1. Create a class that keeps track of the number of instances created. Implement a static variable and method to accomplish this.

Ans:- class InstanceCounter1 {

private static int count = 0; // static variable to keep track of the number of instances

InstanceCounter1() {

count++;

}

public static int getInstanceCount() {

return count;

}

}

public class InstanceCounter {

public static void main(String[] args) {

// Create instances of the class

InstanceCounter1 obj1 = new InstanceCounter1();

InstanceCounter1 obj2 = new InstanceCounter1();

InstanceCounter1 obj3 = new InstanceCounter1();

// Get the number of instances created

int instanceCount = InstanceCounter1.getInstanceCount();

System.out.println(instanceCount); // Output: 3

}

}

1. Write a program and create a constructor with parameters and initialise and the variable using constructor.

Ans:-

class Student {

int age, rollNo;

String name;

public Student(int age, String name){

this.age = age;

this.name = name;

}

public Student(int age, String name, int rollNo){

this.age = age;

this.name = name;

this.rollNo = rollNo;

}

public void disp1(){

System.out.println("age is:- "+age);

System.out.println("name is:- "+name);

}

public void disp2(){

System.out.println("age is:- "+age);

System.out.println("name is:- "+name);

System.out.println("rollNo is:- "+rollNo);

}

}

public class Cons{

public static void main(String[] args){

Student st1 = new Student(20,"rahul");

st1.disp1();

Student st2 = new Student(24,"rehan",46);

st2.disp2();

}

}

1. Using a private keyword for a variable and use setter and getter methods to initialise and print values.

Ans:-

class Demo {

private String name;

private int age;

public void setAge(int age){

this.age = age;

}

public int getAge(){

return age;

}

public void setName(String name){

this.name = name;

}

public String getName(){

return name;

}

public void show(){

System.out.println("age = "+age+" name = "+name);

}

}

public class Getter{

public static void main(String[] args){

Demo obj1 = new Demo();

Demo obj2 = new Demo();

obj1.setAge(20);

System.out.println(obj1.getAge());

obj2.setAge(44);

System.out.println(obj2.getAge());

obj1.setName("rahul");

String name1 = obj1.getName();

System.out.println(name1);

obj2.setName("rehan");

String name2 = obj2.getName();

System.out.println(name2);

}

}

1. Write a program to call a method without creating an object of a class.

Ans:-

class Demo3{

static float pi = 3.14f;

public static void AreaCircle(float rad)

{

float area = pi\*rad\*rad;

System.out.println("area of circle = "+area);

}

}

public class IndependentObj {

public static void main(String[] args){

Demo3.AreaCircle(20);

}

}

1. Write a program which has a static block and constructor overloading , initialise variables using constructors and print it.

class Demo4{

public int age;

public String name;

Demo4()

{

System.out.println("Default constructor");

}

Demo4(int age)

{

super();

this.age = age;

}

Demo4(String name)

{

super();

this.name = name;

}

Demo4(int age, String name)

{

super();

this.age = age;

this.name = name;

}

static

{

System.out.println("static block");

}

}

public class Static {

public static void main(String[] args){

Demo4 obj1 = new Demo4(24);

System.out.println(obj1.age);

Demo4 obj2 = new Demo4("usama");

System.out.println(obj2.name);

Demo4 obj3 = new Demo4(32,"khan");

System.out.println("age = "+obj3.age+" name = "+obj3.name);

}

}