS

Font Formatting with CSS

1. font-family

- The font family of a text is set with the font-family property.
- The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.
- Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Example: **https://github.com/TopsCode/Software-Engineering/blob/master/CSS/Font/font-family.html**

CSS

2.Serif

Examples: Times, Times New Roman, Georgia
Serif typefaces have decorative serifs, or slab-like appendages, on the ends of certain letter strokes.

3. sans-serif

Examples: Arial, Arial Black, Verdana, Trebuchet MS, Helvetica, Geneva
Sans-serif typefaces have straight letter strokes that do not end in serifs.
They are generally considered easier to read on computer .

4. Monospace

Examples: Courier, Courier New, and Andale Mono
In monospace (also called constant width) typefaces, all characterstake up the same amount of

4. Cursive

Examples: Apple Chancery, Zapf-Chancery, and Comic Sans
Cursive fonts emulate a script or handwritten appearance.
These are rarely specified for professional web pages.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

CSS

2.Fantasy

Examples: Impact, Western, or other decorative font
Fantasy fonts are purely decorative and would be appropriate for headlines and other display type.
Fantasy fonts are rarely used for web text due to cross-platform availability and legibility.

Specifying font size:
Use the aptly-named font-size property to specify the size of the text.

font-size
Values: length unit, percentage, xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | inherit

Default: medium Applies to: all elements Inherits: yes
https://github.com/TopsCode/Software-Engineering/blob/master/CSS/Font/font-size.html

CSS Box Model

- **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent

- **Border** - A border that goes around the padding and content. The border is affected by the background color of the box

- **Padding -** Clears an area around the content. The padding is affected by the background color of the box

- **Content** - The content of the box, where text and images appear

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

https://github.com/TopsCode/Software- Engineering/blob/master/CSS/boxModel.html

https://github.com/TopsCode/Software- Engineering/blob/master/CSS/boxmodel2.html

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

[Module 5] Database

DBMS

- DBMS stands for Data Base Management System.
- Data + Management System
- Database is a collection of inter-related data and Management System is a set of programs to store and retrieve those data.
- DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.
- For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

DBMS

Here is a list of some popular database.

- MySQL
- Microsoft Access
- Oracle
- PostgreSQL
- dBASE
- FoxPro
- SQLite
- IBM DB2
- LibreOffice Base

Need of DBMS

- Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data
  Storage:
- According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage.
- Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

RDBMS

- Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data
  Storage:
- According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage.
- Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

