Juan Casanova
Lab6

Problem 9.1

```
import java.text.DecimalFormat;

public class Q9_1 {

	public static void main(String[] args) {

		Rectangle r1 = new Rectangle(4, 40);
		Rectangle r2 = new Rectangle(3.5, 35.9);
		DecimalFormat ab = new DecimalFormat("##.##");

		// These is for the first Rectangle
		System.out.println("The Details of the first rectangle");
		System.out.println("Width of Rectangle1 is: " + r1.width );
		System.out.println("Height of Rectangle1 is: " + r1.height );
		System.out.println("Area of Rectangle1 is: " + ab.format(r1.getArea()));
		System.out.println("Perimeter of Rectangle1 is: " + r1.getPerimeter());
		System.out.println();

		//These is for the second Rectangle
		System.out.println("The Details of the Second rectangle");
		System.out.println("Width of Rectangle2 is: " + r2.width );
		System.out.println("Height of Rectangle2 is: " + r2.height );
		System.out.println("Area of Rectangle2 is: " + ab.format(r2.getArea()));
		System.out.println("Perimeter of Rectangle2 is: " + r2.getPerimeter());
 }

}

public class Rectangle {
		double width = 1.0;
		double height = 1.0;

		Rectangle() {
		}
		Rectangle(double newWidth, double newHeight) {
			width = newWidth;
			height = newHeight;
		}
```

```
	double getArea() {
		return width * height;
	}

	double getPerimeter() {
		return 2 * (width + height);
	}


}
```

Problem 9.5

```
import java.util.Calendar;
import java.util.GregorianCalendar;

public class Q9_5 {

	public static void main(String[] args) {
		Calendar calendar = new GregorianCalendar();
		System.out.println("Year: " + calendar.get(Calendar.YEAR));
		System.out.println("Month: " + calendar.get(Calendar.MONTH));
		System.out.println("Day of The Month: " +
calendar.get(Calendar.DAY_OF_MONTH));

		calendar.setTimeInMillis(123456789765L);
		System.out.println("Year: " + calendar.get(Calendar.YEAR));
		System.out.println("Month: " + calendar.get(Calendar.MONTH));
		System.out.println("Day of The Month: " +
calendar.get(Calendar.DAY_OF_MONTH));

	}

}
```

Problem 9.7

```
import java.util.Date;

class Account {

		private int id;
		private double balance;
```

```
        private double annualInterestRate;
        private Date dateCreated;

Account () {
        id = 0;
        balance = 0;
        annualInterestRate = 0;
        dateCreated = new Date();
}

Account(int ID, double BALANCE) {
        id = ID;
        balance = BALANCE;
        dateCreated = new Date();
}

public int getID() {
        return id;
}
public double getBalance() {
        return balance;
}
public double getAnnualInterestRate() {
        return  annualInterestRate;
}

public void setId(int id) {
        this.id = id;
}
public void setBalance(double balance) {
        this.balance = balance;
}

public void setAnnualInterestRate(double annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
}

public Date getDateCreated() {
        return dateCreated;
}

public double getMonthlyInterestRate() {
        return annualInterestRate/12.0;
```

```
	}

	public void withdraw(double amount) {
		balance = balance - amount;
	}

	public void deposit(int i) {

	}
 }

import java.text.DecimalFormat;

public class test {

	public static void main(String[] args) {
		DecimalFormat cd = new DecimalFormat("##.##");
		Account account = new Account(1122, 20000);
		account.setAnnualInterestRate(4.5);
		account.withdraw(2500);
		account.deposit(3000);

		System.out.println("Balance is: " + cd.format(account.getBalance()));
		System.out.println("Monthly interest is: " +
cd.format(account.getMonthlyInterestRate()));
		System.out.println("Account created date: " +
account.getDateCreated().toString());


	}

}
```

