German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Mohammed Abdel Megeed

Introduction to Computer Programming, Spring Term 2018
Practice Assignment 2

Discussion 24.2.2018 - 1.3.2018

Exercise 2-1 To be discussed in the tutorial
Code Refactoring

Refactor each of the following program fragments (if possible)

- Program 1

```
if (x > y)
    System.out.println("Good Day");
else {
    System.out.println("x is less than or equal to y");
    System.out.println("Good Day");
}
```

Solution:

We can first `extract to back`, thus we get:

```
if (x > y){
}
else {
    System.out.println("x is less than or equal to y");
}
System.out.println("Good Day");
```

Then, we can apply `swap branches` to make the code look nicer, we get:

```
if (!(x > y)) { //negating the condition
    System.out.println("x is less than or equal to y");
}
else {
}
System.out.println("Good Day");
```

We cannot do any more refactoring steps. However, it can be simplified to the following program:

```
if (x <= y) {
    System.out.println("x is less than or equal to y");
}
System.out.println("Good Day");
```

- Program 2

```
if (x == 0) {
    System.out.println("x is even");
    System.out.println(x);
}
if (x%2 == 0) {
    System.out.println("x is even");
    System.out.println(x);
}
if (x == 1) {
    System.out.println("x is odd");
    System.out.println(x);
}
if (x%2 != 0) {
    System.out.println("x is odd");
    System.out.println(x);
}
```

Solution:

The program cannot be refactored because :

a) We cannot apply `swap branches` since we do not have if-else statements.

b) We cannot apply `remove redundant tests` because there is no condition repeated twice.

c) We cannot apply `extract to back` or `extract to front` because there are no if-else statements.

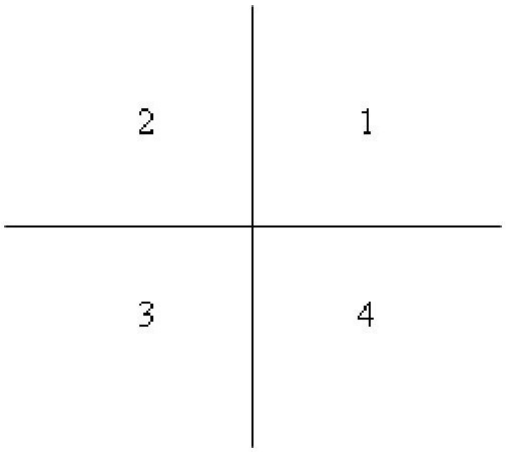
However, the program can be simplified to:

```
if (x%2 == 0) {
    System.out.println("x is even");
}
else {
    System.out.println("x is odd");
}
System.out.println(x);
```

Exercise 2-2

Cartesian Plane

Write a Java program that reads the x-y coordinates of a point in the Cartesian plane and displays a message telling the user the quadrant in which the point is located or the axis on which the point lies.



Solution:

```
import java.util.Scanner;

public class CartesainPlane {

    public static void main(String [] args) {

        Scanner sc = new Scanner(System.in);
        double x,y;

        System.out.print("Please enter the x-coordinate of the point ");
        x = sc.nextInt();
        System.out.print("Please enter the y-coordinate of the point ");
        y = sc.nextInt();

        if ((x == 0) && (y == 0)) {
            System.out.println("The point is the origin");
        }
        else if (x == 0) {
                System.out.println("The point lies on the y-axis");
            }
            else if (y == 0) {
                System.out.println("The point lies on the x-axis");
                }
                else if (x < 0) {
                if (y > 0 ) {
                    System.out.println("The point lies in the second quadrant");
                }
                else {
                    System.out.println("The point lies in the third quadrant");
                }
            }
        else {
            if (y > 0 ) {
                System.out.println("The point lies in the first quadrant");
            }
```

```
            else {
                System.out.println("The point lies in the forth quadrant");
            }
        }
    }
}
```

Exercise 2-3

Score

A sequence of six tests, all scored out of 100, are to be given different weightings in determining a final mark. Write a Java program that computes the appropriate weighted score for one test. The fragment should first read values of testNumber and score. Using a switch statement, it should then compute and print the appropriate value of weightedScore using the weightings given in the following table.

