Bahria University,

Karachi Campus

LAB EXPERIMENT NO.

08

LIST OF TASKS

TASK NO	OBJECTIVE
1	Create a payroll system using classes, inheritance and polymorphism Four types of employees paid weekly a. Salaried employees: fixed salary irrespective of hours b. Hourly employees: 40 hours salary and overtime (> 40 hours) c. Commission employees: paid by a percentage of sales Base-plus-commission employees: base salary and a percentage of sales
2	You have to implement the following diagram including some attributes and other functions: Shapes and its child classes

Submitted On:
16-May-2023
(Date: DD/MM/YY)

Task No. 1: Create a payroll system using classes, inheritance and polymorphism

Four types of employees paid weekly

a. Salaried employees: fixed salary irrespective of hours
b. Hourly employees: 40 hours salary and overtime (> 40 hours)
c. Commission employees: paid by a percentage of sales
d. Base-plus-commission employees: base salary and a percentage of sales

Solution:

Main Method:

```
public class Mavenproject32 {
    public static void main(String[] args) {
            Employee e1 = new SalariedEmployee("Talha","Ahmed","213-311-3222",200000);
        Employee e2 = new CommissionEmployee("Haris","Talha","222-22-2222",20000,2.58);
Employee e3 = new BasePlusCommissionEmployee("Jawad","Ahmed","332-23-
4332",15000,2.3,80000);
        Employee e4 = new HourlyEmployee("Fahad","Hameed","321-23-2985",3000,30);
        System.out.println("First Employee:");
        System.out.println(e1.tostring());
        System.out.println("Salary: "+e1.earnings());
        System.out.println("\nSecond Employee:");
        System.out.println(e2.tostring());
        System.out.println("Salary: "+e2.earnings());
        System.out.println("\nThird Employee:");
        System.out.println(e3.tostring());
        BasePlusCommissionEmployee currentEmployee = (BasePlusCommissionEmployee) e3;
        double oldbasesalary = currentEmployee.getBaseSalary();
        System.out.println("Old Base Salary: "+oldbasesalary);
        currentEmployee.setBaseSalary(1.10*oldbasesalary);
      System.out.println("New salary with 10% increase is: "+currentEmployee.getBaseSalary());
        System.out.println("\nFourth Employee: ");
        System.out.println(e4.tostring());
        System.out.println("Salary: "+e4.earnings());    }
    }
```

Class Of Employee:

```
public class Employee {
      private String fName;
    private String lName;
```

```
    private String CNIC;
  public  Employee(){
        fName="Not Available";
        lName="Not Available";
        CNIC="0";  }
   public Employee(String fName,String lName,String CNIC){
        this.fName=fName;
        this.lName=lName;
        this.CNIC=CNIC;    }
    public String getfName(){
        return fName;    }
    public void setfName(String fName){
 this.fName=fName;    }
    public String getlName() {
        return lName; }
    public void setlName(String lName) {
        this.lName = lName;  }
  public String getCNIC() {
        return CNIC;    }
    public void setCNIC(String CNIC) {
        this.CNIC = CNIC; }
    public String tostring(){
        return fName+" "+lName+" CNIC# "+CNIC;    }
    public double earnings(){
        return  0.00;    }
 }
```

Child Class (Salaried Employee):

```
 public class SalariedEmployee extends Employee {
    private double weeklysalary;
  public SalariedEmployee()  {
        super();
        this.weeklysalary = 0;    }
    public SalariedEmployee(String fName,String lName,String CNIC, double weeklysalary)    {
        super(fName,lName,CNIC);
        this.weeklysalary = weeklysalary;    }
    public double getWeeklysalary() {
        return weeklysalary;    }
```

```
    public void setWeeklysalary(double weeklysalary) {
      if (weeklysalary < 0)       {
         System.out.println("Weekly Salary cannot be negative");       }
      else      {
         this.weeklysalary = weeklysalary;       }    }
    @Override
    public String tostring()    {
       return "Salaried Employee: "+super.tostring();    }
    @Override
    public double earnings()    {
       return weeklysalary;    }    }
```

