JAVA ASSIGNMENT - 3

SESSION 2021-2022

LAB REPORT SUBMITTED

By:

Vivek Kumar Choudhary

(20204234)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY ALLAHABAD

PRAYAGRAJ, INDIA- 211004

1. Write a class named Employee that has the following fields:

- **name. The name field references a String object that holds the employee’s name.**
- **idNumber. The idNumber is an int variable that holds the employee’s ID number.**
- **department. The department field references a String object that holds the name of the department where the employee works.**
- **position. The position field references a String object that holds the employee’s job title.**

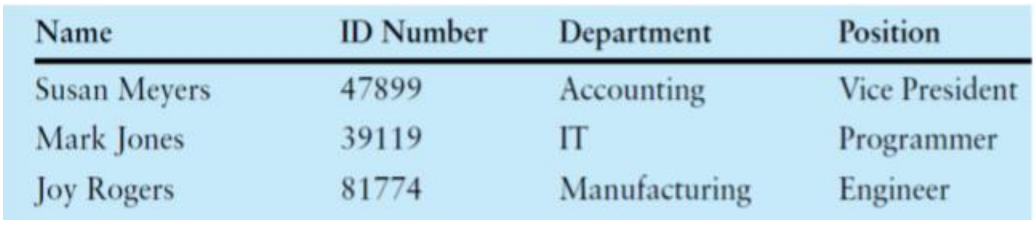
The class should have the following constructors:

- **A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee’s name, employee’s ID number, department, and position.**
- **A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee’s name and ID number. The department and position fields should be assigned an empty string ().**
- **A no-arg constructor that assigns empty strings () to the name, department, and position fields, and 0 to the idNumber field.**

Write appropriate mutator methods that store values in these fields and accessor methods

that return the values in these fields. Once you have written the class, write a separate

program that creates three Employee objects to hold the following data:



Name	ID Number	Department	Position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

The program should store this data in the three objects and then display the data for each employee on the screen.

CLASS CODE:-

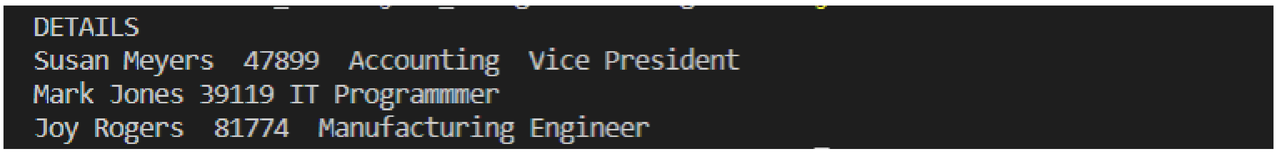
```
public class Employee{
    private String name;
    private int idNumber;
    private String department;
    private String position;

    Employee(String n,int id, String dept, String pos){
         name=n;
         idNumber=id;
         department=dept;
         position=pos;
    }
    // constructor with two arguments
    Employee(String n,int id){
        name=n;
        idNumber=id;
        department="";
        position="";
    }
// no args constructor
    Employee(){
        name="";
        idNumber=0;
        department="";
        position="";
```

```
}

// Accessors(getter) are
public String getName(){
    return name;
}
public int getId(){
    return idNumber;
}
public String getDept(){
    return department;
}
public String getPos(){
    return position;
}
// Mutators (setters)
public void setName(String name){
   this.name=name;
}
public void setId(int idNumber){
    this.idNumber=idNumber;
}
public void setDept(String department){
    this.department=department;
}
public void setPos(String position){
        this.position=position;
    }
}
```

MAIN CODE(Driver Code)

```
public class Main {
    public static void main(String[] args) {
        Employee emp1=new Employee();
        // using mutators to set the different data
        emp1.setName("Susan Meyers");
        emp1.setId(47899);
        emp1.setDept("Accounting");
        emp1.setPos("Vice President");
System.out.println("DETAILS ");

        // using accessors to print data
      System.out.print(emp1.getName()+"  ");
      System.out.print(emp1.getId()+"  ");
      System.out.print(emp1.getDept()+"  ");
      System.out.println(emp1.getPos()+"  ");

      Employee emp2=new Employee("Mark
Jones",39119,"IT","Programmmer");
      System.out.print(emp2.getName()+" ");
      System.out.print(emp2.getId()+" ");
      System.out.print(emp2.getDept()+" ");
      System.out.println(emp2.getPos()+" ");
    Employee emp3=new Employee("Joy
Rogers",81774,"Manufacturing","Engineer");


    System.out.print(emp3.getName()+"  "+emp3.getId()+"
"+emp3.getDept()+" "+emp3.getPos()+" ");


  }

}
```

OUTPUT



```
DETAILS
Susan Meyers  47899  Accounting  Vice President
Mark Jones 39119 IT Programmmer
Joy Rogers  81774  Manufacturing Engineer
```

2. Write a class named Car that has the following fields:

• yearModel. The yearModel field is an int that holds the car's year model.

