|  |  |
| --- | --- |
| **Name** | Muhammad Asad |
| **Registration no.** | 2019EE383 |
| **Group** | G1 |

**Programming Fundamentals**

**Lab # 07 (Class Writing)**

**Objectives:**

In this lab we learnt about writing class and how to make a class of sphere, car.

**Task # 01:**

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 x = new Sphere();

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

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

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

Sphere y = new Sphere();

y.setdiam(9); y.surf(); y.volum();

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

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

}}

public class Sphere {

privateintdiam;

private double radi;

private double surf;

private double volum;

private final double pie = 22.7;

public Sphere(){}

public void setdiam(intnewdiam){

diam = newdiam;}

publicintgetdiiam(){

returndiam;}

public void surf(){

radi = diam/2;

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

public void volum(){

radi = diam/2;

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

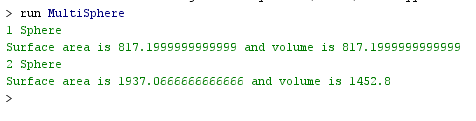
public String toString(){

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

}

}

**Output:**



**Task # 02:**

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 x1 = new Car();

x1.setMake("Honda");

x1.setModel("Civic");

x1.setYear(2019);

Car y2 = new Car();

y2.setMake("Corolla");

y2.setModel("Gli");

y2.setYear(2019);

Car z3 = new Car();

z3.setMake("Suzuki");

z3.setModel("Wagonar");

z3.setYear(2018);

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

System.out.println(x1.toString());

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

System.out.println(y2.toString());

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

System.out.println(z3.toString());

}}

public class Car {

private String make;

private String model;

privateint year;

public Car(){}

public void setMake(String newmake){

make = newmake;}

public void setModel(String newmodel){

model = newmodel;}

public void setYear(intnewyear){

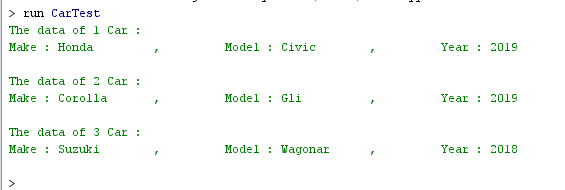
year = newyear;}

public String toString(){

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

}

**Output:**



**Task # 03:**

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 RollingDie\_2 {

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 = 5;

privateintfaceValue;

publicPairofDies(){

faceValue = 1;}

publicint roll(){

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

returnfaceValue;}

public void setFaceValue (int value){

faceValue = value;}

publicintgetFaceValue(){

returnfaceValue;}

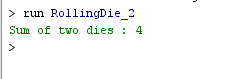
publicintgetValue(){

returnfaceValue + faceValue;

}

}

**Output:**



**Task # 04:**

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 x1 = new Flight("PIA",4556,"kis","gapur");

Flight x2 = new Flight ("Air",5477,"Sapur","many");

Flight x3 = new Flight ("Sa",348,"many","Tkey");

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

System.out.println(x1.toString());

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

System.out.println(x2.toString());

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

System.out.println(x3.toString());

}}

public class Flight {

private String air;

privateint numb;

private String orig;

private String city;

public Flight(String newair,intnewnumb,Stringneworig,Stringnewcity){

air = newair;

numb = newnumb;

orig = neworig;

city = newcity;}

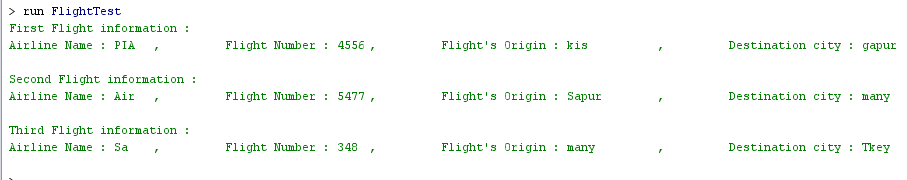
public String toString(){

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

}

}

**Output:**



**Conclusion:**

In this lab we learnt about how to writing different classes and also leant about the different functions of classes as I create the classes e.gsphere , car class.

In this lab we also learnt how we use different variables in a class and compare the two variables.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------