CSE1007 – Java Programming – ELA

Winter 2019-20

LAB ASSIGNMENT – 2

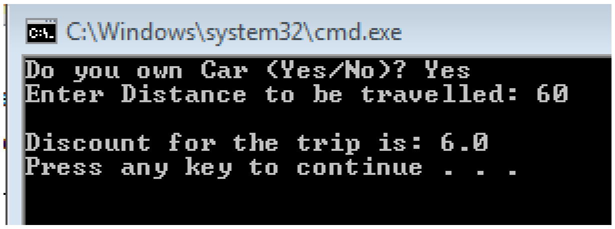
Date – 07-Jan-2020

SATYAM SINGH CHAUHAN
17BCE0581

QUESTION 1

A travels agency named, XYZ, wish to let the customer to use their own vehicles (only vehicle) for travelling trips. So, it is decided to give a discount to the passengers who choose their own vehicle for travel for any distance. The discount amount is calculated as distance * 0.1 i.e 10%. Write an appropriate Java class for the 'Passenger' and test the functionalities of them.

```
import java.util.*;
import java.io.*;

class passenger {
        private double discount;
        private double distance;

        void getDist(double distance)
        {
                distance = this.distance;
        }

        void calDiscount(double distance)
        {
                discount = distance * 0.1;
                System.out.print("\nDiscount for the trip is: ");
                System.out.println(discount);
        }

        public static void main(String[] args)
        {
                Scanner sc = new Scanner(System.in);

                System.out.print("Do you own Car (Yes/No)? ");
                String own_car = sc.next();

                if(own_car.equals("Yes"))
                {
                        System.out.print("Enter Distance to be travelled: ");
                        double distance = sc.nextDouble();
                        passenger p = new passenger();
                        p.getDist(distance);
                        p.calDiscount(distance);
                }
                else
                {
                        System.out.print("Sorry, No Discount Available Now.");
                }
        }
}
```



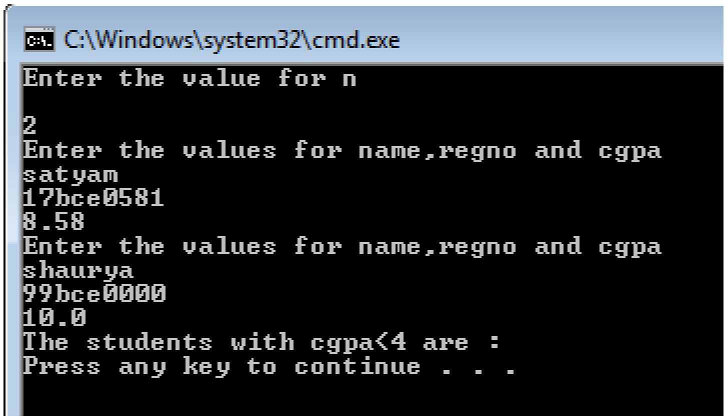
```
Do you own Car (Yes/No)? Yes
Enter Distance to be travelled: 60

Discount for the trip is: 6.0
Press any key to continue . . .
```

QUESTION 2

Write a class definition for ‘stu’ with name, regno, and cgpa values and required methods as members of the class. Create an array ‘A’ of objects of ‘stu’ for ‘n’ number of students in G1 slot. Write a Java program to display the name and registration numbers of the students who have CGPA less than 4 in array A.
[Hint: create the array A first and instantiate the objects in a ‘for’ loop]

```
import java.io.*;
import java.util.*;
public class student
{
        String name;
        String regno;
        double cgpa;
        student()
        {
                name="";
                regno="";
                cgpa=0;
        }
        public void get()
        {
```

```
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the values for name,regno and cgpa");
        name = sc.nextLine();
        regno = sc.next();
        cgpa = sc.nextDouble();
    }
    public void set()
    {
        System.out.println(name + "\t" +regno);
    }
    public double get_cgpa()
    {
        return(cgpa);
    }
    public static void main(String args[])
    {
        student A[] = new student[100];
        int n,i;
        System.out.println("Enter the value for n");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        for(i=0;i<n;i++)
        {
            A[i] = new student();
        }
        for(i=0;i<n;i++)
        {
            A[i].get();
        }
        System.out.println("The students with cgpa<4 are :");
        for(i=0;i<n;i++)
        {
            if(A[i].get_cgpa()<4)
            A[i].set();
        }
    }
}
```