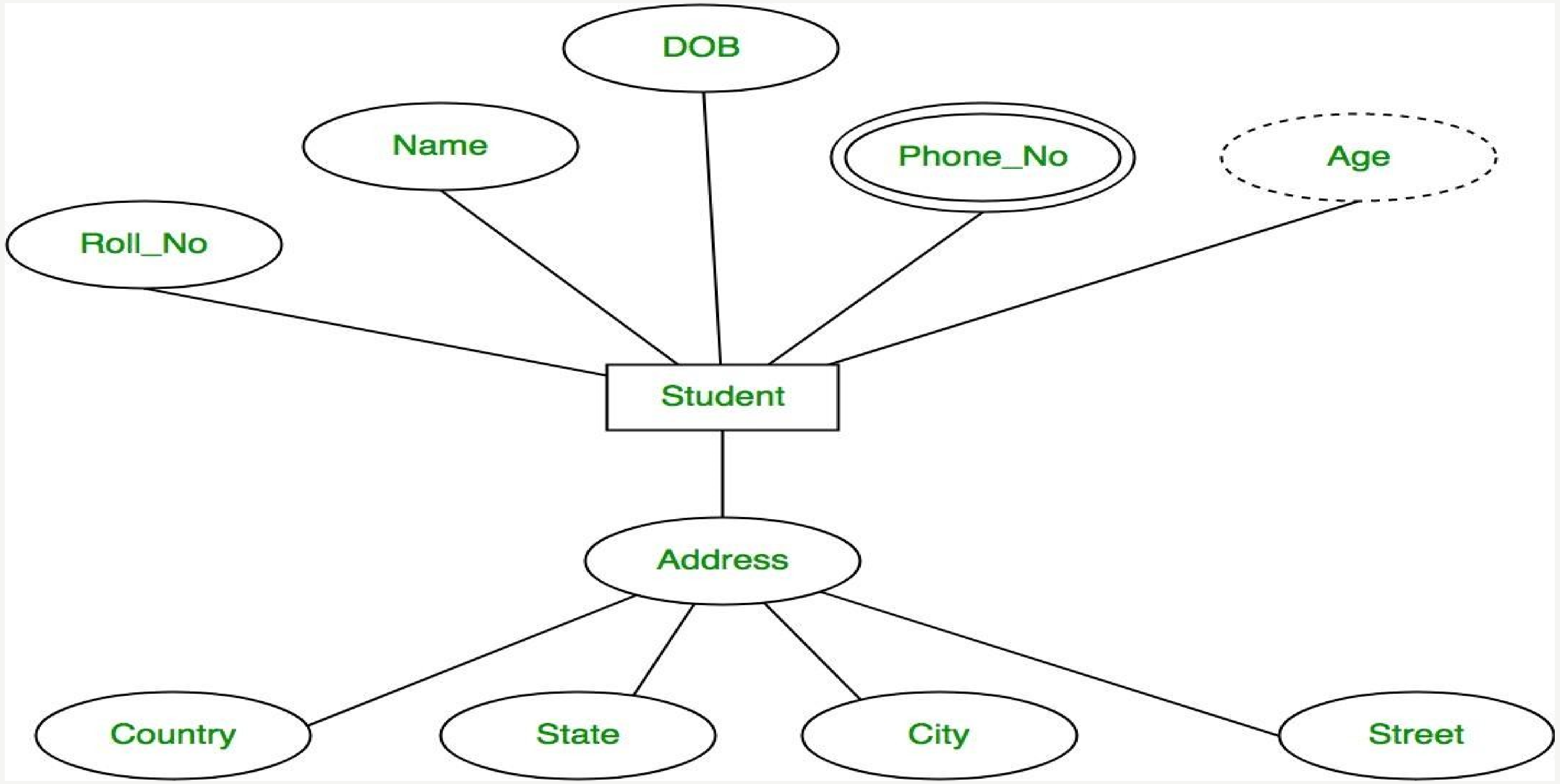
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

E-R Model

- The ER or (Entity Relational Model) is a high-level conceptual data model diagram.
- Entity-Relation model is based on the notion of real-world entities and the relationship between them.
- ER modeling helps you to analyze data requirements systematically to produce a well-designed database.
- So, it is considered a best practice to complete ER modeling before implementing your database.
- Entity relationship diagram displays the relationships of entity set stored in a database.
- In other words, we can say that ER diagrams help you to explain the logical structure of databases.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

E-R Model



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Algebra

- Relational Algebra is procedural query language, which takes Relation as input and generate relation as output. Relational algebra mainly provides theoretical foundation for relational databases and SQL.

Unary Relational Operations

- SELECT (symbol: σ)
- PROJECT (symbol: π) RENAME (symbol:)

Relational Algebra Operations From Set Theory

- UNION (υ) INTERSECTION ( ), DIFFERENCE (-) CARTESIAN PRODUCT (x)