• make. The make field references a String object that holds the make of the car.

• speed. The speed field is an int that holds the car's current speed.

In addition, the class should have the following constructor and other methods.

• Constructor. The constructor should accept the car's year model and make as arguments. These values should be assigned to the object's yearModel and make fields.The constructor should also assign 0 to the speed field.

• Accessors. Appropriate accessor methods should get the values stored in an object's yearModel, make, and speed fields.

• accelerate. The accelerate method should add 5 to the speed field each time it iscalled.

• brake. The brake method should subtract 5 from the speed field each time it is called.

Demonstrate the class in a program that creates a Car object, and then calls the accelerate method five times. After each call to the accelerate method, get the current speed of the car and display it. Then call the brake method five times. After each call to the brake method, get the current speed of the car and display it.

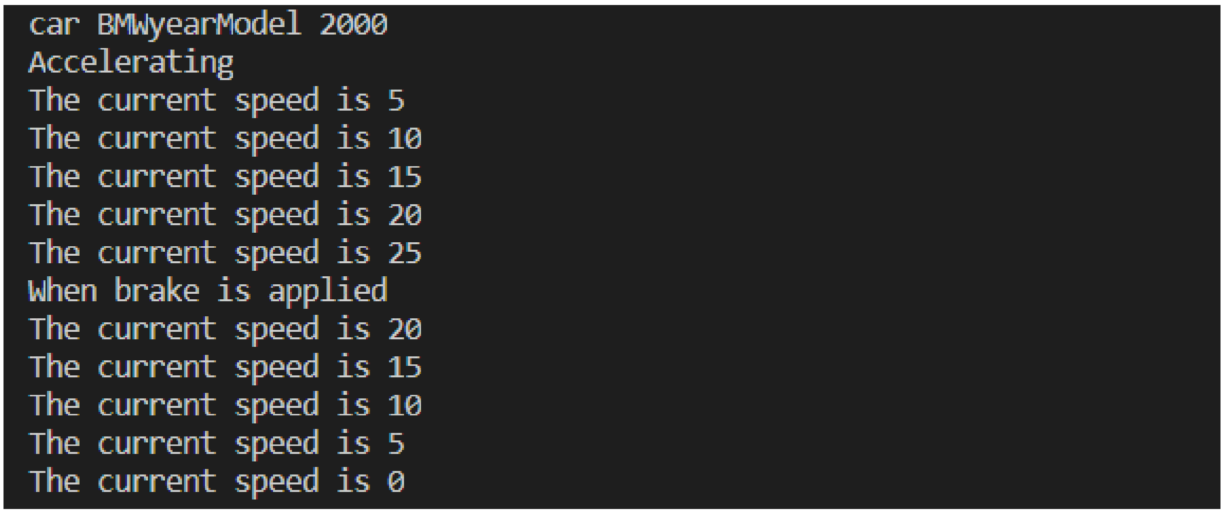
CODE:-

```
class Car{
  private int yearModel;
  private  String make;
  private  int speed;
   Car(int yearModel,String make){
      this.yearModel=yearModel;
      this.make=make;
      this.speed=0;
   }
   public void accelerate(){
        this.speed=this.speed+5;
         }

   public void brake(){
      speed=speed-5;
```

```
}
// Accessors
public int getSpeed(){
  return speed;
}
public int getYearModel(){
  return yearModel;
}
public String getmake(){
  return make;
}
}
public class Ques2 {
  public static void main(String[] args) {
     // creating car object
     Car car1=new Car(2000,"BMW");
     System.out.println("car "+car1.getmake()+" yearModel "+car1.getYearMod
el());
     System.out.println("Accelerating ");
     for(int i=0;i<5;i++){
     car1.accelerate();
     System.out.println("The current speed is "+car1.getSpeed());
     }
     System.out.println("When brake is applied");


     for(int i=0;i<5;i++){
       car1.brake();
```

```
        System.out.println("The current speed is "+car1.getSpeed());
        }
  }
}
```

OUTPUT



```
car BMWyearModel 2000
Accelerating
The current speed is 5
The current speed is 10
The current speed is 15
The current speed is 20
The current speed is 25
When brake is applied
The current speed is 20
The current speed is 15
The current speed is 10
The current speed is 5
The current speed is 0
```

3. Design a class PersonalInformation that holds the following personal data: name, address, age, and phone number. Write appropriate accessor and mutator methods. Demonstrate the class by writing a program that creates three instances of it. One instance should hold your information, and the other two should hold your friends' or family members' information.

