Java

A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. (. Class )

The JVM is detailed by a specification that formally describes what is required in a JVM implementation



JRE – JAVA RUNTIME ENVIROMENT

JDK – JAVA DEVELOP KIT

JVM- JAVA VIRTUAL MACHINE

1. Program to print Hello World

class HelloWorld

{

public static void main(String args[])

{

System.out.println("HelloWorld");

}

}

**Answer**

HelloWorld

1. Program to print our class details (Line by line)(Subject name ,teacher name, students name)

class SeleniumBatch

{

public static void main(String args[])

{

System.out.println("Selenium Automation");

System.out.println("Rakhi Teacher");

System.out.println("Jaisree Student");

}

}

**Answer**

Selenium Automation

Rakhi Teacher

Jaisree Student

1. Program to print your full address. (Line by line)(name, house name, place, dist,state,pincode) include comments.

class FullAddress

{

public static void main(String args[])

{

System.out.println("Door no 92");

System.out.println("Arcot City");

System.out.println("Boston County");

System.out.println("MA State");

System.out.println("45001");

}

}

**Answer**

Door no 92

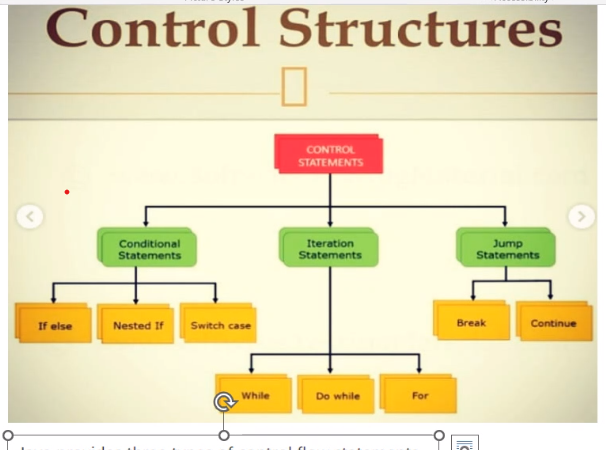
Arcot City

Boston County

MA State

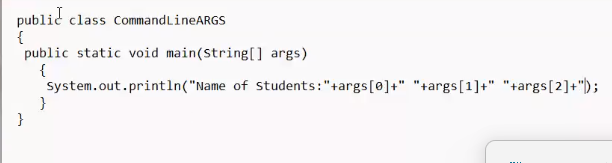
45001

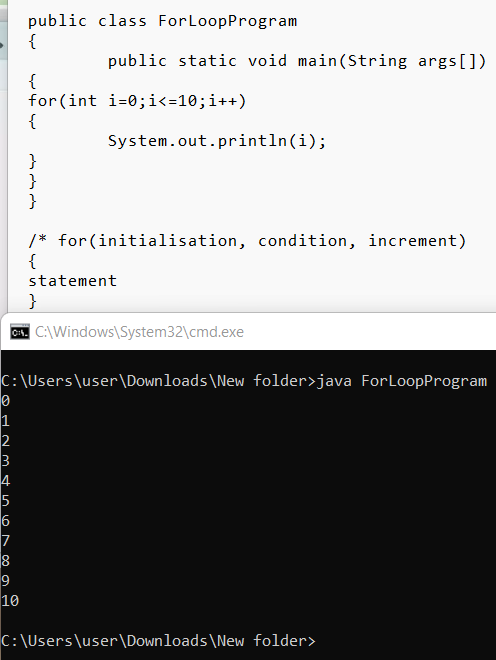
24/07/2023 Monday



Decision making, loop, Branching statement

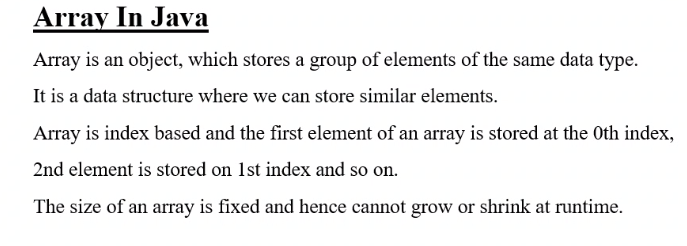
Documentation comment





27/07/2023

Array



It’s a fixed sized array

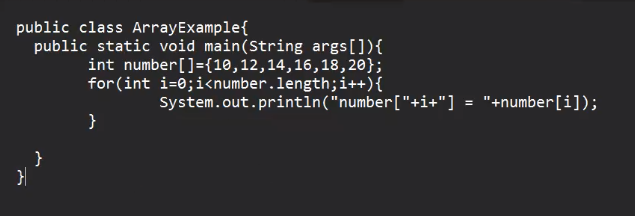
Int num[];//array declaration

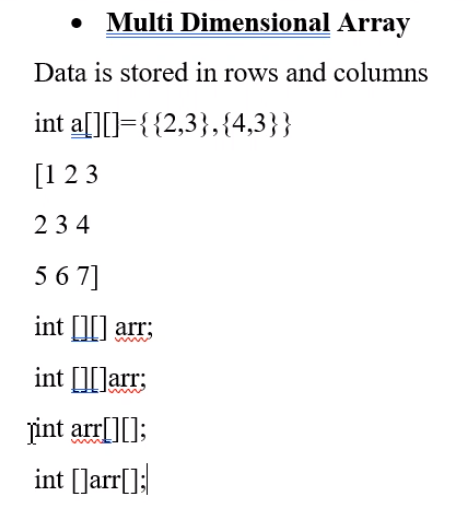
Num[]=new int[5];array instantiation

Num[0]=10;

Num[0]=20;

Public lass ArrayExample





Position

00 01 02

10 11 12

20 21 22

int arr[][]=new int[2][3];

arr[0][0]=10;

arr[0][1]=12;

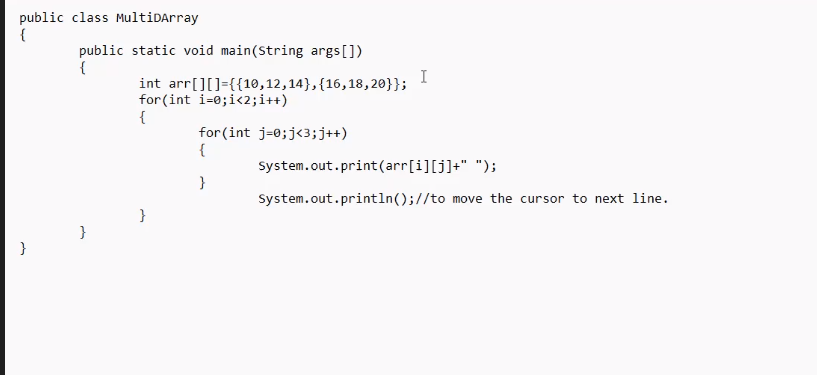
arr[0][2]=14;

arr[1][0]=16;

arr[1][1]=18;

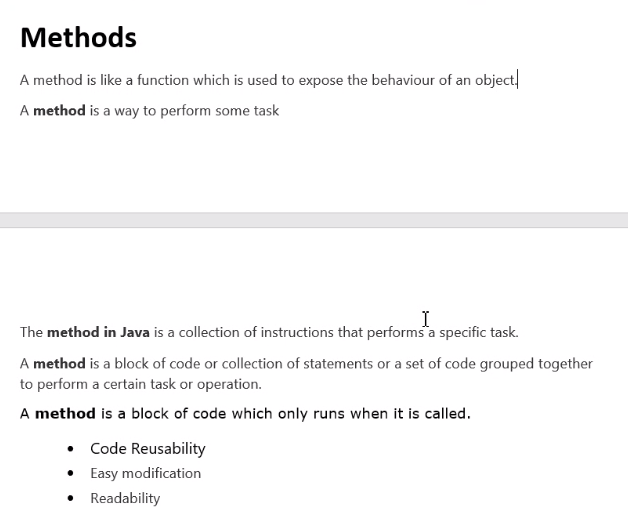
arr[1][2]=20;