Binary Relational Operations

- JOIN DIVISION

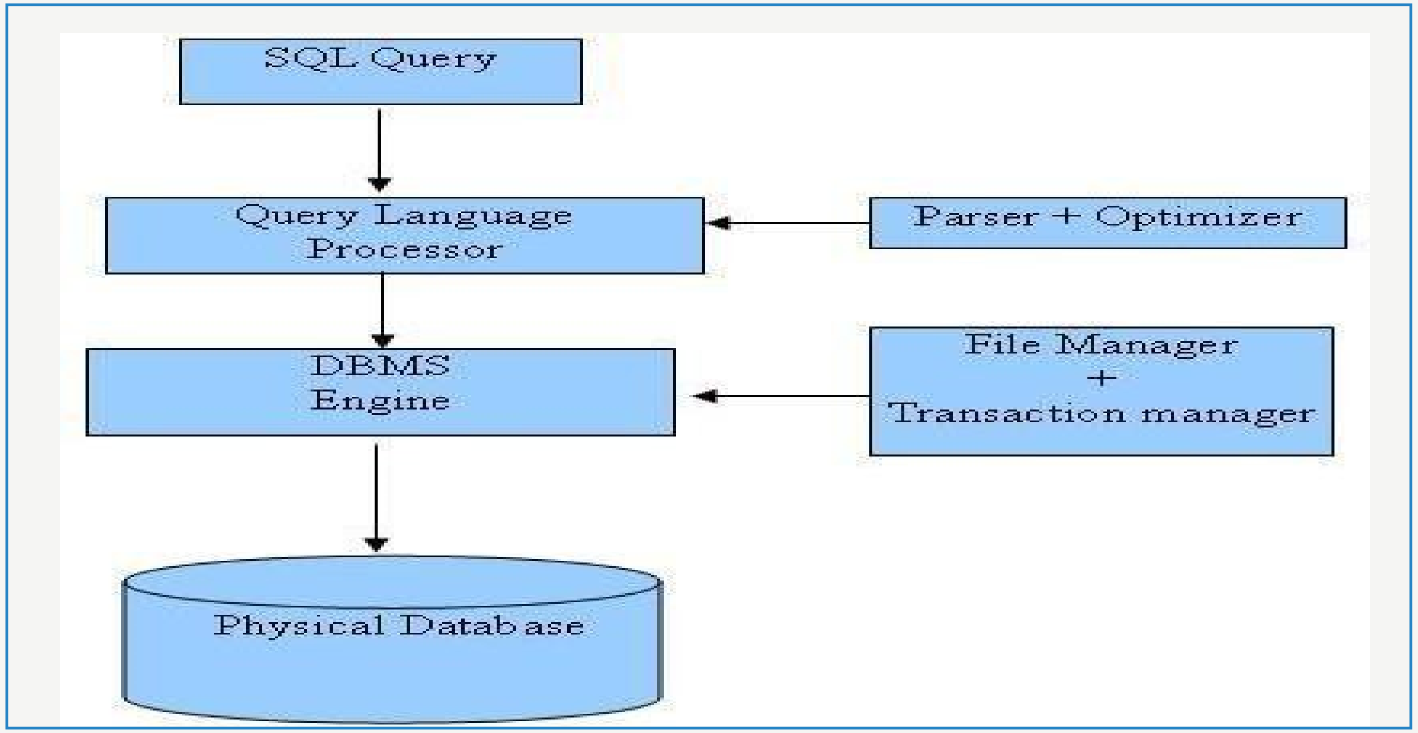
TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

What is SQL?

- SQL is Structured Query Language, which is a computer language for storing,
- manipulating and retrieving data stored in relational database.
- SQL is a language of database, it includes database creation, deletion, fetching rows and modifying rows etc.
- SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server use SQL as standard database language.
- Also, they are using different dialects, such as:
- MS SQL Server using T-SQL, ANSI SQL
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc

Objectives

- **"**The large majority of today's business applications revolve around relational databases and the SQL programming language (Structured Query Language). Few businesses could function without these technologies…"



What is SQL?

- Allows users to access data in relational database management systems. Allows users to describe the data.
- Allows users to define the data in database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database. Allows users to set permissions on tables, procedures, and views
- SQL stands for Structured Query Language
- SQL allows you to access a database
- SQL is an ANSI standard computer language
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert new records in a database

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

What is SQL?

- SQL can delete records from a database
- SQL can update records in a database
- SQL is easy to learn
- action queries insert, update & delete data
- select queries retrieve data from DB

Constraints(Keys)

Primary Key:

- A primary key is a column of table which uniquely identifies each
- tuple (row) in that table.
- Primary key enforces integrity constraints to the table.
- Only one primary key is allowed to use in a table.
- The primary key does not accept the any duplicate and NULL values.
- The primary key value in a table changes very rarely so it is chosen with care
- where the changes can occur in a seldom manner.
- A primary key of one table can be referenced by foreign key of another table.

Constraints(Keys)

Unique Key:

- Unique key constraints also identifies an individual table uniquely in a relation or table.
- A table can have more than one unique key unlike primary key.
- Unique key constraints can accept only one NULL value for column.
- Unique constraints are also referenced by the foreign key of another table.

