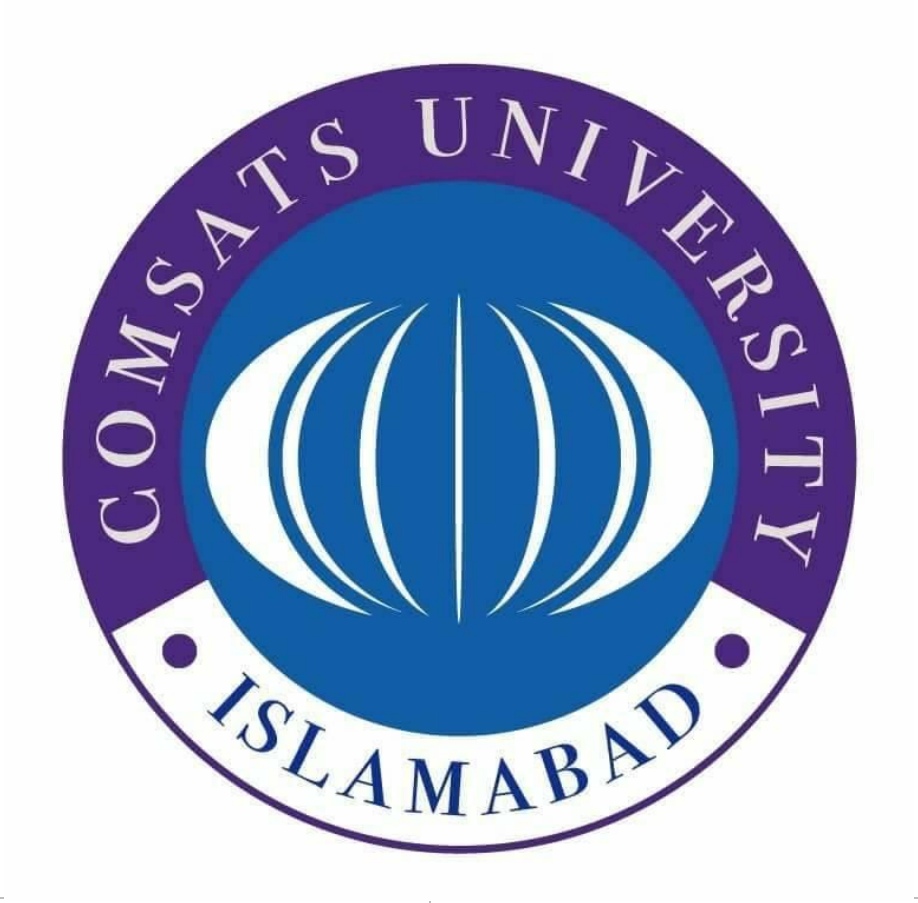

Submitted By	**Habib ur Rehman (116)**
Subject	OOP
Assignment	Assignment 03
Date	Nov 15th , 2024

Submitted to:

Moderator	**Ms, Sajida Kalsoom**

Question no 01:

```
// Create an inheritance hierarchy to represent various types of packages. Use Package as the super class
// of the hierarchy, then include classes TwoDayPackage and OvernightPackage that derive from Package.


class packages{
   protected String Sname;
   protected String Rname;
   protected String Saddress;
   protected String Raddress;
   protected double weight;
   protected double costperounce;


   public packages() {


 }


   public packages(String sname, String rname, String saddress, String raddress, double weight,double c)
{
      Sname = sname;
      Rname = rname;
      Saddress = saddress;
      Raddress = raddress;
      if (weight>0) {
         this.weight = weight;
      }
      else{
         weight=0;
      }
```

```
    if (costperounce>0) {
        this.costperounce=c;
    }
    else{
        costperounce=0;
    }
}

public void setSname(String sname) {
    Sname = sname;
}

public void setRname(String rname) {
    Rname = rname;
}

public void setSaddress(String saddress) {
    Saddress = saddress;
}

public void setRaddress(String raddress) {
    Raddress = raddress;
}

public void setWeight(double weight) {
    if (weight>0) {
        this.weight = weight;

    }
```

```
    else{
        System.out.println("Invalid Entry");
    }
}

 public void setCostperounce(double costperounce) {
    if(costperounce>0)
    this.costperounce = costperounce;
    else{
        System.out.println("Invalid Entry");
    }
}


public double getCostperounce() {
    return costperounce;
}

public String getSname() {
    return Sname;
}

public String getRname() {
    return Rname;
}

public String getSaddress() {
    return Saddress;
}
```

```
    public String getRaddress() {
        return Raddress;
    }

    public double getWeight() {
        return weight;
    }

     public double calculateShippingCost() {
        return weight * costperounce;
    }

    @Override
    public String toString() {
        return "packages [Sname=" + Sname + ", Rname=" + Rname + ", Saddress=" + Saddress + ",
Raddress=" + Raddress
                + ", weight=" + weight + ", costperounce=" + costperounce + "]";
    }

}
```

```
class TwoDayPackage extends packages{
    private double flatfee;

    public TwoDayPackage() {

    }

    public TwoDayPackage(String sname, String rname, String saddress, String raddress, double weight,
double c,
            double flatfee) {
        super(sname, rname, saddress, raddress, weight, c);
        this.flatfee = flatfee;
    }

    public double getFlatfee() {
        return flatfee;
    }

    public void setFlatfee(double flatfee) {
        this.flatfee = flatfee;
    }


     public double calculateShippingCost() {
        return flatfee+(weight * costperounce);
    }

    @Override
```

```
    public String toString() {
        return "TwoDayPackage [Sname=" + Sname + ", Rname=" + Rname + ", Saddress=" + Saddress + ",
Raddress="
                + Raddress + ", weight=" + weight + ", costperounce=" + costperounce + ", flatfee=" + flatfee +
"]";
    }



}

class OvernightPackage extends packages{
    private double additonalfee;



    public OvernightPackage(double additonalfee) {
        this.additonalfee = additonalfee;
    }



    public OvernightPackage(String sname, String rname, String saddress, String raddress, double weight,
double c,
            double additonalfee) {
        super(sname, rname, saddress, raddress, weight, c);
        this.additonalfee = additonalfee;
    }



    public double getAdditonalfee() {
        return additonalfee;
    }
```

```
    public void setAdditonalfee(double additonalfee) {
        this.additonalfee = additonalfee;
    }
    public double calculateShippingCost() {

        double baseCost = super.calculateShippingCost();
        return baseCost + additonalfee;
    }

    @Override
    public String toString() {
        return "OvernightPackage [Sname=" + Sname + ", Rname=" + Rname + ", Saddress=" + Saddress + ",
Raddress="
                + Raddress + ", weight=" + weight + ", costperounce=" + costperounce + ", additonalfee=" +
additonalfee
                + "]";
    }



}



public class task1{
    public static void main(String[] args) {

        packages regularPackage = new packages("habib", "jamil", "159", "g13/1", 10, 2.5);
        TwoDayPackage twoDayPackage = new TwoDayPackage("ha", "ta", "1233", "347", 5, 3.0, 10);
```

```
        OvernightPackage overnightPackage = new OvernightPackage("ali", "Rana", "118 G-13.4", "567 Pine St", 8, 4.0, 20);
        // System.out.println(regularPackage.toString());
        System.out.println(twoDayPackage.calculateShippingCost());
        System.out.println(overnightPackage.calculateShippingCost());
        System.out.println(twoDayPackage.toString());
        System.out.println(overnightPackage.toString());



    }
}
```