int arr[][]={{10,12,14},{16,18,20}}



28/07/2023

Method

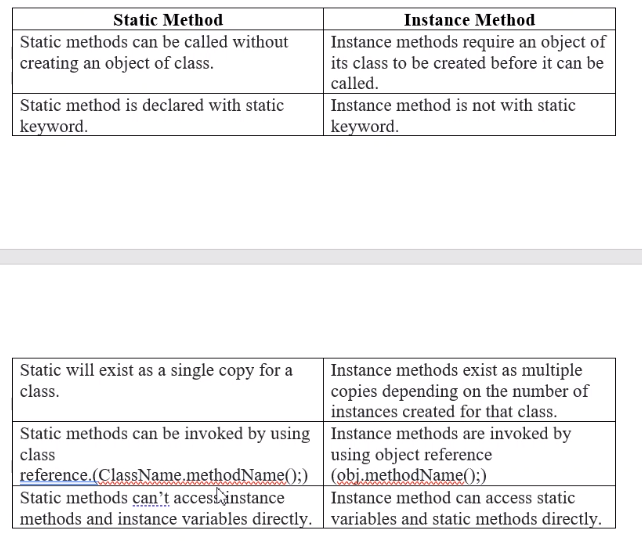


**Predefined Method**[length(),println(),equals(),max(),min()]

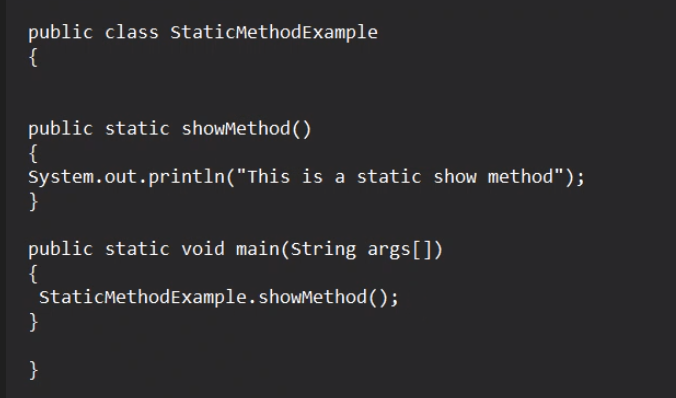
**User-defined Method**

Instance method/non static method

Static method



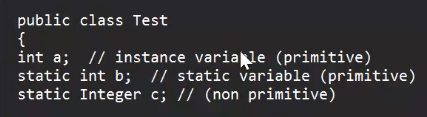
Static inside class outside method



Inside class inside method – local variable

Inside Class Outside method – instance

Static Integer b; static method



**Static method**

Class name.method name()

// Static method is being called using class name

// Static variable is accessed with the class name

Static method can’t access instance variable and instance method directly

**Instance method**

Obj.method name()

// Instance method is being called using object of the class

//Instance variables can be accessed inside a static method by creating an object of the class

Instance method can access Static variable and Static method directly// by creating obj to class

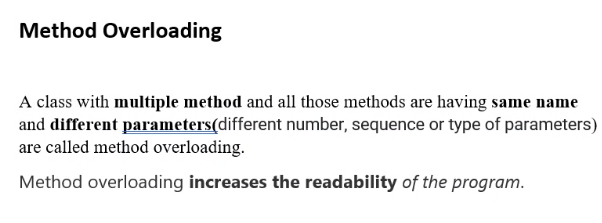
static int (static variable, primitive)

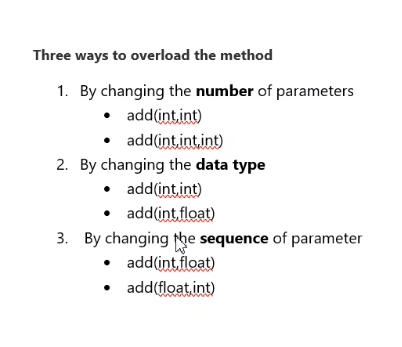
static Integer c=(Non Primitive)

int a = (instance variable,primitive)

August 01 2023

Method overloading





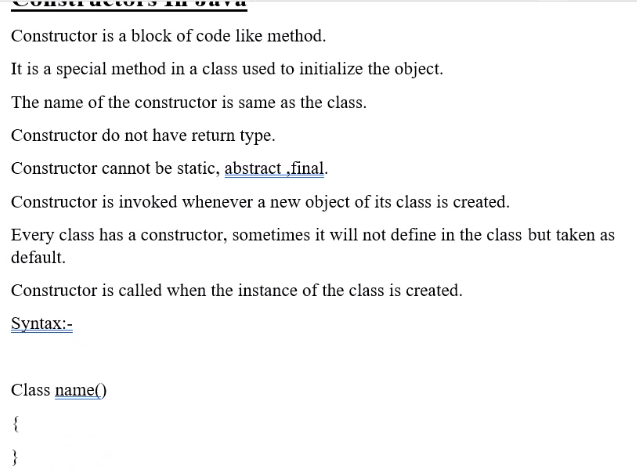
Real world entity is a properties and behaviors object.