Constraints(Keys)

Foreign Key:

- When, "one" table's primary key field is added to a related "many" table in order to create the common field which relates the two tables, it is called a foreign key in the "many" table.
- In the example given below, salary of an employee is stored in salary table.
- Relation is established via is stored in "Employee" table. To identify the salary of "Jforeign key column “Employee_ID_Ref” which refers “Employee_ID” field in Employee table.
- of "Jhon" is stored in "Salary" table. But his employee info
- For example, salary hon", his "employee id" is stored with each salary record.

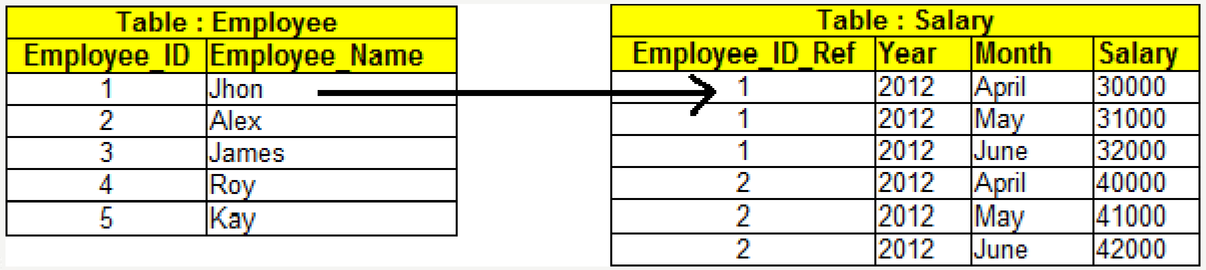


Table : Employee	
Employee_ID	Employee_Name
1	Jhon
2	Alex
3	James
4	Roy
5	Kay

Table : Salary			
Employee_ID_Ref	Year	Month	Salary
1	2012	April	30000
1	2012	May	31000
1	2012	June	32000
2	2012	April	40000
2	2012	May	41000
2	2012	June	42000

Database Normalization

- Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations.
- Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.
- Most Commonly used normal forms:

First Normal Form:

- First normal form(1NF) Second normal form(2NF) Third normal form(3NF) Boyce & Code normal form (BCNF)
- If a relation contain composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute.
- A relation is in first normal form if every attribute in that relation is singled valued attribute.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Database Normalization

Second Normal Form:

- To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency.
- relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.
- Partial Dependency – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

Database Normalization

Third Normal Form:

- A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.
- A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y

  X is a super key.

  Y is a prime attribute (each element of Y is part of some candidate key).
- Transitive dependency – If A->B and B->C are two FDs then A->C is called transitive dependency.

SQL Statement Types

- DDL – Data Definition Language
- DML – Data Manipulation Language
- DCL – Data Control Language
- DQL – Data Query Language
- **SQL Join Types**
- INNER JOIN: returns rows when there is a match in both tables.
- LEFT JOIN: returns all rows from the left table, even if there are no
- matches in the right table.
- RIGHT JOIN: returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN: returns rows when there is a match in one of the tables.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

DDL - Data Definition Language

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

DQL – Data Query Language

Command	Description
SELECT	Retrieves certain records from one or more tables

- **DML – Data Manipulation Language**

Command	Description
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

DCL – Data Control Language

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

SQL CREATE DATABASE STATEMENT
CREATE DATABASE database_name;

SQL DROP DATABASE Statement:
DROP DATABASE database_name;

SQL USE STATEMENT
USE DATABASE database_name;

SQL CREATE TABLE STATEMENT
CREATE TABLE table_name(column1 datatype, column2 datatype, column3 datatype, ... , columnN datatype, PRIMARY KEY(one or more columns));

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

SQL DROP TABLE STATEMENT

```
DROP TABLE table_name;
```

SQL TRUNCATE TABLE STATEMENT

```
TRUNCATE TABLE table_name;
```

SQL ALTER TABLE STATEMENT (RENAME)

```
ALTER TABLE table_name RENAME TO new_table_name;
```

SQL INSERT INTO STATEMENT

```
INSERT INTO table_name( column1, column2....columnN) VALUES ( value1, value2.
        valueN);
```

SQL Statement Types

SQL UPDATE STATEMENT
UPDATE table_name SET column1 = value1, column2 = value2. columnN=valueN
[WHERE CONDITION];