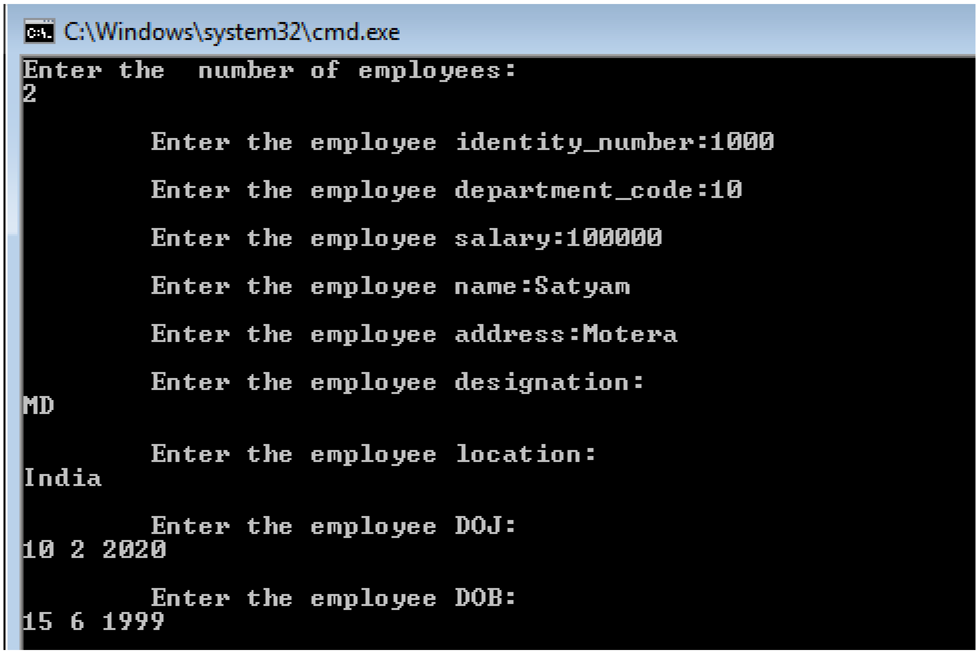


```
Enter the value for n

2
Enter the values for name,regno and cgpa
satyam
17bce0581
8.58
Enter the values for name,regno and cgpa
shaurya
99bce0000
10.0
The students with cgpa<4 are :
Press any key to continue . . .
```