Question no02:

```
abstract class Person {
    protected String name;

    public Person(String n) {
        this.name = n;
    }

    public void setName(String n) {
        this.name = n;
    }

    public String getName() {
        return name;
    }

    public abstract boolean isOutstanding();
```

```
}

class Student extends Person {
    private double CGPA;

    public Student(String n, double CGPA) {
        super(n);
        this.CGPA = CGPA;
    }

    public void setCGPA(double CGPA) {
        this.CGPA = CGPA;
    }

    public double getCGPA() {
        return CGPA;
    }

    @Override
    public boolean isOutstanding() {
        return CGPA > 3.5;
    }
}

class Professor extends Person {
    private int numberOfPublications;

    public Professor(String name, int n) {
        super(name);
```

```
        this.numberOfPublications = n;
    }

    public void setNumberOfPublications(int n) {
        this.numberOfPublications = n;
    }

    public int getNumberOfPublications() {
        return numberOfPublications;
    }

    @Override
    public boolean isOutstanding() {
        return numberOfPublications > 50;
    }
}

public class Main {
    public static void main(String[] args) {
        Person[] people = new Person[3];
        people[0] = new Student("habib", 3.73);
        people[1] = new Professor("Abu Bakar", 40);
        people[2] = new Professor("Ali", 60);



        for (Person person : people) {
            System.out.println(person.getName() + " is outstanding: " + person.isOutstanding());
        }
```

```
        Professor professor = (Professor) people[1];
        professor.setNumberOfPublications(100);

        System.out.println(professor.getName() +
 " is outstanding after update: "
+ professor.isOutstanding());
    }
}
```

Question no 03:

```
abstract class Convert {
    protected double val1;
    protected double val2;

    public Convert(double val1) {
        this.val1 = val1;
    }
        public double getVal1() {
        return val1;
    }


    public void setVal1(double val1) {
```

```
        this.val1 = val1;
    }

        public double getVal2() {
        return val2;
    }


    public abstract void compute(){};


    @Override
    public String toString() {
        return "Initial Value: " + val1 + "  Converted Value: " + val2;
    }


}

class Litretogallon extends Convert {

    public Litretogallon(double v) {
        super(v);
    }

    @Override
    public void compute() {
```

```
        val2 = val1 * 0.264172;
    }
}

class ForentoCel extends Convert {

    public ForentoCel(double v) {
        super(v);
    }

    @Override
    public void compute() {
        val2 = (val1 - 32) * 5 / 9;
    }
}

class FToM extends Convert {

    public FToM(double val1) {
        super(val1);
    }

    @Override
    public void compute() {
        val2 = val1 * 0.3048;
    }
}
```

```
public class task3 {
   public static void main(String[] args) {

      Litretogallon litersToGallons = new Litretogallon(48);
      ForentoCel fahrenheitToCelsius = new ForentoCel(94.5);
      FToM feetToMeters = new FToM(34);

      litersToGallons.compute();
      fahrenheitToCelsius.compute();
      feetToMeters.compute();
      System.out.println(litersToGallons.toString());
      System.out.println(fahrenheitToCelsius.toString());
      System.out.println(feetToMeters.toString());

      System.out.println("Liters to Gallons: Initial Value = " + litersToGallons.getVal1() + "
Converted Value = " + litersToGallons.getVal2());
      System.out.println("Fahrenheit to Celsius: Initial Value = " + fahrenheitToCelsius.getVal1() +
" Converted Value = " + fahrenheitToCelsius.getVal2());
      System.out.println("Feet to Meters: Initial Value = " + feetToMeters.getVal1() + " Converted
Value = " + feetToMeters.getVal2());
   }
}
```