**DAY-07**

**ARRAY LIST**

ArrayList:-

-> Array is having fixed size once it is declared we cant change

-> to increase the array size dynamically we have to use arrayList it belongs to collection framework

FrameWork:-

FrameWork is nothing but predefined methods and it will guide the program

Syntax:-

list<Wrapper class> ArrayListName = new ArrayList<>();

ex:-

ArrayList <Integer> leaders = new ArraList<>();

->List is an interface and interface contains incomplete methods

-> Wrapper class is not an primitive data-type it's a class and in collections we have to use wrapper classes only

-<> angular brackets called Generics(<>)

->ArrayList is a class and which contains complete methods

-> ArrayList is an dynamic Array, if my array size is 2 if im trying to add third element the size will be doubled and it'll copy

previous elements vice-versa the array will become half(when you remove elements)

-> As arrayList is a class you have to import the ArraList class so we can access all the predefined methods of ArrayList

2D Array:

2-dimensional arrays consists of rows and coloumns

Syntax:

int[][] arr = new int[3][2]; -> the first square represents "rows" and second square bracket represents "coloumns"

-> While declaring the array we have to declare number of rows and coloumn numbers are optional

-> Each row will act as an individual array

arr[5] = {1,2,3} size = 3, length = 5

for 2-d array the size will be first sq.bracket

Code:

public class q2{

public static void main(String[] args){

Demo d1 = new Demo();

Demo d2 = new Demo();

}

}

//when craeted objects constructor invoked automatically.

class Demo{

static{//syntax for static block-(singleton-block)

System.out.println("stb");

}

Demo(){//constructor->initialization of objectts

System.out.println("cns");

}

}

//we created two objects so 2 times the constructor will be called!

**STATIC CONCEPT IN JAVA:**

Constructor:

->Constructores are used to initialize the objects

->for every class there will be a default constructor

->whenever we create an object the constructor will be called automatically! there,the object will be created

->Constructor having class name and method properties

Static\_Block:

->Static block will be created with just static keyword

->static methods called automatically no need to call with either objects nor class

Static Method:

->For Static method no need to call with objects you can directly call with class name and here Static

will act as instance or an object

import java.util.\*;

class Main{

public static void main(String[] args){

Student obj = new Student();

obj.setAge(19);

int res = obj.getAge();

obj.setName("lavanya");

String res1 = obj.getName();

obj.setIspassedout(true);

boolean res2 = obj.getIspassedout();

obj.setMarks(89);

float res3 = obj.getMarks();

obj.setErp(220303);

long res4 = obj.getErp();

System.out.println(res);

System.out.println(res1);

System.out.println(res2);

System.out.println(res3);

System.out.println(res4);

}

}

class Student{

int age;

String name;

boolean passed;

float marks;

long ERP;

void setAge(int age1){

this.age=age1;

}

int getAge(){

return age;

}

void setName(String name1){

this.name = name1;

}

String getName(){

return name;

}

void setIspassedout(boolean passed1){

this.passed = passed1;

}

boolean getIspassedout(){

return passed;

}

void setMarks(float marks1){

this.marks=marks1;

}

float getMarks(){

return marks;

}

void setErp(long ERP1){

this.ERP=ERP1;

}

long getErp(){

return ERP;

}

}