SQL DELETE STATEMENT
DELETE FROM table_name WHERE {CONDITION}

SQL SELECT STATEMENT
SELECT column1, column2....columnN FROM table_name;

SQL DISTINCT CLAUSE
SELECT DISTINCT column1, column2....columnN FROM table_name;

SQL Statement Types

SQL WHERE CLAUSE

SELECT column1, column2....columnN FROM table_name WHERE CONDITION;

SQL AND/OR CLAUSE

SELECT column1, column2....columnN FROM table_name WHERE CONDITION-1 {AND|OR} CONDITION-2;

SQL IN CLAUSE

SELECT column1, column2. column table_name WHERE column_name IN FROM (val-1, val-2, val-N);

SQL BETWEEN CLAUSE

SELECT column1, column2. column table_name WHERE column_name FROM BETWEEN val-1 AND val-2;

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

SQL LIKE CLAUSE

SELECT column1, column2. column table_name WHERE column_name LIKE {
FROM PATTERN };

SQL ORDER BY CLAUSE

SELECT column1, column2. columnN table_name WHERE CONDITION ORDER
FROM BY column_name {ASC|DESC};

SQL GROUP BY CLAUSE

SELECT SUM(column_name) table_name WHERE CONDITION GROUP BY
FROM column_name;

SQL COUNT CLAUSE

SELECT COUNT(column_name)FROM table_name WHERE CONDITION;

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

SQL HAVING CLAUSE
SELECT SUM(column_name) FROM table_name WHERE CONDITION GROUP BY column_name HAVING (arithematicfunction condition)

SQL CREATE INDEX Statement :
CREATE UNIQUE INDEX index_name ON table_name(column1, column2,...columnN);

SQL DROP INDEX STATEMENT
ALTER TABLE table_name DROP INDEX index_name;

SQL DESC Statement :
DESC table_name;

SQL COMMIT STATEMENT
COMMIT

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

SQL ROLLBACK STATEMENT

```
ROLLBACK;
```

JOIN

A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them.

Different types of Joins are:

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. FULL JOIN

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

- **1. Inner Join**
  The most frequently used and important of the joins is the INNER JOIN. They are also referred to as an EQUIJOIN.
- The INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. T
- he query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

Syntax :

SELECT table1.column1, table2.column2...FROM table1INNER JOIN table2ON table1.common_filed = table2.common_field

SQL Statement Types

- **Left Join**

- The SQL LEFT JOIN returns all rows from the left table, even if there are no matches in the right table. This means that if the ON clause matches 0 (zero) records in right table, the join will still return a row in the result, but with NULL in each column from right table.
- This means that a left join returns all the values from the left table, plus matched values from the right table or NULL in case of no matching join predicate.
- **Syntax:**
- SELECT table1.column1, table2.column2...FROM table1LEFT JOIN table2ON table1.common_filed = table2.common_field;

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

- **Right Join**
- The SQL RIGHT JOIN returns all rows from the right table, even if there are no matches in the left table. This means that if the ON clause matches 0 (zero) records in left table, the join will still return a row in the result, but with NULL in each column from left table.
- This means that a right join returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.

- **Syntax:**
- SELECT table1.column1, table2.column2...FROM table1RIGHT JOIN table2ON table1.common_filed = table2.common_field

SQL Statement Types

- **Full Join**
- The SQL FULL JOIN combines the results of both left and right outer joins.
- The joined table will contain all records from both tables, and fill in NULLs for missing matches on either side.

- **Syntax:**
- SELECT table1.column1, table2.column2...FROM table1FULL JOIN table2ON table1.common_filed = table2.common_field;

SQL Statement Types

Function

- SQL has many built-in functions for performing calculations on
- data. They are divided into 2 categories:

1. Aggregate Function
2. Scalar Function

1. **Aggregate Function**

- These functions are used to do operations from the values of the column and a single value is returned.
- AVG() - Returns the average value COUNT() - Returns the
- number of rows FIRST() - Returns the first value
- LAST() - Returns the last value MAX() - Returns the largest value MIN() - Returns the smallest value SUM() - Returns the sum

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

1. **AVG():**

Syntax: SELECT AVG(column_name) FROM table_name;
Example:
SELECT AVG(AGE) AS AvgAge FROM Students;

