

King Fahd University of Petroleum and Minerals
 College of Computer Sciences and Engineering
 Information and Computer Science Department
Second Semester 2009/2010 (092)
ICS 102 - Introduction to Computing I

Major Exam 02
 Sunday, 16th May 15, 2010
 Time: 120 minutes

Name:

Key

ID#:

--	--	--	--	--	--	--	--	--	--

G:

Please circle your section number below:

Section	01	02	03	04	05
Instructor	Ghouti	Ghouti	Zhioua	Al-Suhaim	Almuhammadi
Day and Time	SM 07:00-07:50	SM 08:00- 08:50	SM 09:00 - 09:50	SM 13:10-14:00	SM 10:00-10:50

Question #	Maximum Marks	Obtained Marks	Remarks
1	15		
2	25		
3	20		
4	20		
5	20		
Total	100		

~Good Luck~

Q1. [15 marks, 10 + 5] Java requires all variables to be declared and instantiated before they are used.

1. Write code fragments to perform each of the following:

- (a) Declare a one-dimensional array to keep the names of 100 students.

```
String[] names = new String[100];
```

- (b) Declare and instantiate a two-dimensional array with 8 rows and 9 columns that contains **boolean** values.

```
boolean[][] = new boolean[8][9];
```

- (c) Declare and instantiate 2 two-dimensional arrays, A and B, such that A stores a 5×5 matrix, and B stores a 5×1 matrix.

```
int[][] A = new int[5][5];  
int[][] B = new int[5][1];
```

- (d) Declare an array to hold the names of the months in a year.

```
String[] months = new String[12];  
// or String[] months = {"January", "February", "March", "April",  
"May", "June", "July", "August", "September", "October",  
"November", "December"};
```

- (e) Declare a two-dimensional array indexed from 0 through 10 for the first dimension and from 0 to 15 for the second dimension that contains **char** values.

```
char[][] charValues = new char[11][16];
```

2. Write a code fragment that declares a two-dimensional array, M, to store a 100×100 identity matrix.

Hint: the identity matrix has 1's in the main diagonal and 0's elsewhere, like this:

```
1 0 0  
0 1 0  
0 0 1
```

```
int[][] identityMatrix = new int[100][100];  
  
for(int i = 0; i < identityMatrix.length; i++)  
    identityMatrix[i][i] = 1;
```

Q2. [25 marks] Determine the output of each of the following error-free code fragments. If the output is infinite or has more than 6 lines, then write only the first 6 lines.

	Code Fragment	Output
a)	<pre>int counter = 0; for (int j = 0; j <= 500; j++) counter++; System.out.println(counter); counter = 0; for (int j = 100; j > 20; j--) counter++; System.out.println(counter); for (int j = 400; j < 500; j++) counter = j; System.out.println(counter);</pre>	<p>[3 marks]</p> <p>501 80 499</p>
b)	<pre>int x = 10, y = 0; while (x > ++y) x--; System.out.println(x + " " + y); x = 10; y = 0; do { x--; }while (x > ++y); System.out.println(x + " " + y);</pre>	<p>[4 marks]</p> <p>5 6 5 5</p>
c)	<pre>int i = 0; int j = 0; int k = 0; for (i = 1; i < 4; i++) for (j = 1; j < i; j++) System.out.println(i + " " + j); k++; System.out.println(i+ " " +j+ " " +k); // Becareful: no style</pre>	<p>[6 marks]</p> <p>2 1 3 1 3 2 4 3 1</p>
d)	<pre>int [] a = {10, 20, 30}; int [][] b = {{1,2,3}, {4,5,6}, {7,8,9}}; a[1] = a[2]++; b[1] = b[2]; for(int k = 0; k < b.length; k++) { System.out.println(a[k]+ " > " +b[k][1]); a[2-k]++; b[k][k] = 99; }</pre>	<p>[6 marks]</p> <p>10 > 2 30 > 8 32 > 99</p>
e)	<pre>String x = "*SureNiceExam*:p"; int a = 1, b = 0; do{ System.out.print(x.charAt(a++)); //same line if(a == 3) { System.out.print(x.charAt(5*a)); b = 4 } while (a <= b) System.out.print(x.charAt(b--)); }while(b != 2); System.out.println(x.substring(9,13));</pre>	<p>[6 marks]</p> <p>SuperExam</p>

Q3. [20 marks] The price of a product is marked down by 10% if you buy more than three kilos, and it is reduced by 20% if you buy over six kilos. Write a java program that prompts a user for a price per kilo (double) and the desired number of kilos (double). The program should print the total price for the product. The desired number of kilos must be at most 20.