Child Class (HourlyEmployee):

```
  public class HourlyEmployee extends Employee{
     private double wage;
     private double hours;
     public HourlyEmployee()    {
        super();
        this.hours = 0;
        this.wage  = 0;
     }
     public HourlyEmployee(String fName,String lName,String CNIC,double wage,double
  hours)  {
        super(fName,lName,CNIC);
        this.hours = hours;
        this.wage = wage;
     }
     public double getWage() {
        return wage;
     }
     public void setWage(double wage) {
        if (wage < 0) {
           System.out.println("Wage cannot be below 0");
        }
        else{
          this.wage = wage;
        }
```

```
    }
    public double getHours() {
       return hours;
    }
    public void setHours(double hours) {
       if (hours < 0)        {
          System.out.println("Hours cannot be negative");      }
       else    {
           this.hours = hours;     }
    }

    public String tostring()    {
       return "Hourly Employee: "+super.tostring();    }

    public double earnings(){
       if (hours <= 40){
          return wage*hours;      }
       else    {
          return 40*wage +(hours-40)*wage*1.5;        }
    }
  }
```

Child Class (CommissionEmployee):

```
public class CommissionEmployee  extends Employee{
   private double grossSales;
   private double commissionRate;
   public CommissionEmployee()    {
      super();
      this.commissionRate= 0;
      this.grossSales = 0;    }
   public CommissionEmployee(String fName,String lName,String CNIC, double grossSales,
double commissionRate)    {
      super(fName,lName,CNIC);
      this.commissionRate =commissionRate;
      this.grossSales=grossSales;  }
   public double getGrossSales() {
      return grossSales;    }
   public void setGrossSales(double grossSales) {
      if (grossSales < 0)        {
         System.out.println("Gross Sales cannot be negative");
```

```
        }
        else  {
          this.grossSales = grossSales;           }
    }
    public double getCommissionRate() {
        return commissionRate;
    }
    public void setCommissionRate(double commissionRate) {
          if (commissionRate < 0){
           System.out.println("Commision Rate cannot be negative");
          }
        else       {
        this.commissionRate = commissionRate;          }
    }
    @Override
    public String tostring()    {
        return "Commission Employee: "+super.tostring();
    }
    @Override
    public double earnings()    {
        return grossSales* commissionRate;    }
}
```

Child Class (BasePlusCommissionEmployee):

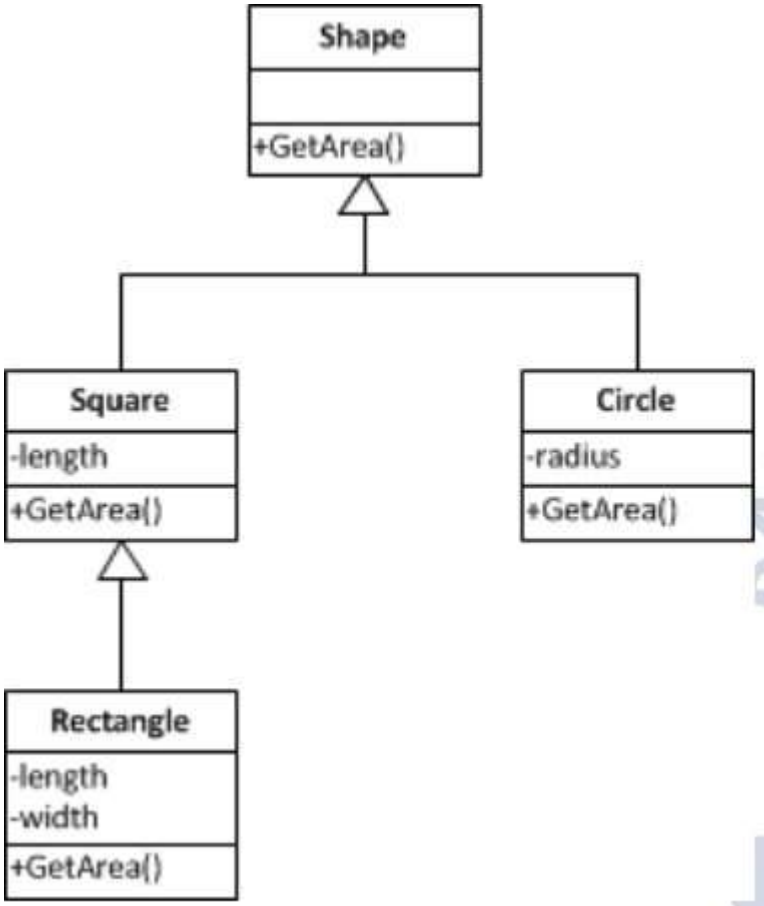
```
public class BasePlusCommissionEmployee  extends CommissionEmployee{
    private double baseSalary;
    public BasePlusCommissionEmployee()    {
        super();
        this.baseSalary = 0;
    }
    public BasePlusCommissionEmployee(String fName,String lName,String CNIC, double
grossSales, double commissionRate, double baseSalary)    {
        super(fName,lName,CNIC,grossSales,commissionRate);
        this.baseSalary = baseSalary;
    }
    public double getBaseSalary() {
        return baseSalary;
    }
    public void setBaseSalary(double baseSalary) {
        if (baseSalary < 0)       {
            System.out.println("Base Salary cannot be negative");       }
```

```
        else  {
            this.baseSalary = baseSalary;  }
    }
    @Override
    public String tostring()  {
        return "Base plus "+super.tostring();
    }
    @Override
    public double earnings()    {
        return baseSalary*super.earnings();    }
}
```