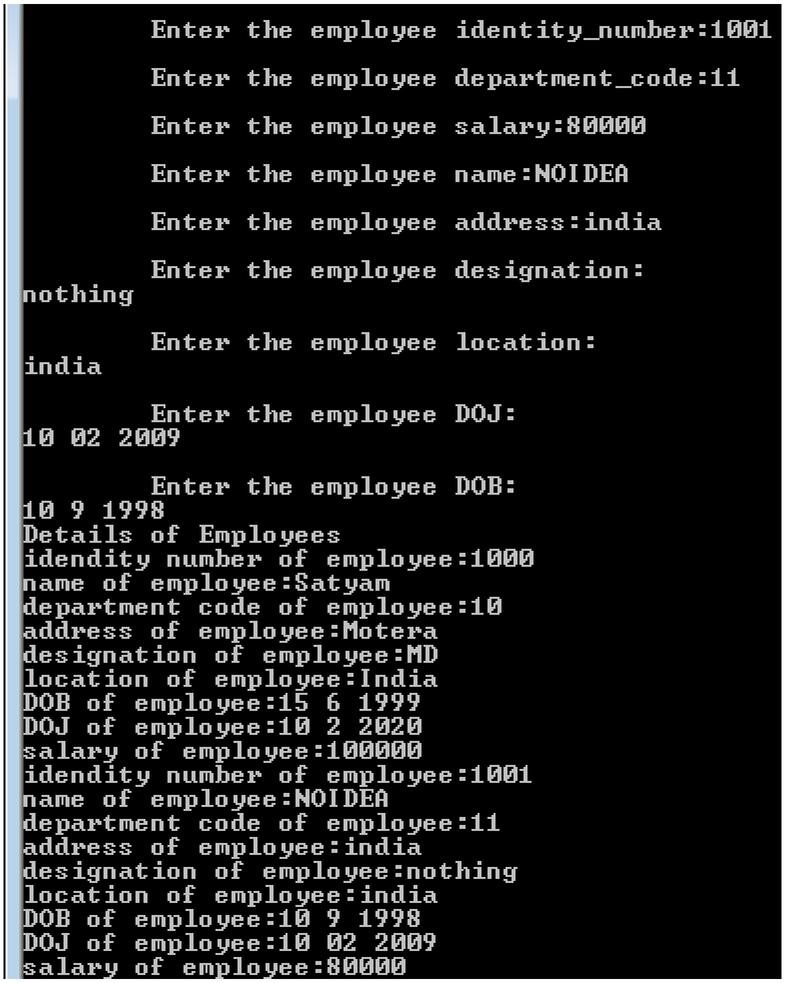
QUESTION 3

Assume you have a class 'Employee' with all basic information and a method to display its details. Using this class create new classes like 'Manager', 'Team Lead', 'Developer', and 'Trainee' with their own specific information. For the yearly appraisal they should earn 10 evaluation points in the last financial year. Add 15% of their salary to their current salary as an increment if they are eligible. Redefine the display method in the new classes to display updated details.

```
import java.io.*;
import java.util.*;

class Employee
 {
 int identity_number,department_code,salary;
 String name,address,designation,location,DOJ,DOB;
 void GetData()
 {
 Scanner sc=new Scanner(System.in);
 System.out.print("\n\tEnter the employee identity_number:"); identity_number=Integer.parseInt(sc.nextLine());
System.out.print("\n\tEnter the employee department_code:");
 department_code=Integer.parseInt(sc.nextLine());
 System.out.print("\n\tEnter the employee salary:"); salary=Integer.parseInt(sc.nextLine());
System.out.print("\n\tEnter the employee name:"); name=sc.nextLine();
System.out.print("\n\tEnter the employee address:"); address=sc.nextLine();
System.out.println("\n\tEnter the employee designation:");
designation=sc.nextLine();
System.out.println("\n\tEnter the employee location:");
location=sc.nextLine();
System.out.println("\n\tEnter the employee DOJ:"); DOJ=sc.nextLine();
 System.out.println("\n\tEnter the employee DOB:");DOB=sc.nextLine();
}
void PutData()
{
System.out.println("idendity number of employee:"+identity_number);
System.out.println("name of employee:"+name);
System.out.println("department code of employee:"+department_code);
System.out.println("address of employee:"+address);
System.out.println("designation of employee:"+designation);
System.out.println("location of employee:"+location);
System.out.println("DOB of employee:"+DOB);
System.out.println("DOJ of employee:"+DOJ);
System.out.println("salary of employee:"+salary);

}

public static void main(String args[])

{
int n,sum=0;
Scanner s=new Scanner(System.in);
 System.out.println("Enter the  number of employees:");

n=s.nextInt();
Employee[] Emp=new Employee[100];
 for(int i=0;i<n;i++)
{
 Emp[i]=new Employee();
}
for(int i=0;i<n;i++)
 {
 Emp[i].GetData();
}
System.out.println("Details of Employees");

for(int i=0;i<(n+1);i++)
{
Emp[i].PutData();}

}}
```



```
Enter the  number of employees:
2

        Enter the employee identity_number:1000

        Enter the employee department_code:10

        Enter the employee salary:100000

        Enter the employee name:Satyam

        Enter the employee address:Motera

        Enter the employee designation:
MD

        Enter the employee location:
India

        Enter the employee DOJ:
10 2 2020

        Enter the employee DOB:
15 6 1999
```



```
        Enter the employee identity_number:1001

        Enter the employee department_code:11

        Enter the employee salary:80000

        Enter the employee name:NOIDEA

        Enter the employee address:india

        Enter the employee designation:
nothing

        Enter the employee location:
india

        Enter the employee DOJ:
10 02 2009

        Enter the employee DOB:
10 9 1998
Details of Employees
idendity number of employee:1000
name of employee:Satyam
department code of employee:10
address of employee:Motera
designation of employee:MD
location of employee:India
DOB of employee:15 6 1999
DOJ of employee:10 2 2020
salary of employee:100000
idendity number of employee:1001
name of employee:NOIDEA
department code of employee:11
address of employee:india
designation of employee:nothing
location of employee:india
DOB of employee:10 9 1998
DOJ of employee:10 02 2009
salary of employee:80000
```

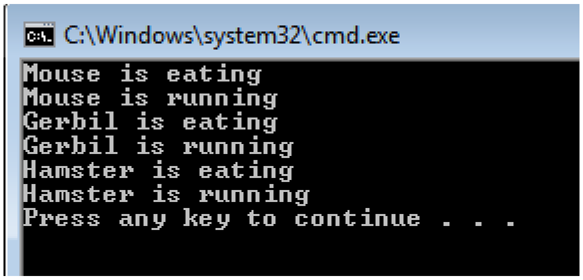
QUESTION 4

Create an inheritance hierarchy of Rodent, Mouse, Rat, Gerbil, and Hamster in Java. In the base class provide methods that are common to all Rodents i.e lifetime(), food(), abitat(), and uses() Override these methods in the derived classes to perform different behaviours, depending on the specific type of Rodent. Create an array of Rodent, fill it with different specific types of Rodents and call your base class methods.