2. COUNT()

Syntax: SELECT COUNT(column_name) FROM table_name;
Example: SELECT COUNT(*) AS NumStudents FROM Stuents;

3. FIRST()

Syntax: SELECT FIRST(column_name) FROM table_name
Example:
SELECT FIRST(MARKS) AS MarksFirst FROM Students;

SQL Statement Types

4. LAST()
Syntax: SELECT LAST(column_name) FROM table_name;
Example: SELECT LAST(MARKS) AS MarksLast FROM Students;

5. MAX()
Syntax: SELECT MAX(column_name) FROM table_name;
Example :SELECT MAX(MARKS) AS MaxMarks FROM Students;

6. **MIN():** Similar to Max() we can use MIN() function.

7. SUM() :
Syntax: SELECT SUM(column_name) FROM table_name;
Example: SELECT SUM(MARKS) AS TotalMarks FROM Stuents**;**

SQL Statement Types

PROCEDURE

- A stored procedure is a prepared SQL code that you can save, so the code can be
- reused over and over again.
- So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.
- You can also pass parameters to a stored procedure, so that the stored procedure
- can act based on the parameter value(s) that is passed.
- To create procedure, use syntax:

```
        CREATE PROCEDURE procedure_name AS sql_statement GO;
```

- To execute created procedure, use syntax:

```
        EXEC procedure_name;
```

SQL Statement Types

Transaction Control

- **The following commands are used to control transactions**

1. COMMIT – to save the changes.
2. ROLLBACK – to roll back the changes.
3. SAVEPOINT – creates points within the groups of transactions in which to ROLLBACK.

- **Commit:**
- The COMMIT command is the transactional command used to save changes invoked by a transaction to the database.
- The COMMIT command saves all the transactions to the database since the last
- COMMIT or ROLLBACK command.

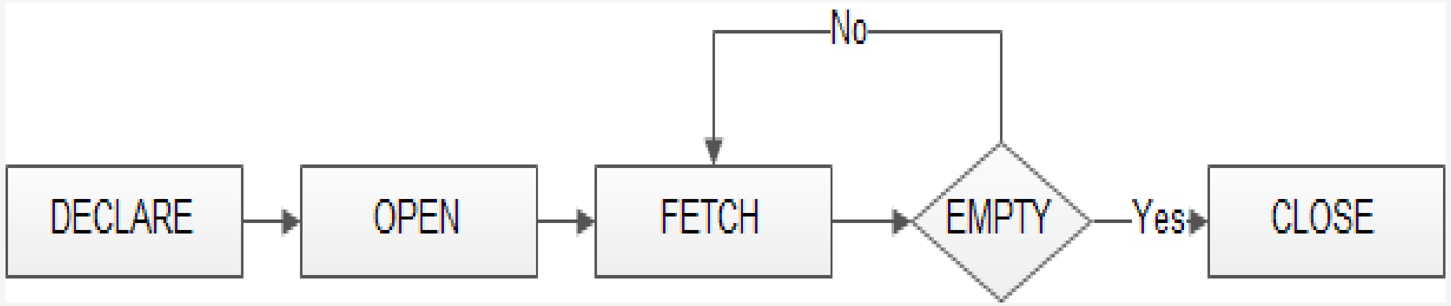
SQL Statement Types

- **Rollback:**
- The ROLLBACK command is the transactional command used to undo transactions that have not already been saved to the database.
- This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.
- The syntax for a ROLLBACK command is as follows – ROLLBACK;

- **Savepoint:**
- A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.
- The syntax for a SAVEPOINT command is as shown below.
- SAVEPOINT SAVEPOINT_NAME;
- This command serves only in the creation of a SAVEPOINT among all the transactional statements. The ROLLBACK command is used to undo a group of transactions.

SQL Statement Types

- **Cursor:**
- It is a temporary area for work in memory system while the execution of a statement is done.
- A Cursor in SQL is an arrangement of rows together with a pointer that recognizes a present row.
- It is a database object to recover information from a result set one row at once.
- It is helpful when we need to control the record of a table in a singleton technique, at the end of the day one row at any given moment. The arrangement of columns the cursor holds is known as the dynamic set.