Object is the instance of the class with example

Class is a blue print

Defining Object and properties (collection of objects)

Constructor



Example:

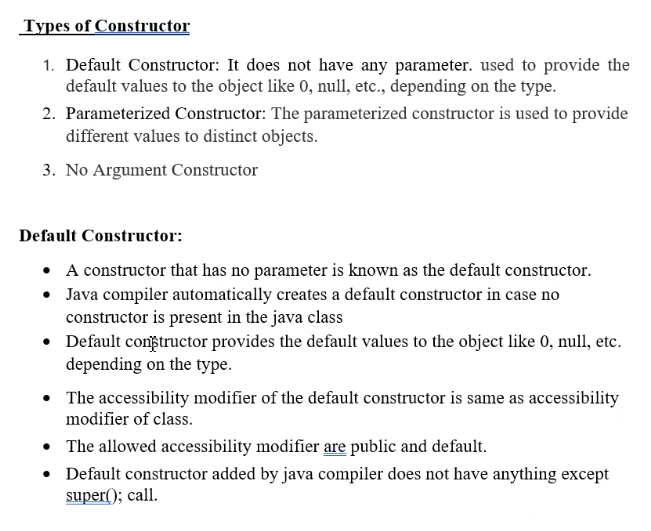
Public class student{

Student() // this is constructor

{

}

No argument constructor { statement}



In Java, objects are instances of classes. A class serves as a blueprint or a template for creating objects. It defines the structure and behavior of objects of that type. When you create an object, you are instantiating a class, which means you are creating a specific instance of that class with its own unique data.

|  |  |
| --- | --- |
| Class | object |
| In Java, a class is a blueprint or a template for creating objects. It defines the structure and behavior of objects that belong to that particular class. In simple terms, a class is a user-defined data type that encapsulates data (attributes) and methods (functions) related to a specific entity or concept.  The class serves as a blueprint that describes how objects of that type will be constructed and what actions they can perform. When you create an object from a class, you are instantiating the class, which means you are creating a specific instance of that class with its own unique data. | In Java, an object is a fundamental concept and a key building block of object-oriented programming (OOP). It is a self-contained unit that encapsulates data and behavior (methods) related to a particular entity or concept. In simple terms, an object can be thought of as a real-world entity that has attributes (data) and actions (methods) that can be performed on it.  In Java, objects are instances of classes. A class serves as a blueprint or a template for creating objects. It defines the structure and behavior of objects of that type. When you create an object, you are instantiating a class, which means you are creating a specific instance of that class with its own unique data. |

This key word to refer current variable on the class

Package-string

04/08/2023

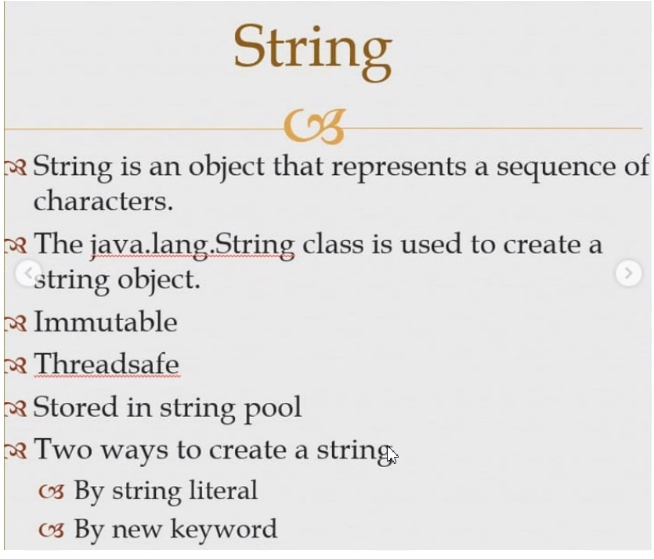
Package

Userdefined package – user can create

import package.MyProject

Predefined package/builtin – already defined package

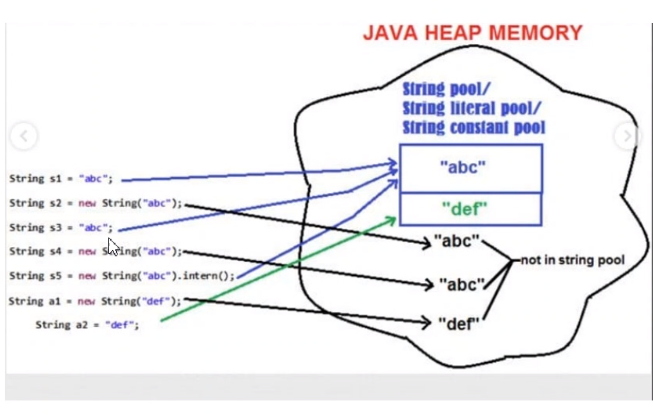
String



String literal s1 string

New keyword is stringpool

S



07/08/2023

String builder mutable non thread safe not a synchronized it is efficient.1.5

String buffer mutable thread safe synchronized1.4



Slow

fast

08/08/2023

Maven

Project management tool

Local repository apache

central repository

Remote repository \_ which is not in the central (cloud)

Pom.xml