[Hint: Rodent - a gnawing mammal of an order that includes rats, mice, squirrels, hamsters, porcupines, and their relatives, distinguished by strong constantly growing incisors and no canine teeth. They constitute the largest order of mammals.]

```
class Rodent
{
        void eat()
        {
                System.out.println("Rodent is eating");
        }
        void run()
        {
                System.out.println("Rodent is running");
        }
}
class Mouse extends Rodent
{
        void eat()
        {
                System.out.println("Mouse is eating");
        }
        void run()
        {
                System.out.println("Mouse is running");
        }
}
class Gerbil extends Rodent
{
        void eat()
        {
                System.out.println("Gerbil is eating");
        }
        void run()
        {
                System.out.println("Gerbil is running");
        }
}
```

```
class Hamster extends Rodent
{
    void eat()
    {
        System.out.println("Hamster is eating");
    }
    void run()
    {
        System.out.println("Hamster is running");
    }
}
class RodentsDemo
{
    public static void main(String args[])
    {
        Rodent r[] = new Rodent[3];
        r[0] = new Mouse();
        r[1] = new Gerbil();
        r[2] = new Hamster();

        r[0].eat();
        r[0].run();

        r[1].eat();
        r[1].run();

        r[2].eat();
        r[2].run();
    }
}
```



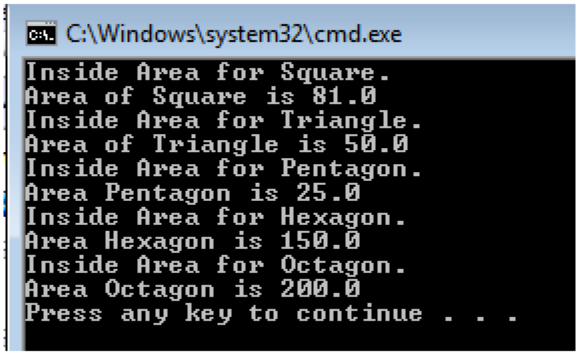
```
Mouse is eating
Mouse is running
Gerbil is eating
Gerbil is running
Hamster is eating
Hamster is running
Press any key to continue . . .
```

QUESTION 5

Write a Java program to create an abstract class named Shape that contains empty methods named numberOfSides( ) and calculateArea(). Derive classes named Triangle, Square, Pentagon, Hexagon, and Octagon from Shape. Each one of the derived classes contains the method definitions that shows the number of sides and the area calculations in the given geometrical figures.

[Hint: Use Runtime polymorphism and refer the following images for area calculations]

```
abstract class Figure
{
 double dim1;
 double dim2;
 Figure(double a)
{
 dim1 = a;
}

Figure(double a, double b)
{
		dim1 = a;
		dim2 = b;
}

 abstract double area();
}
class Square extends Figure
{
 Square(double a) {
 super(a);
 }
 double area()
 {
 System.out.println("Inside Area for Square.");
 return dim1 * dim1;
 }
}

class Triangle extends Figure
{
 Triangle(double a, double b)
 {
 super(a, b);
 }
 double area()
 {
 System.out.println("Inside Area for Triangle.");
 return dim1 * dim2 / 2;
 }
}

class Pentagon extends Figure
{
 Pentagon(double a, double b)
 {
 super(a, b);
 }
 double area()
 {
 System.out.println("Inside Area for Pentagon.");
 return dim1 * dim2 / 2;
 }
}

class Hexagon extends Figure
{
 Hexagon(double a, double b)
 {
 super(a, b);
 }
 double area()
 {
 System.out.println("Inside Area for Hexagon.");
 return dim1 * dim2 * 3;
 }
}

class Octagon extends Figure
{
 Octagon(double a, double b)
 {
 super(a, b);
 }
 double area()
 {
 System.out.println("Inside Area for Octagon.");
 return dim1 * dim2 * 4;
 }
}
class AbstractAreas
{
 public static void main(String args[])
{
 Square r = new Square(9);
 Triangle t = new Triangle(10, 10);
 Pentagon u = new Pentagon(10, 5);
 Hexagon v = new Hexagon(10, 5);
 Octagon w = new Octagon(10, 5);
 Figure figref;
 figref = r;
 System.out.println("Area of Square is " + figref.area());
 figref = t;
 System.out.println("Area of Triangle is " + figref.area());
 figref = u;
 System.out.println("Area Pentagon is " + figref.area());
 figref = v;
 System.out.println("Area Hexagon is " + figref.area());
 figref = w;
 System.out.println("Area Octagon is " + figref.area());
 }
}
```



```
Inside Area for Square.
Area of Square is 81.0
Inside Area for Triangle.
Area of Triangle is 50.0
Inside Area for Pentagon.
Area Pentagon is 25.0
Inside Area for Hexagon.
Area Hexagon is 150.0
Inside Area for Octagon.
Area Octagon is 200.0
Press any key to continue . . .
```