SQL Statement Types

- **Cursor:**

Syntax:

```
DECLARE variables;
records;
 create a cursor;
 BEGIN
OPEN cursor;
FETCH cursor;
process the records;
CLOSE cursor;
 END;
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

SQL Statement Types

Cursor Example:
https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Cursor/example

Stored Procedure with one parameter:
https://github.com/TopsCode/Software-Engineering/tree/master/SQL/Cursor

Stored Procedure with multiple parameter:
https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Cursor/multipleParam

SQL Statement Types

Trigger

- A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs
- For example, a trigger can be invoked when a row is inserted into a specified table.

Syntax:

create trigger [trigger_name] [before | after]
{insert | update | delete} on [table_name] [for each row] [trigger_body]

Before Trigger:

https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Trigger/beforeTrigger

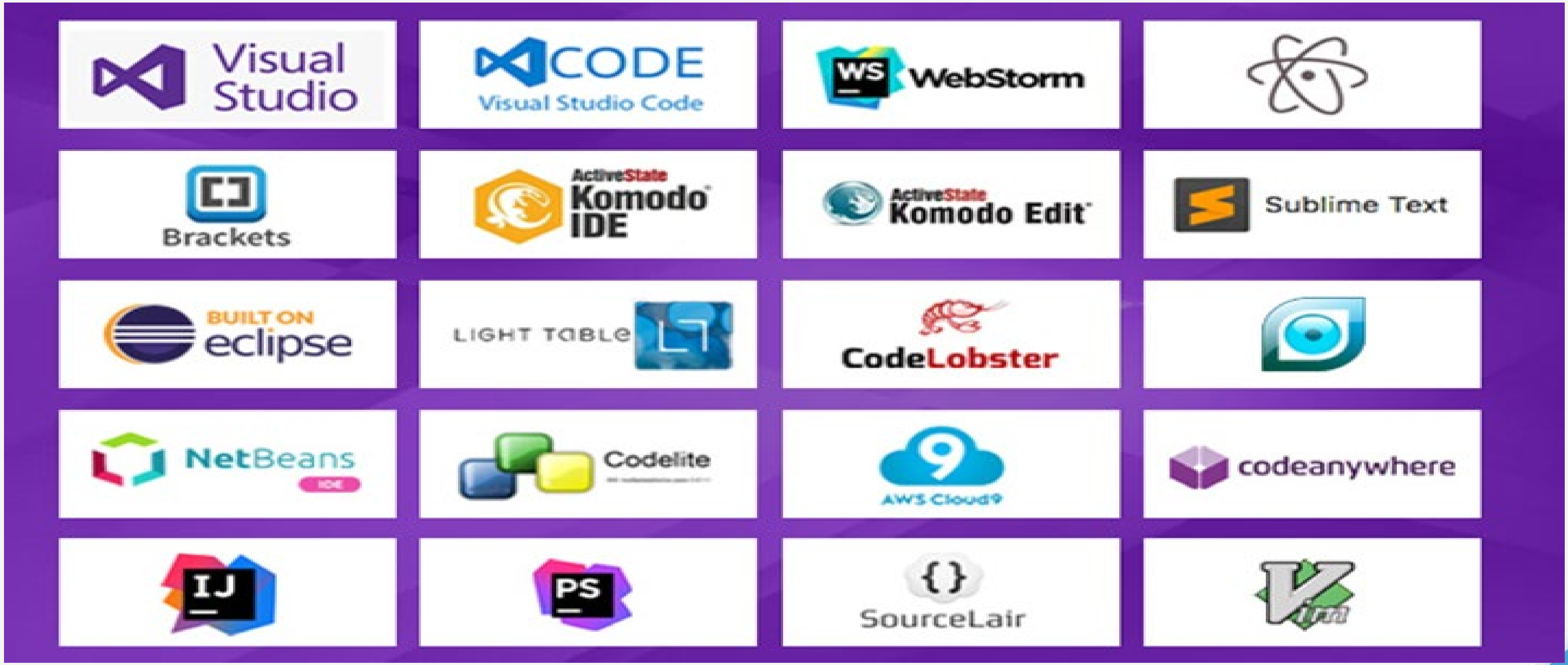
After Trigger:

https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Trigger/afterTrigger

11.Tools

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Tools can be used in Programing



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

THANK YOU