Problem 9.11

```
	class LinearEquation {
		private double a;
		private double b;
		private double c;
		private double d;
		private double e;
		private double f;
```

```
LinearEquation( double A, double B, double C,
                double D, double E, double F) {
        a = A;
        b = B;
        c = C;
        d = D;
        e = E;
        f = F;
}
public double getA() {
        return a;
}
public double getB() {
        return b;
}
public double getC() {
        return c;
}
public double getD() {
        return d;
}
public double getE() {
        return e;
}
public double getF() {
        return f;
}

public boolean isSolvable() {
        if(((a * d) - (b * c)) == 0)
                return false;
        else
                return true;
}
public double getX() {
        double numerator  = ((e * d) -(b * f));
        double denominator = ((a * d) - (b * c));
        return numerator/denominator;
}
public double getY() {
        double numerator  = ((a * f) -(e * c));
        double denominator = ((a * d) - (b * c));
        return numerator/denominator;
```

```
	}


}

import java.text.DecimalFormat;
import java.util.Scanner;

public class test2 {

	public static void main(String[] args) {

		double a = 0.0;
		double b = 0.0;
		double c = 0.0;
		double d = 0.0;
		double e = 0.0;
		double f = 0.0;

		DecimalFormat gh = new DecimalFormat("##.##");

		Scanner input = new Scanner(System.in);

		System.out.print("Enter the value of a: ");
		String num = input.next();
		a = Double.parseDouble(num);

		System.out.print("Enter the value of b: ");
		String num1 = input.next();
		b = Double.parseDouble(num1);

		System.out.print("Enter the value of c: ");
		String num11 = input.next();
		c = Double.parseDouble(num11);

		System.out.print("Enter the value of d: ");
		String num111 = input.next();
		d = Double.parseDouble(num111);

		System.out.print("Enter the value of e: ");
		String num1111 = input.next();
		e = Double.parseDouble(num1111);
```

```
                System.out.print("Enter the value of f: ");
                String num11111 = input.next();
                f = Double.parseDouble(num11111);

                LinearEquation ij = new LinearEquation(a,b,c,d,e,f);
                 if(ij.isSolvable()) {
                        System.out.println("X = " + gh.format(ij.getX()));
                        System.out.println("Y = " + gh.format(ij.getY()));
                 } else {
                        System.out.println("The equation has no solution.");
                 }

        }

}
```

Problem 9.13

```
import java.text.DecimalFormat;
import java.util.Scanner;

public class Location {
        static double[][] myArray = null;
        public static int row = 0;
        public static int column = 0;
        public static double maxValue = 0;


        public static void main(String[] args) {
                System.out.print("Enter the number of rows and columns of the array: " );

                Scanner input = new Scanner(System.in);
                String sizeString = null;
                String arrayElements = null;
                String[] temp1 = null;
                String[] temp2 = null;

                sizeString = input.nextLine();
                temp1 = sizeString.split(" ");
                if(temp1.length != 2) {
                        System.out.println("Either Row or Column value not entered. Program
will exit. ");
```

```
                    System.exit(0);
                }
                for(int i=0; i<temp1.length; i++) {
                    if(Integer.parseInt(temp1[i]) <= 0 ) {
                        System.out.println("Negative Row or Column value entered.
Program will exit. ");
                        System.exit(0);
                    }
                }

                row = Integer.parseInt(temp1[0]);
                column = Integer.parseInt(temp1[1]);
                myArray = new double [row][column];

                System.out.println("Enter the array: ");
                for(int i=0; i<row; i++) {
                    arrayElements = input.nextLine();
                    temp2 = arrayElements.split(" ");
                    if(temp2.length != column) {
                        System.out.println("Invalid number of entry in the array. Program
will exit. ");
                        System.exit(0);
                    }
                    for(int j=0; j<column; j++) {
                        myArray[i][j] = Double.parseDouble(temp2[j]);
                    }
                    arrayElements = null;
                }
                locateLargest(myArray);
        }

        public static Location locateLargest(double[][] a) {
                DecimalFormat kl = new DecimalFormat("##.##");
                maxValue = a[1][1];
                int posI = 0;
                int posJ = 0;
                for(int i=0; i<row; i++) {
                    for(int j=0; j<row; j++) {
                        if(a[i][j] > maxValue) {
                            maxValue = a[i][j];
                            posI = i;
                            posJ = j;
                        }
```

```
                    }
                }
                System.out.println("The location of the largest element is: " +
kl.format(maxValue) + " at" +
                " ("+posI +"," + posJ +") ");
                return new Location();
        }

}
```