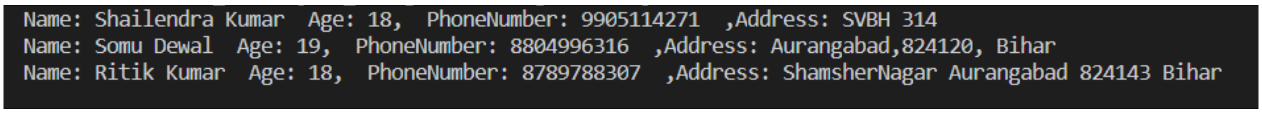
CODE:-

```
 class PersonalInformation {
    private String name,address,phoneNumber;
    private int age;
    // accessors
```

```
public String getName(){
return name;
}
public String getAddress(){
return address;
}
public int getAge(){
return age;
}
public String getPhoneNumber(){
return phoneNumber;
}
// mutators
public void setName(String name){
this.name=name;
}
public void setAddress(String address){
this.address=address;
}
public void setAge(int age){
this.age=age;
}
public void setPhoneNumber(String phoneNumber){
this.phoneNumber=phoneNumber;
}
//constructor
```

```
     PersonalInformation(String name,int age,String phoneNumber,String addres
s){
    this.name=name;
    this.address=address;
    this.age=age;
    this.phoneNumber=phoneNumber;
     }
     public String toString() {
        return "Name: " + this.name + "  Age: "
              + this.age+ ",  PhoneNumber: " + this.phoneNumber+"  ,Address: "+thi
s.address ;
    }
    }



public class Ques3 {
    public static void main(String[] args) {
       PersonalInformation self=new PersonalInformation("Shailendra Kumar", 18
, "9905114271", "SVBH 314");
       PersonalInformation friendSomu=new PersonalInformation("Somu Dewal",
 19, "8804996316", "Aurangabad,824120, Bihar");
     PersonalInformation friendRitik=new PersonalInformation("Ritik Kumar", 18
, "8789788307", "ShamsherNagar Aurangabad 824143 Bihar");
        System.out.println(self.toString());
        System.out.println(friendSomu.toString());
        System.out.println(friendRitik.toString());
    }
}
```

OUTPUT

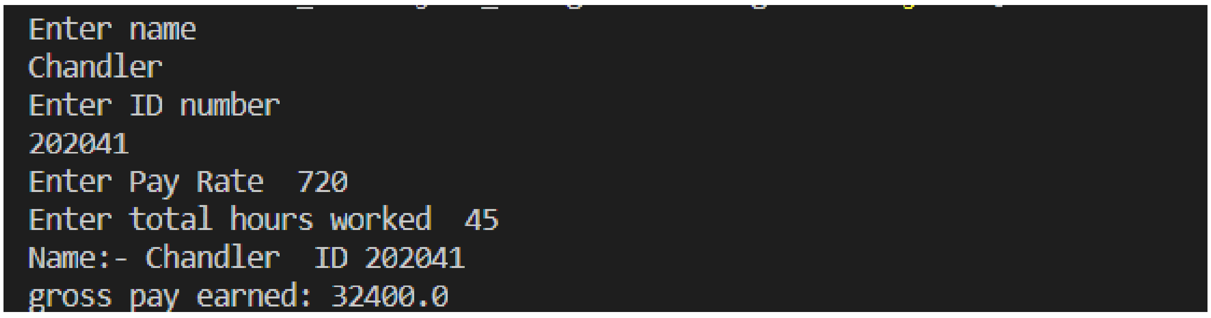

```
Name: Shailendra Kumar  Age: 18,  PhoneNumber: 9905114271  ,Address: SVBH 314
Name: Somu Dewal  Age: 19,  PhoneNumber: 8804996316  ,Address: Aurangabad,824120, Bihar
Name: Ritik Kumar  Age: 18,  PhoneNumber: 8789788307  ,Address: ShamsherNagar Aurangabad 824143 Bihar
```