Test Number	Weight
1	10%
2	20 %
3	20 %
4	15%
5	15%
6	20%

Solution:

```
import java.util.Scanner;

public class Score {

    public static void main(String [] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("please enter the test number and the score");
        int testNumber = sc.nextInt();
        int score = sc.nextInt();
        double weight = 0;
        switch (testNumber) {
            case 1:
            weight = 0.1 * score;
            break;
            case 2:
            case 3:
            case 6:
            weight = 0.2 * score;
            break;
            case 4:
            case 5:
            weight = 0.15 * score;
            break;
            default:
            weight = 0;
        }
        System.out.println("A score of " + score+" on test "+testNumber+" gives a
        weighted score of "+weight +".");
    }
```

```
}
```

Exercise 2-4 To be discussed in the Tutorial
Maximum

Write a Java program to calculate the maximum of three numbers. **Solve using conditional operator only.**

Solution:

```
import java.util.Scanner;

public class Maximum {

    public static void main(String [] args) {

        int num1, num2, num3, max;
        Scanner sc = new Scanner(System.in);

        System.out.print("Please enter the first number: ");
        num1 = sc.nextInt();
        System.out.print("Please enter the second number: ");
        num2 = sc.nextInt();
        System.out.print("Please enter the third number: ");
        num3 = sc.nextInt();

        max =  num1 > num2  ? (num1 > num3 ? num1 : num3 )
                            : (num2 > num3 ? num2 : num3 );
        System.out.println("Maximum = " + max);
    }
}
```

Exercise 2-5 To be discussed in the lab
Months

Write a Java program that prints the number of days for any given month.

```
Please enter the month number (1-12): 5
31 days.
```

Solution:

```
import java.util.Scanner;

public class Month {

    public static void main(String [] args) {

        int month;
        String numberofdays;
        Scanner sc = new Scanner(System.in);
        System.out.print("Please enter the month number (1-12): ");
```

```
        month = sc.nextInt();
        switch (month) {
        case 1:
            /* falls through */
        case 3:
            /* falls through */
        case 5:
            /* falls through */
        case 7:
            /* falls through */
        case 8:
            /* falls through */
        case 10:
            /* falls through */
        case 12:
            numberofdays = "31 days.";
            break;
        case 4:
            /* falls through */
        case 6:
            /* falls through */
        case 9:
            /* falls through */
        case 11:
            numberofdays = "30 days.";
            break;
        case 2:
            numberofdays = "either 28 or 29 days.";
            break;
        default:
            numberofdays = "The value you entered for the month is not correct!";
            break;
        }
        System.out.println(numberofdays);
    }
}
```

Exercise 2-6

Quadratic Equation

Write a Java program that reads from the user three double numbers a, b, c representing the coefficients of a quadratic equation $ax^2+bx+c=0$. The program should calculate the roots of the quadratic equation using the following formulae:
$x_1 = \frac{-b+\sqrt{(b^2-4ac)}}{2a}$
$x_2 = \frac{-b-\sqrt{(b^2-4ac)}}{2a}$
If $a = 0$ or if $b^2 - 4ac < 0$ the output of the program should be: No Solutions !

Solution:

```
import java.util.Scanner;

public class Quadratic {

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

```
        double a,b,c,underroot;
        Scanner sc = new Scanner(System.in);

        System.out.println("Please enter the coefficient a");
        a = sc.nextDouble();
        System.out.println("Please enter the coefficient b");
        b = sc.nextDouble();
        System.out.println("Please enter the coefficient c");
        c = sc.nextDouble();
        underroot = Math.pow(b,2) - (4*a*c);

        if ((a == 0) || (underroot < 0)) {
            System.out.println("No Solution !");
        }
        else if (underroot == 0) {
            x1 = -b/(2*a);
            x2 = x1;
            System.out.println("X1 = X2 = "+x1 );
        }
        else {
            x1 = (-b + Math.sqrt(underroot))/(2*a);
            x2 = (-b - Math.sqrt(underroot))/(2*a);
            System.out.println("X1 = "+x1);
            System.out.println("X2 = "+x2);
        }
    }
}
```

Exercise 2-7 To be discussed in the tutorial

Calculator

Write a Java program that designs a simple calculator. The program should read two rational numbers and a character that indicates the type of operation desired. Those operations include addition, subtraction, multiplication, division and calculating remainder. **Solve using switch statement only.**

Solution:

```
import java.util.Scanner;

public class Calculator {

    public static void main(String args[]) {
        double operand1, operand2, result;
        char operator;
        Scanner sc = new Scanner(System.in);

        System.out.print("Please type in the first operand: ");
        operand1 = sc.nextDouble();
        System.out.print("Please type in the second operand: ");
        operand2 = sc.nextDouble();
        System.out.print("Please type in the first character of the operation you want: ");
        operator = sc.next().charAt(0) ;
        switch (operator) {
        case 'a':
            /* falls through */
        case 'A':
```

