King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Information and Computer Science Department
Second Semester 2010/2011 (102)
ICS 102 - Introduction to Computing I

Final Exam
Sunday, June 12, 2011
Time: 150 minutes

Name: _____

ID#: _____ **C:** _____

Please circle your section number below:

Section	01	02	03	04	05
Instructor	Baqais	Baqais	Al-Sukairi	Ghouti	Al-Sukairi
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

Note:

1. This is a closed book, closed notes exam.
2. Usage of calculators, laptops and cell phones is prohibited during the exam.
3. Please **switch off** your cell phones NOW.

Question #	Out Of	Grade
1	20	
2	20	
3	20	
4	20	
5	20	
Total	**100**	

~Good Luck ~

Q1. [20 marks] Consider the following code definition:

```
public class Matrix {
     int[][] matrix;
     private static final int NROWS = 10, NCOLS = 10;
}
```

For each of the following methods in this class, write only the headers of the methods as described below (Do not provide the body of the methods).

Hint: The methods can be **public**, **private**, **static**, **non-static**, **void**, etc...

a) *getAverageRow* - returns the average row whose elements give the average of the rows of **matrix**.

b) *getAverageColumn* - returns the average column whose elements give the average of the columns of **matrix**.

c) *isSquare* - checks whether this **matrix** has the same number of rows and columns.

d) *diagonal* – returns the elements located on the diagonal of **matrix**.

e) *matrixAdd* - takes two matrices and return their sum.

Note: The elements on the diagonal of a matrix as shown below in a shaded color.

1	4	5	7	8
2	**0**	-4	34	11
14	23	**178**	1678	3
123	13	-34	**234**	0
456	12	324	12	**22**

Q2. [30 marks] Consider the following definition of the class **`Rectangle2D`**

```
public class Rectangle2D {
  private Point upperLeftCorner;   // Point representing the upper left corner
  private int width, height;   // width and height of the Rectangle2D
```

(a) Provide the following constructors and methods:

1- A *full-argument* constructor that creates a circle object and initializes its fields to a given **upperLeftCorner**, **width** and **height** (in this order).

2- A *no-argument* (*default*) constructor that creates a Rectangle2D object of ***width = 5***, ***height = 5*** and whose upper left corner is located at the origin (**0, 0**).

3- A *copy constructor* for this class.

4- Accessor methods for **upperLeftCorner**, **width** and **height**.

5- Mutator methods for **upperLeftCorner**, **width** and **height**.

6- A method **toString** that converts the information of this Rectangle2D object into a string as in the following example: **Rectangle2D whose upper left corner is located at (6, 12) has width = 4 and height = 6.**

7- A method **equals** that returns **true** if this Rectangle2D object has the same width and height as the given Rectagle2D and their upper left corners are located at the same point. Otherwise, the method should return **false**.

8- A method **isSquare** that returns **true** if this Rectangle2D object has the same width and height and returns **false** otherwise.

9- A method **getPerimeter()** that returns the perimeter of this Rectangle2D object.
Note: The perimeter of a rectangle is given by: **2 * (width + height)**

10- A method **getArea ()** that returns the area of this Rectangle2D object.
Note: The area of a rectangle is given by: **width * height**

(b) The class **Rectangle2D** assumes the existence of the **Point** class as follows:

```
public class Point {
  private int xPos, yPos; // x-y coordinates

  // other code
}
```

You may assume the availability of the following constructors and methods:

1- Constructors:

```
public Point()   public Point(int x, int y)   public Point(Point p)
```

2- Accessor and mutator methods:

```
public int getX() public int getY()   public void setX(int x)  public void setY(int y)
```

3- Other methods:

```
public String toString()   public boolean equals()
```

Q3. [20 marks]
Write a program that reads 50 exam grades from the file "c:\data.txt" and computes and displays the following: highest grade, average grade, and standard deviation.

$$standard\_deviation = \sqrt{\frac{\sum grade^2}{No\_of\_grades} - average^2}$$

Note: this program can be solved without using arrays.

Q4. [20 marks]

a. What is the output of the following main program?

```
public static void main(String[] args) {
   int x,y;
   a = 5;
   b = 1;
   while (a > 0) {
         a = a – 1;
         b = b * a;
         System.out.println(b);
   }
    }
```

1.
   55
   24
   24
   24
   0
2.
   5
   12
   12
   24
   24
3.
   4
   12
   24
   24
   0
4.
   None of the Above

b. What is the output of the following program?

```
public class Example {
  public static void main(String args[]) {
    int x=0, y=2;
    do {
      x=++x;
      y--;
    } while(y>0);
    System.out.println(x);
  }
}
```

1. 0
2. 1
3. 2
4. Compilation Error

c. What output is displayed as the result of executing the following statement?

```
System.out.println("// Looks like a comment.");
```

1. // Looks like a comment.
2. The statement results in a compilation error
3. Looks like a comment
4. No output isdisplayed

d. What is the output of the following code for exampleprint(1)?

```
static void exampleprint(int inputitem) {
     if (inputitem == 0) {
        System.out.print("*");
     }
     else {
        System.out.print("[");
        exampleprint(inputitem - 1);
        System.out.print(",");
        exampleprint(inputitem - 1);
        System.out.println("]");
     }
}
```

1. *,*,*
2. [*,*]
3. *
4. None of the Above

Q5. [20 marks] For each code on the left column, determine the content of the arrays on the right column:

<table>
<tr><th>Code</th><th>Array content</th></tr>
<tr><td>

```
int [] a = new int [5];
a[0] = 3;
for(int i = 1; i < a.length; i++)
    a[i] = a[i-1] + 2;
```

</td><td>

a:

3	5	7	9	11

1 point each

</td></tr>
<tr><td>

```
boolean [] a = {true, true, false, false, true};
boolean [] b = a.clone();
for(int i = 0; i < 5; i++)
   b[i] = a[4-i] && b[i];
```

</td><td>

b:

T	F	F	F	T

1 point each

</td></tr>
<tr><td>

```
char [][] a = new char[3][3];
char [] b = {'J', 'A', 'V', 'A'};
for(int i = 0; i < 3; i++)
  for(int j = 0; j < 3; j++)
    a[j][i] = b[(i * 3 + j) % 4];
```

</td><td>

a:

J	A	V
A	J	A
V	A	J

-0.5 point for each incorrect value

</td></tr>
<tr><td>

```
int a [][] = {{92, 88, 75},{80, 89, 71},{96, 80, 85}};
int b [] = a[1];
int x = b[0];
for(int i = 0; i < 3; i++)
  if(x > b[i]) x = b[i];
for(int i = 0; i < 3; i++)
  for(int j = 0; j < 3; j++)
    a[i][j] -= x;
```

</td><td>

a:

21	17	4
9	18	0
25	9	14

-0.5 point for each incorrect value

</td></tr>
</table>