LAB Assignment –2

1. Write a program to print numbers from 1 to 10, but stop printing when the number7 is reached. Use the break statement to exit the loop when the number reaches 7.

class J1\_Number

{

public static void main(String[] args)

{

for (int i = 1; i <= 10; i++)

{

if (i == 7)

{

break;

}

System.out.println(i);

}

}

}

Output:

1

2

3

4

5

6

1. Write a program to print the numbers from 1 to 10, but skip the number5. Use the continue statement to skip printing when the number is 5.

class J2\_Number

{

public static void main(String[] args)

{

for (int i = 1; i <= 10; i++)

{

if (i == 5)

{

continue;

}

System.out.println(i);

}

}

}

Output:

1

2

3

4

6

7

8

9

10

1. Write a program to calculate the sum of all the elements in a one-dimensional array of integers.

class J3\_1DArray

{

public static void main(String[] args)

{

int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int sum = 0;

for (int i = 0; i < numbers.length; i++)

{

sum += numbers[i];

}

System.out.println("The sum of the array elements is: " + sum);

}

}

Output:

The sum of the array elements is: 55

1. Write a program to calculate the sum of all elements in a jagged array of integers.

class J4\_JaggedArray

{

public static void main(String[] args)

{

int arr[][] ={

{1,2,3,4}

};

int sum = 0;

for (int i = 0; i < arr.length; i++)

{

for(int a:arr[i])

{

sum = sum + a;

}

}

System.out.println("Sum of all the elements of an array: " + sum);

}

}

Output:

Sum of all the elements of an array: 10

1. Write a program to create a Car class with the following attributes: brand (String),model (String), year (int) Define methods to set the values of these attributes and display the car's information.

class Method

{

String brand;

String model;

int year;

public void setBrand(String brand)

{

this.brand = brand;

}

public void setModel(String model)

{

this.model = model;

}

public void setYear(int year)

{

this.year = year;

}

public void displayCarInfo() {

System.out.println("Car Information:");

System.out.println("Brand is: " + brand);

System.out.println("Model is: " + model);

System.out.println("Year is: " + year);

}

}

class Main

{

public static void main(String[] args)

{

Method myCar = new Method();

myCar.setBrand("Toyota");

myCar.setModel("Corolla");

myCar.setYear(2020);

myCar.displayCarInfo();

}

}

Output:

Car Information:

Brand is: Toyota

Model is: Corolla

Year is: 2020

1. Write a programming to implement encapsulation in a Person class with the name,age, and address attributes and create getter and setter methods to access and update the private variables.

class Person

{

private String name;

private int age;

private String address;

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

public int getAge()

{

return age;

}

public void setAge(int age)

{

this.age = age;

}

public String getAddress()

{

return address;

}

public void setAddress(String address)

{

this.address = address;

}

public void displayPersonInfo()

{

System.out.println("Person Information:");

System.out.println("Name: " + name);

System.out.println("Age: " + age);

System.out.println("Address: " + address);

}

}

class Main

{

public static void main(String[] args)

{

Person person = new Person();

person.setName("Jenish Kidecha");

person.setAge(19);

person.setAddress("Rajkot");

person.displayPersonInfo();

System.out.println("\nAccessing values using getter methods:");

System.out.println("Name is: " + person.getName());

System.out.println("Age is: " + person.getAge());

System.out.println("Address is: " + person.getAddress());

}

}

Output:

Person Information:

Name: Jenish Kidecha

Age: 19

Address: Rajkot

Accessing values using getter methods:

Name is: Jenish Kidecha

Age is: 19

Address is: Rajkot

1. Write a program to demonstrate abstraction.

abstract class Animal

{

public abstract void sound();

public void sleep()

{

System.out.println("The animal is sleeping.");

}

}

class Dog extends Animal

{

public void sound()

{

System.out.println("The dog barks.");

}

}

class Cat extends Animal

{

public void sound()

{

System.out.println("The cat meows.");

}

}

class Main

{

public static void main(String[] args)

{

Animal myDog = new Dog();

myDog.sound();

myDog.sleep();

Animal myCat = new Cat();

myCat.sound();

myCat.sleep();

}

}

Output:

The dog barks.

The animal is sleeping.

The cat meows.

The animal is sleeping.

1. Write a program to demonstrate method overloading and method overriding.

class Program {

static int add(int a, int b) {

return a + b;

}

static Double add(Double a, Double b) {

return a + b;

}

}

class Parent {

void print() {

System.out.println("hello world");

}

}

class Subclass extends Parent {

@Override

void print() {

System.out.println("subclass");

}

}

class Main {

public static void main(String[] args) {

System.out.println(Program.add(90, 78));

System.out.println(Program.add(45.45, 23.88));

Subclass s1 = new Subclass();

s1.print();

}

}

Output:

168

69.33

Subclass

1. Demonstrate multilevel inheritance, where a class inherits from another class, which itself inherits from another class.

class Vehicle

{

public void start()

{

System.out.println("The vehicle is starting.");

}

}

class Car extends Vehicle

{

public void drive()

{

System.out.println("The car is driving.");

}

}

class ElectricCar extends Car

{

public void charge()

{

System.out.println("The electric car is charging.");

}

}

class Main

{

public static void main(String[] args)

{

ElectricCar myElectricCar = new ElectricCar();

myElectricCar.start();

myElectricCar.drive();

myElectricCar.charge();

}

}

Output:

The vehicle is starting.

The car is driving.

The electric car is charging.

10.Demonstrate hierarchical inheritance, where multiple subclasses inherit from a single superclass.

class student

{

public void information()

{

System.out.println("I am student of TNR collage...");

}

}

class teacher extends student

{

public void details ()

{

System.out.println("I am teacher of TNR collage....");

}

}

class principal extends student

{

public void info()

{

System.out.println("I am principal of TNR collage...");

}

}

class Main

{

public static void main(String[] args)

{

principal p1= new principal();

p1.info();

p1.information ();

teacher t1= new teacher ();

t1.details ();

t1.information ();

}

}

Output:

I am principal of TNR collage...

I am student of TNR collage...

I am teacher of TNR collage....

I am student of TNR collage...

11.Write a program to demonstrate constructor.

class Main

{

String name;

Main()

{

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

name="Jenish Kidecha";

}

public static void main(String[] args)

{

Main obj =new Main();

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

}

}

Output:

the constructor...

the name is :Jenish Kidecha

12. Write a program to demonstrate constructor overloading.

class Main

{

String name;

int rollno = 5;

Main()

{

System.out.println("The first constructor...");

name = "Jenish Kidecha";

}

Main(String name)

{

System.out.println("The second constructor...");

this.name = name;

}

public static void main(String[] args)

{

Main obj1 = new Main();

System.out.println("The name is: " + obj1.name);

System.out.println("The roll number is: " + obj1.rollno);

Main obj2 = new Main("jenish kidecha");

System.out.println("The name is: " + obj2.name);

System.out.println("The roll number is: " + obj2.rollno);

}

}

Output:

The first constructor...

The name is: Jenish Kidecha

The roll number is: 5

The second constructor...

The name is: jenish kidecha

The roll number is: 5