```
            result = operand1 + operand2;
            System.out.println(operand1 + " + " + operand2 +" = "+ result);
            break;
        case 's':
            /* falls through */
        case 'S':
            result = operand1 - operand2;
            System.out.println(operand1 + " - " + operand2 +" = "+ result);
            break;
        case 'm':
            /* falls through */
        case 'M':
            result = operand1 * operand2;
            System.out.println(operand1 + " * " + operand2 +" = "+ result);
            break;
        case 'd':
            /* falls through */
        case 'D':
            result = operand1 / operand2;
            System.out.println(operand1 + " / " + operand2 +" = "+ result);
            break;
        case 'r':
            /* falls through */
        case 'R':
            result = operand1 % operand2;
            System.out.println(operand1 + " % " + operand2 +" = "+ result);
            break;
        default:
            System.out.println("There is no operation corresponding to this input!");
            break;
        }
    }
}
```

Exercise 2-8 To be discussed in the lab

Zodiac

Write a Java program that requests a month number (1-12) and a day number (1-31). The program should print the Zodiac Sign according to the user's input.

Sign	From	To
Capricorn	December 22	January 19
Aquarius	January 20	February 17
Pisces	February 18	March 19
Aries	March 20	April 19
Taurus	April 20	May 20
Gemini	May 21	June 20
Cancer	June 21	July 22
Leo	July 23	August 22
Virgo	August 23	September 22
Libra	September 23	October 22
Scorpio	October 23	November 21
Sagittarius	November 22	December 21

Solution:

```
import java.util.Scanner;

public class Zodiac {

    public static void main(String [] args) {

        int month, day;
        String horoscope;
        Scanner sc = new Scanner(System.in);

        System.out.print("Please enter the month(1-12): ");
        month = sc.nextInt();
        System.out.print("Please enter the day(1-31): ");
        day = sc.nextInt();

        switch (month) {
        case 1: horoscope = (day <= 19) ? "Capricorn" : "Aquarius";
                break;
        case 2: horoscope = (day <= 17) ? "Aquarius" : "Pisces";
                break;
        case 3: horoscope = (day <= 19) ? "Pisces": "Aries";
                break;
        case 4: horoscope = (day <= 19) ? "Aries" : "Taurus";
                break;
        case 5: horoscope = (day <= 20) ? "Taurus" : "Gemini";
                break;
        case 6: horoscope = (day <= 20) ? "Gemini" : "Cancer";
                break;
        case 7: horoscope = (day <= 22) ? "Cancer" : "Leo";
                break;
        case 8: horoscope = (day <= 22) ? "Leo" : "Virgo";
                break;
        case 9: horoscope = (day <= 22) ? "Virgo": "Libra";
                break;
        case 10: horoscope = (day <= 22) ? "Libra": "Scorpio";
                break;
        case 11: horoscope = (day <= 21) ? "Scorpio" : "Sagittarius";
                break;
        case 12: horoscope = (day <= 21) ? "Sagittarius" : "Capricorn";
                break;
        default: horoscope = "The value you entered for the month is not correct!";
                break;
        }
        System.out.print(horoscope);
    }

}
```

Exercise 2-9

Tire's Pressure

Write a program that reads in the pressure of the four tires and writes a message that says if the inflation is OK or not. Tires don't have to have exactly the same pressure. The front tires can be within 3 psi of each other, and the rear tires can be within 3 psi of each other. You must make sure that each tire has a pressure between 35 and 45.

```
Input right front pressure : 35
Input left front pressure : 37
Input right rear pressure : 41
Input left rear pressure : 44
Inflation is OK
```

Solution:

```
import java.util.Scanner;

public class Tire
{
    public static void main(String [] args) {
        Scanner sc = new Scanner(System.in);;
        int leftfront, rightfront, leftrear, rightrear;

        System.out.print ("Input the right front pressure : ");
        rightfront = sc.nextInt();
        System.out.print ("Input the left front pressure : ");
        leftfront = sc.nextInt();
        System.out.print ("Input the right rear pressure : ");
        rightrear = sc.nextInt();
        System.out.print ("Input the left rear pressure : ");
        leftrear = sc.nextInt();

        if((rightfront >= 35) && (rightfront <= 45 )
            && (leftfront >= 35) && (leftfront <= 45)
            && (rightrear >= 35) && (rightrear <= 45)
            && (leftrear >= 35) && (leftrear <= 45 )
            && ((rightfront-leftfront <= 3 && rightfront-leftfront >=0)
                || (rightfront-leftfront >= -3 && rightfront-leftfront <= 0 ))
            && ((rightrear-leftrear <= 3 && rightrear-leftrear >= 30
                || (rightrear-leftrear >= -3 && rightrear-leftrear <= 0 ))){
             System.out.println("The inflation is OK.");
        }
        else {
            System.out.println("Problem with inflation!");
        }
    }
}
```