4. Design a Payroll class that has fields for an employee's name, ID number, hourly pay rate, and number of hours worked. Write the appropriate accessor and mutator methods and a constructor that accepts the employee's name and ID number as arguments. The class should also have a method that returns the employee's gross pay, which is calculated as the number of hours worked multiplied by the hourly pay rate. Write a program that demonstrates the class by creating a Payroll object, then asking the user to enter the data for an employee. The program should display the amount of gross pay earned.

```
import java.util.Scanner;
class Payroll{

    private String name;
    private int idNumber;
  private float payRate,hoursWorked;

    Payroll(String name,int idNumber){
        this.name=name;
        this.idNumber=idNumber;

    }
    // Accessors(getter) are
    public String getName(){
```

```
    return name;
}
public int getId(){
    return idNumber;
}
public float getpayRate(){
    return payRate;
}
public float gethoursWorked(){
    return hoursWorked;
}

// Mutators (setters)
public void setName(String name){
   this.name=name;
}
public void setId(int idNumber){
    this.idNumber=idNumber;
}
public void setpayRate(float payRate){
    this.payRate=payRate;
}
public void sethoursWorked(float hoursWorked){
    this.hoursWorked=hoursWorked;
}
public float income(){
```

```java
            return hoursWorked*payRate;
        }
    }


public class Ques4 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter name ");
        String a=sc.next();
        System.out.println("Enter ID number ");
        int id= sc.nextInt();
    Payroll emp1=new Payroll(a,id);
         System.out.print("Enter Pay Rate  ");
        emp1.setpayRate(sc.nextFloat());
        System.out.print("Enter total hours worked  ");
        emp1.sethoursWorked(sc.nextFloat());
       float grossPay= emp1.income();
       System.out.println("Name:- "+emp1.getName()+"  ID "+emp1.getId());
        System.out.println("gross pay earned: "+grossPay);
    } }
```



```
Enter name
Chandler
Enter ID number
202041
Enter Pay Rate  720
Enter total hours worked  45
Name:- Chandler  ID 202041
gross pay earned: 32400.0
```

5. Design a TestScores class that has fields to hold three test scores. The class should have a constructor, accessor and mutator methods for the test score fields, and a method that returns the average of the test scores. Demonstrate the class by writing a separate program that creates an instance of the class. The program should ask the user to enter three test scores, which are stored in the TestScores object. Then the program should display the average of the scores, as reported by the TestScores object

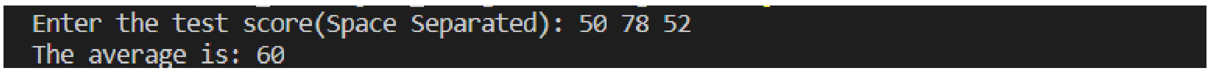
CLASS CODE:-

```
public class TestScore {
    private int m1,m2,m3;
    public TestScore(int m1,int m2,int m3){
    this.m1=m1;
    this.m2=m2;
    this.m3=m3;
    }
    public TestScore(){

    }
    public int getM1() {
    return m1;
    }
    public int getM2() {
    return m2;
    }
    public int getM3() {
    return m3;
```

```
	}
	public void setM1(int m1) {
	this.m1 = m1;
	}
	public void setM2(int m2) {
	this.m2 = m2;
	}
	public void setM3(int m3) {
	this.m3 = m3;
	}
	public int average(){
	return (this.m1+this.m2+this.m3)/3;
	}
       }
```

MAIN CODE

```
import java.util.Scanner;
public class Ques5 {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter the test score(Space Separated): ");
 TestScore s = new TestScore();
 s.setM1(sc.nextInt());
 s.setM2(sc.nextInt());
 s.setM3(sc.nextInt());
 System.out.println("The average is: "+s.average());
 sc.close();
 }
}
```



```
Enter the test score(Space Separated): 50 78 52
The average is: 60
```

6. Write a Circle class that has the following fields:

• radius: a double

• PI: a final double initialized with the value 3.14159

The class should have the following methods:

• Constructor. Accepts the radius of the circle as an argument.

• Constructor. A no-arg constructor that sets the radius field to 0.0.

• setRadius. A mutator method for the radius field.

• getRadius. An accessor method for the radius field.

