|  |  |
| --- | --- |
| Name | Umar Hayyat |
| Roll No | 2019-EE-360 |

**Lab 7**

**Writing Class**

**Objectives:**

In this lab:

* I will learn how we declare a method.
* I will learn how we use setter and getter method.
* I will learn about toString method.

**Task 1:**

Design and implement a class called Sphere that contains instance data that represents the sphere’s diameter. Define the Sphere constructor to accept and initialize the diameter, and include getter and setter methods for the diameter. Include methods that calculate and return the volume and surface area of the sphere. Include a toString method that returns a one-line description of the sphere. Create a driver class called MultiSphere, whose main method instantiates and updates several Sphere objects.

**Code:**

public class MultiSphere {

public static void main(String[] args){

Sphere s1 = new Sphere();

s1.setdiam(6); s1.surf(); s1.volum();

System.out.println("Sphere 1");

System.out.println(s1.toString());

Sphere s2 = new Sphere();

s2.setdiam(14); s2.surf(); s2.volum();

System.out.println("Sphere 2");

System.out.println(s2.toString());

Sphere s3 =new Sphere();

s3.setdiam(18); s3.surf(); s3.volum();

System.out.println("Sphere 3");

System.out.println(s3.toString());

}

}

public class Sphere {

private int diam;

private double rad;

private double surf;

private double volum;

private final double pie = 22.7;

public Sphere(){}

public void setdiam(int newdiam){

diam = newdiam; }

public int getdiam() {

return diam;}

public void surf(){

rad = diam/2;

surf = (double)((4\*pie\*(rad\*rad\*rad))/3);}

public void volum(){

rad = diam/2;

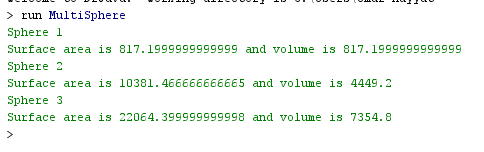
volum = (double)(4\*pie\*(rad\*rad));}

public String toString(){

return "Surface area is "+surf+" and volume is "+volum;

}}

**Output:**



**Task 2:**

Design and implement a class called Car that contains instance data that represents the make, model, and year of the car. Define the Car constructor to initialize these values. Include getter and setter methods for all instance data, and a toString method that returns a one-line description of the car. Create a driver class called CarTest, whose main method instantiates and updates several Car objects.

**Code:**

public class CarTest {

public static void main(String[] args){

Car car1 = new Car();

car1.setMake("Mehran");

car1.setModel("Civic");

car1.setYear(2020);

Car car2 = new Car();

car2.setMake("Corolla");

car2.setModel("Gli");

car2.setYear(2015);

Car car3 = new Car();

car3.setMake("Suzuki");

car3.setModel("Wagonar");

car3.setYear(2016);

System.out.println("The data of 1 Car :");

System.out.println(car1.toString());

System.out.println("The data of 2 Car :");

System.out.println(car2.toString());

System.out.println("The data of 3 Car :");

System.out.println(car3.toString());

}

}

public class Car {

private String make;

private String model;

private int year;

public Car(){}

public void setMake(String newmake){

make = newmake;}

public void setModel(String newmodel){

model = newmodel;}

public void setYear(int newyear){

year = newyear;}

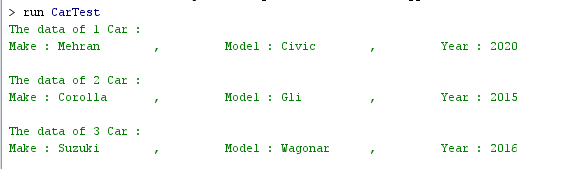
public String toString(){

return "Make : "+make+"\t,\tModel : "+model+"\t,\tYear : "+year+"\n";

}

}

**Output:**



**Task 3:**

Using the Die class defined in example, design and implement a class called PairOfDice, composed of two Die objects. Include methods to set and get the individual die values, a method to roll the dice, and a method that returns the current sum of the two die values. Create a driver class called RollingDice2 to instantiate and use a PairOfDice object.

**Code:**

public class RollingDie2 {

public static void main(String[] args){

PairofDies die1 = new PairofDies();

PairofDies die2 = new PairofDies();

int sum = die1.roll()+die2.roll();

System.out.println("Sum of two dies : "+die1.getValue());

}

}

public class PairofDies{

private final int MAX = 4;

private int faceValue;

public PairofDies(){

faceValue = 1;}

public int roll(){

faceValue = (int)(Math.random() \* MAX) + 1;

return faceValue;}

public void setFaceValue (int value){

faceValue = value;}

public int getFaceValue(){

return faceValue;}

public int getValue(){

return faceValue + faceValue;

}

}

**Output:**



**Task 4:**

Design and implement a class called Flight that represents an airline flight. It should contain instance data that represents the airline name, flight number, and the flight’s origin and destination cities. Define the Flight constructor to accept and initialize all instance data. Include getter and setter methods for all instance data. Include a toString method that returns a one-line description of the flight. Create a driver class called FlightTest, whose main method instantiates and updates several Flight objects.

**Code:**

public class FlightTest {

public static void main(String[] args){

Flight flight1 = new Flight("PIA",556,"Karachi","Lahore");

Flight flight2 = new Flight ("ABQ",377,"Lahore","London");

Flight flight3 = new Flight ("SEP",234,"Islamabad","Istanbul");

System.out.println("First Flight information :");

System.out.println(flight1.toString());

System.out.println("Second Flight information :");

System.out.println(flight2.toString());

System.out.println("Third Flight information :");

System.out.println(flight3.toString());

}}

public class Flight {

private String air;

private int num;

private String origion;

private String city;

public Flight(String newair,int newnum,String neworigion,String newcity){

air = newair;

num = newnum;

origion = neworigion;

city = newcity;}

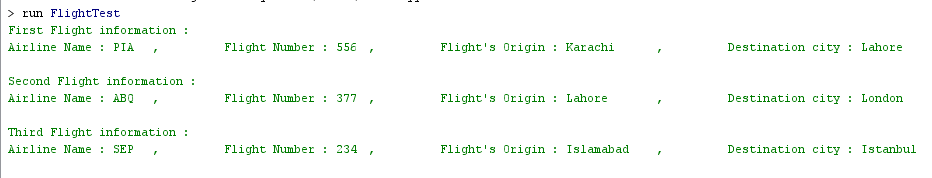
public String toString(){

return "Airline Name : "+air+"\t,\tFlight Number : "+num+"\t,\tFlight's Origin : "+origion+"\t,\tDestination city : "+city+"\n" ;

}

}

**Output:**



**Conclusion:**

In this lab, I learnt how we write classes. I also learnt how we declare a data and method in class. I also learnt about instance data and encapsulation. I also learnt how we write and functions in class. I also learnt how we use or get and change the value of variable by using getter and setter method respectively.