**Index**

**Delivery Life Cycle--------------------------------------------------------------------------------------------------2**

**Version Management System-----------------------------------------------------------------------------------2**

**GitHub-----------------------------------------------------------------------------------------------------------------2**

**Cloud-------------------------------------------------------------------------------------------------------------------2**

# 1.Delivery Life Cycle

Life cycle of project management(develop,test,live)

# 2.Version Management System

Baseline(production)

Logo-Color 2 phase

Unit Test-module test

Integration Test-Full Test(combine module)

# 3.Cloud-server rent

Buy/rent server instead of build own server(local-high cost)

Ex: Amazon AWS, Microsoft AZURE, Google Cloud

# 4.GitHub

Get solution for issues/question(ex:project)

web-based platform used for version control and collaboration on software development projects.

tracking changes made to code over time, and facilitating collaboration among developers

1 byte= 8 bit, bit is binary digit 0 and 1(on, off)

# 5.Data type-int, char, float(memory allocation)

Type value for the variable (char,int,float,Boolean)

# 6.ASCII-Numeric value for machine understand

American Std Code information interchange

Ex: Key A-65 (Numeric value for machine use)

Storing

Decimal-binary (97 Divide by 2) 97/2(D48-Q1)-48/2(D24-Q0)

Binary-Decimal 97 binary value (1100001) 1\*2^0=1,0\*2^1=0

# 7.Loops

IF-Construct

For, While

# 8.Arror-Collection of item stored in continues memory location

Array is collection of item stored in continues memory location

Array take 4 byte

**Syntax: A[]={1,2,3,4,5}**

A[30] base value 30 find value of location 2 (2\*4=8) base value 30+8=38

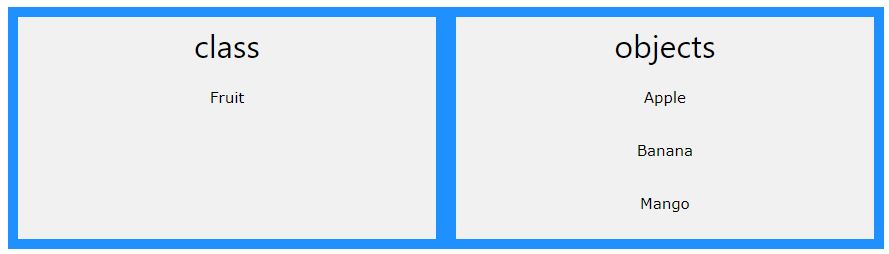
# 9.OOPS

Object oriented programming

Includes

# 9.1.object and class

Object- primary source to implement what is to happen in the code. Objects are seen by the viewer or user, performing tasks assigned



class is a template for objects. Objects is instance of class.

# Create class name as main

Public class main

{

Int x=10;

}

# **Create object with class:** Class Name follow object name use new keyword

Public class main

{ Int x=10;

Public static void main(string[] args)

{

Man myAge = new man();

System.out.println(myAge.x);

}

}

# **9.2.Abstraction**-hide certain details, shows only essential

abstract class Bike{

    void millege(){

        System.out.println("the millege");

    }

}

class TVS extends Bike{

    void millege(){

        System.out.println("The TVS Millege is 50");

    }

}

class Pulsar extends Bike{

    void millege(){

        System.out.println("The Pulsar Millege is 30");

    }

}

class Check {

    public static void main(String[] args) {

        TVS ob = new TVS();

        ob.millege();

        Pulsar ob1 = new Pulsar();

        ob1.millege();

    }

}

# 9.3.Inheritance- Inherit method from one class to another

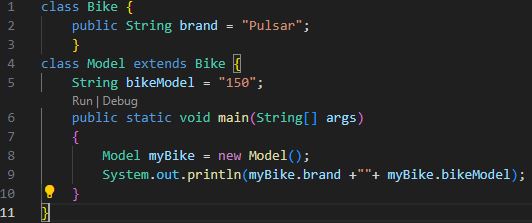
Sub class-Child

Super class-Parent

Types of Inheritance- Single, Multiple, Hierarchical

We cannot inherit one class to another in java. So we use keyword **“Extends”**

**Keys:** public, private, static, this

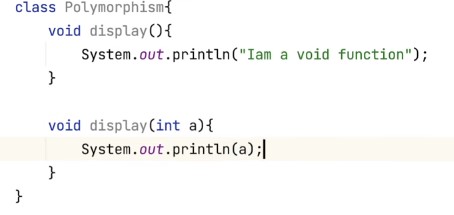


# 9.4.Encapsulation**-**Declare function. Use get and set to update the function in module.

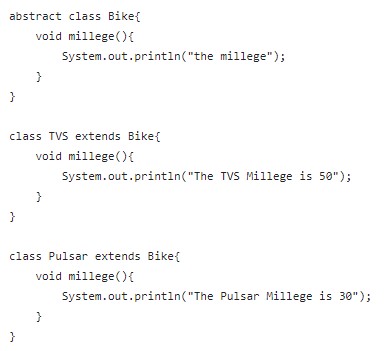
# 9.5.Polymorphism-the task performs a single action in different ways

message to be displayed in more than one form

Ex: A person at the same time can have different characteristics. Like a man at the same time is a father, and an employee. So the same person possesses different behaviours in different situations.



# Overriding- subclass has the same method as the parent class



# Overloading- two methods/ functions in a class with the same name and different arguments/ parameters

Ex: Class Bike

Function TVS()

Print tvs

FunctionTVS (int X)

Print tvs X

Main

Bike ob= new Bike();

Ob.TVS();

Ob.TVS(5);

# Access Specifier-used to set access level of class-Public, Private, Protected

Public-Any class use the code

Private-Particular class only use the code

Protected-Childrens/subclass under the parents.

# Package- group of similar types of classes, sub-packages(java.io)

Java.io predefine packages in java