Example:

Enter the price per kilo: **2.5**
Enter the desired number of kilos (20k max): **30**
Invalid!
Enter the desired number of kilos (20k max): **88**
Invalid!
Enter the desired number of kilos (20k max): **8**

Total amount after 20% off is: 16.0

```
import java.util.*;
public class TotalPrice {
    public static void main(String[] args) {

        Scanner myKeyboard = new Scanner(System.in);
        double pricePKG, numKG, totalPrice;
        do {
            System.out.print("Enter the price per kilo: ");
            pricePKG = myKeyboard.nextDouble();
        } while(pricePKG < 0.0);

        boolean invalid = false; // to check whether entered data is invalid
        do {
            System.out.print("Enter the desired number of kilos (20k max): ");
            numKG = myKeyboard.nextDouble();

            invalid = (numKG < 0.0 | numKG > 20.0);
            if(invalid)
                System.out.println("Invalid!");
        } while(invalid);

        if(numKG > 6.0) {
            totalPrice = (pricePKG * numKG) - ( 0.2 * (pricePKG * numKG));
            System.out.println("Total amount after 20% off is: " + totalPrice);
        }
        else if(numKG > 3.0) {
            totalPrice = (pricePKG * numKG) - ( 0.1 * (pricePKG * numKG));
            System.out.println("Total amount after 10% off is: " + totalPrice);
        }
        else {
            totalPrice = pricePKG * numKG;
            System.out.println("Total amount after 0% off is: " + totalPrice);
        }
    }
}
```

Q4. [20 marks] Write a java program that computes the multiplication table for integers from 0 to 1000. It saves the table in a text file named "MulTable.txt". The output should be as follows:

	0	1	2	3	4	5	6	7	8	9	...	1000
0	0	0	0	0	0	0	0	0	0	0		
1	0	1	2	3	4	5	6	7	8	9		
2	0	2	4	6	8	10	12	14	16	18		
.....												
1000	0	1000	2000	3000	4000	5000	6000	7000	8000	9000	1000000

```

public class MultiplicationTable {
    public static void main(String[] args) throws FileNotFoundException {

        int n = 1000; // for a Multiplication-Table of size  n x n

        System.out.print("Opening output file...");
        PrintWriter fileOut = new PrintWriter(new FileOutputStream("MulTable.txt"));
        System.out.println("Ok!");

        System.out.print("Writing to output file. Please wait...");

        //Output header row 0  1  2  ... n
        for (int i = 0; i <= n; i++)
            fileOut.print("\t" + i);
        fileOut.println();

        //Output table rows 0..n
        for (int i = 0; i <= n; i++)
        {
            fileOut.print(i);
            for (int j = 0; j <= n; j++)
                fileOut.print("\t" + (i * j) );
            fileOut.println();
        }

        fileOut.close();
        System.out.println(" Done!!");

    }
}

```

Q5. [20 marks] Assume you have a table representing results of soccer season of 32 teams, like:

	Win	Draw	Loss
Team 1	11	2	1
Team 2	8	3	3
Team 3	8	2	4
...			
Team 32	1	4	9

Suppose the table is saved in a file named "data.txt" that has 32 lines, and each line in the file contains 3 integers representing the win, draw, and loss counts for each team. Write a java program that stores these values in an appropriate 2-D array then creates a 1-D array to store the points. (win = 3 points, draw = 1 point, loss = 0 points). Then your program should print the information appropriately, like:

Team 1: 35 points
 Team 2: 27 points
 Team 3: 26 points
 ...
 Team 32: 7 points

```

public class Soccer {
    public static void main(String[] args) throws FileNotFoundException {

        Scanner fileReader = new Scanner (new FileInputStream("data.txt"));

        int dataArray[][] = new int [32][3];

        int pointsArray[] = new int [32];

        for(int i=0 ; i<dataArray.length ; i++)
            for(int j=0 ; j<dataArray[i].length ; j++ )
                dataArray[i][j] = fileReader.nextInt();

        for(int i=0 ; i<dataArray.length ; i++)
        {
            pointsArray[i] = dataArray[i][0] * 3 + dataArray[i][1];

            System.out.println("Team " + (i+1) + " : "
                               + pointsArray[i] + "points" );
        }
    }
}
  
```