Output:

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.1.0:exec (default-cli) @ mavenproject32 ---
First Employee:
Salaried Employee: Ahad Ahmed CNIC# 1234-12345-2345
Salary: 20000.0

Second Employee:
Commission Employee: Haris Tahiri CNIC# 222-22-2222
Salary: 15480.0

Third Employee:
Base plus Commission Employee: Jawad Ahmed CNIC# 332-23-4332
Old Base Salary: 80000.0
New salary with 10% increase is: 88000.0

Fourth Employee:
Hourly Employee: Qazi Khizar Ali CNIC# 321-23-2985
Salary: 210000.0
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Task No. 2: You have to implement the following diagram including some attributes and other functions.



Solution:

Main Method:

```
public class Mavenproject33 {
    public static void main(String[] args) {
        Shape s1=new Square(15);
    s1.getArea();
    Shape s2=new Rectangle(35,15);
    s2.getArea();
    Shape s3=new Circle(23.5);
    s3.getArea();
```

```
    }
}
```

Class Of Shapes:

```
;public class Shape {
     public void getArea(){
      System.out.println("Hello Shape!");
     }

}
```

Child Class (Circle):

```
public class Circle extends Shape{
    private double radius,pi=3.142;
    Circle(){
        radius=0;
    }
    Circle(double radius){
        this.radius=radius;
    }
    public void getArea(){
System.out.println("\nArea of the circle is:"+pi*radius*radius+" meters square");
    }
}
```

Child Class (Square):

```
public class Square extends Shape{
    private double length;
    public Square(){
        length=0;
    }
    public Square(double length) {
        this.length = length;}
    public double getLength() {
        return length;
    }
    public void setLength(double length) {
        this.length = length;
    }
    public void getArea(){
System.out.println("Area of Square is:"+length*length+" meter square");


    }
```

```
}
```

Child Class (Rectangle):

```
public class Rectangle extends Square{
   private double width;
   Rectangle(){
      super();
      width=0;
   }
   Rectangle(double length,double width){
      super(length);
      this.width=width;
   }
public double getWidth() {
      return width;
}
   public void setWidth(double width) {
      this.width = width;
   }
   public void getArea(){
 System.out.println("\nArea of Rectangle is:"+getLength()*getWidth()+" meters square");
   }
}
```

Output:

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.1.0:exec (default-cli) @ mavenproject33 ---
Area of Square is:225.0 meter square

Area of Rectangle is:525.0 meters square

Area of the circle is:1735.1695 meters square
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.808 s
Finished at: 2023-05-16T01:03:32+05:00
------------------------------------------------------------------------
```