

First Semester 2014/2015 (141)  
ICS 102 - Introduction to Computing I

Final Exam  
Tuesday, 6<sup>th</sup> January 2015  
Time: 120 minutes

Name:

ID#:

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

Please circle your section number below:

Section	01	02	04
Instructor	Said	Garout	Garout
Day and Time	UT 08:00-08:50	UT 10:00- 10:50	UT 09:00 - 09:50

Question #	Maximum Score	Score
1	24	
2	8	
3	15	
4	20	
5	33	
<b>Total</b>	<b>100</b>	

~Good Luck~

**Question#1 (24 points: 8 \* 3):**

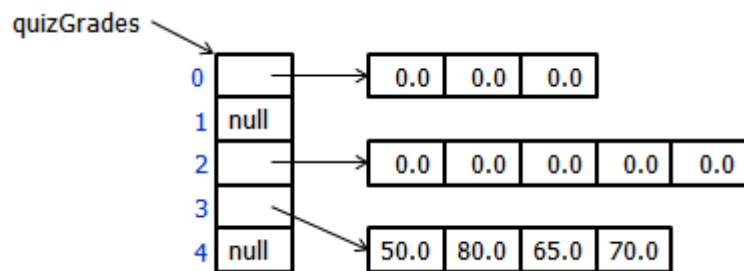
Identify the error(s), if any, in each of the following fragments. If a fragment has no errors, write its output.

	Program Fragment	Output
1.	<pre> public class Q1_01{     private static int count = 12;     private static void method1(){         int count = 7;     }     public static void main(String[] args){         method1();         System.out.println(count);     } } </pre>	
2.	<pre> public class Q1_02{     public static void main(String[] args) {         int[] y = {8, 9, 6};         int[] x = {1, 2, 3, 4, 0, 0, 0};         int k;         for(k = 2; k &lt;= 3; k++){             x[k + 3] = x[k];             x[k] = y[k - 2];         }         x[k] = y[k - 2];         for(k = 5; k &gt;= 0; k--)             System.out.print(x[k]+ " ");     } } </pre>	
3.	<pre> public class Q1_03{     public static void main(String[] args){         int[][] x = { {5, 2, 4}, {3, 7, 1}, {9, 6, 8}};         for(int k = 2; k &gt;= 0; k--){             for(int m = 2; m &gt;= 0; m--){                 System.out.print(x[m][k] + " ");             }             System.out.println();         }     } } </pre>	
4.	<pre> public class Q1_04{     public static void main(String[] args){         int[] b = {11, 22, 33};         C4 c = new C4(b);         int[] d = c.getX();         for(int i = 0; i &lt; d.length; i++)             d[i] = i;         System.out.println(c);     } } class C4{     private int[] x;     C4(int[] a){         x = a;     }     public int[] getX(){         return x;     }     public String toString(){         String S = x[0] + "";         for(int i = 1; i &lt; x.length; i++)             S = S + "\t" + x[i];         return S;     } } </pre>	

	Program Fragment	Output
5.	<pre> public class Q1_05{     public static void main(String[] args){         int k = 2;         String str = "KFUPM";         nPrint(str, k);         System.out.println(str + ", " + k);     }     static void nPrint(String str, int n){         while (n &gt; 0){             str = str.substring(0,n);             System.out.println(str);             n--;         }     } } </pre>	
6.	<pre> public class Q1_06{     public static void main(String[] args){         C6 obj1 = new C6(2);         method1(obj1);         System.out.println(obj1.x);         C6 obj2 = new C6(3);         method2(obj2);         System.out.println(obj2.x);     }     public static void method1(C6 obj){         obj.x = 4;     }     public static void method2(C6 obj){         obj = new C6(7);     } } class C6{     public int x;     C6(int x){         this.x = x;     } } </pre>	
7.	<pre> public class Q1_07{     public static void main(String[] args){         System.out.println(method( ));     }     int method( ){         return 20;     } } </pre>	
8.	<pre> public class Q1_08{     public static void main(String[] args){         C8 obj = new C8( );         System.out.println(obj.x);     } } class C8{     private int x;     C8( ){         x = 100;     } } </pre>	

**Question#2 (8 points):**

Write Java statements to create the following arrays:

**Question#3 (15 points):**

Write a static boolean method **arrayEquals** that takes as input two 2-D arrays of type double and it compares the two arrays for equality. Your method must work in all cases.

**Question#4 (20 points):**

Write a complete Java program that contains two static methods:

- (a) A static Java method **rotateLeftN** that takes as input an array of integers and an integer **n**, it then returns an array that is an n-left rotation of the passed array. Your method must not modify the array whose reference is passed to the method. It must also throw an appropriate exception if n is less than 1 or greater or equal to the passed array size.

**Example:** If the passed array is:

1	5	3	8	12	4
---	---	---	---	----	---

Then, a 1-left rotation will result in:

5	3	8	12	4	1
---	---	---	----	---	---

A 2-left rotation will result in:

3	8	12	4	1	5
---	---	----	---	---	---

- (b) A main method that initializes an array of integers with appropriate values. It then calls rotateLeftN method. It finally displays the returned array.

**Note:** Your program must be general

**Question#5 (33 points):**

- (a) A **Faculty** has an ID and a number of advisees. The ID cannot be modified after it is initialized to a non-default value.

Write an appropriate **Faculty** class that enables:

- **Faculty** objects are assigned unique consecutive IDs starting at id 2014001.
- The counting of total number of advisees for all **Faculty** objects.

The class must have:

- All its variables to be private
- A one-argument constructor that throws an appropriate exception
- A copy constructor
- Appropriate get-methods
- An appropriate set method that throws an appropriate exception.
- An appropriate **toString** method

- (b) A **University** has private number of faculty and a private array of **Faculty** objects.

Write an appropriate **University** class that has:

- A one-argument constructor that takes the maximum number of faculty in the university. The constructor throws an appropriate exception if the argument is less than 50.
- **addFaculty** method. The method adds a faculty to the array of **Faculty** objects. The method throws an appropriate exception. Your method must not have reference leaks.
- **getFacultyArray** method. Your method must not have reference leaks.

- (c) Write an appropriate **UniversityDriverClass** that performs the following in order:

- Creates a **University** object with a maximum of 60 faculty.
- Adds at least four **Faculty** objects to the **University** object.
- Decreases by 2 the number of advisees of the last **Faculty** object in the array of **Faculty** objects. [Your code must be general; it must work for any array length]
- Displays the total number of advisees for all faculty. [The counting must neither be done in the driver class nor use loops]
- Displays the Faculty array.
- Determines whether the faculty with ID 201403 exists in the array or not by displaying an appropriate message.

**Sample program run:**

Total number of advisees: 33

Faculty list:

[ID#: 2014001, Number of advisees: 15]

[ID#: 2014002, Number of advisees: 10]

[ID#: 2014003, Number of advisees: 6]

[ID#: 2014004, Number of advisees: 2]

The faculty 2014003 exists