• getArea. Returns the area of the circle, which is calculated as area = PI * radius *radius

• getDiameter. Returns the diameter of the circle, which is calculated as diameter =radius * 2

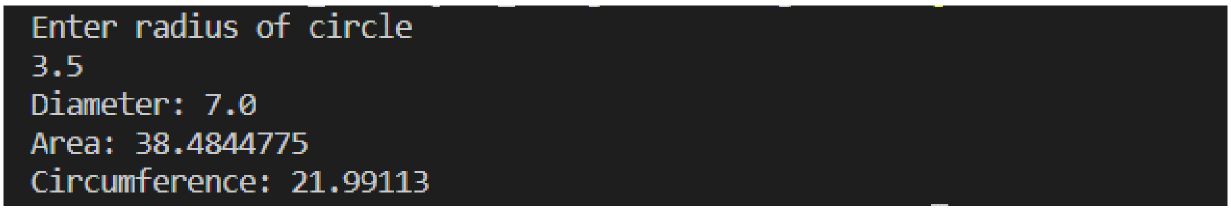
• getCircumference. Returns the circumference of the circle, which is calculated ascircumference = 2 * PI * radius

Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

CODE:-

```
import java.util.Scanner;
class Circle{
  private double radius;
  double PI=3.14159;
   public double getRadius() {
      return radius;
   }
   public void setRadius(double radius) {
      this.radius = radius;
   }
   Circle(){
      this.radius=0.0;
   }
   Circle(double radius){
      this.radius=radius;
   }
   public double getArea(){
      return radius*radius*PI;
   }
   public double getDiameter(){
      return radius*2;
   }
   public double getCircumference(){
      return radius*2*PI;
   }
}

public class Ques6 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter radius of circle ");
        double r=sc.nextDouble();
        Circle c1=new Circle(r);
        System.out.println("Diameter: "+c1.getDiameter());
        System.out.println("Area: "+c1.getArea());
        System.out.println("Circumference: "+c1.getCircumference());
    }
}
```

OUTPUT



```
Enter radius of circle
3.5
Diameter: 7.0
Area: 38.4844775
Circumference: 21.99113
```

7. Write a class named MonthDays. The class's constructor should accept two arguments:

- **An integer for the month (1 5 January, 2 February, etc.).**
- **An integer for the year**

The class should have a method named getNumberOfDays that returns the number of days in the specified month. The method should use the following criteria to identify leap years:

(a) Determine whether the year is divisible by 100. If it is, then it is a leap year if and if only it is divisible by 400. For example, 2000 is a leap year but 2100 is not.

(b) If the year is not divisible by 100, then it is a leap year if and if only it is divisible by 4. For example, 2008 is a leap year but 2009 is not.

Demonstrate the class in a program that asks the user to enter the month (letting the user enter an integer in the range of 1 through 12) and the year. The program should then display the number of days in that month.

Here is a sample run of the program:

Enter a month (1-12): 2 [Enter]

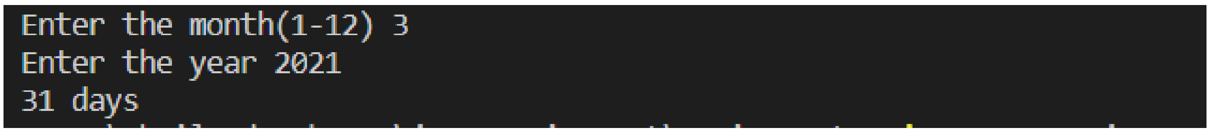
Enter a year: 2008 [Enter]

29 days

CODE:-

```
import java.util.Scanner;
class MonthDays{
   private int month,year;

MonthDays(int month, int year){
   this.month=month;
   this.year=year;
}
public void SetMonth(int month){
    this.month=month;
}
public int getMonth(){
    return month;
}
public void SetYear(int year){
    this.year=year;
}
public int getyear(){
    return year;
}
public int getNumberOfDays(){
    if(year%100==0&&year%400==0){
        if(month==2){
            return 29;
        }
    }
    else if(year%100!=0&&year%4==0){
        if(month==2){
            return 29;
        }
    }else{
        if(month==2){
            return 28;
        }
    }
    if(month==6||month==4||month==9||month==11){
        return 30;
    }
    if(month==1||month==3||month==5||month==7||month==8||month==10
||month==12){
        return 31;
    }
    return 0;
}

}

public class Ques7 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the month(1-12) ");
        int m=sc.nextInt();
        System.out.print("Enter the year ");
        int y=sc.nextInt();
        MonthDays days= new MonthDays(m,y);
        int day=days.getNumberOfDays();
        System.out.println(day+" days");

    }
}
```

OUTPUT



```
Enter the month(1-12) 3
Enter the year 2021
31 days
```