

King Fahd University of Petroleum and Minerals  
College of Computer Sciences and Engineering  
Information and Computer Science Department  
Second Semester 2016/2017 (162)  
ICS 102 - Introduction to Computing I

**Final Exam**  
**Tuesday, May 23<sup>rd</sup>, 2017**  
**Time: 120 minutes**

Name:

ID#:

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

Please circle your section number below:

Section	01	02	03	04
Instructor	Zhioua	Zhioua	Ghouti	Ghouti
Day and Time	UT 09:00-09:50	UT 09:00- 09:50	UT 11:00 - 10:50	UT 10:00 - 10:50

Question #	Maximum Marks	Obtained Marks
1. Concepts I	20	
2. Concepts II	40	
3. Programming I	30	
4. Programming II	30	
<b>Total</b>	<b>120</b>	

Important Note: Make sure to answer in the answers' sheet paper

**Q1. [20 Points] Program Understanding I**

Answer each of the following questions using **True/False** in the table given in the answer sheets:

- 1) A variable of type **boolean** can be explicitly converted to that of type **int**.
- 2) The modulus operator, %, returns the remainder of integer division.
- 3) In Java, **Strings** are immutable objects. Immutable objects can be changed.
- 4) The **Scanner** class has a method **next()** that allows an entire line of string text to be read.
- 5) In a switch statement, the default case is always executed.
- 6) Not including the break statements within a switch statement results in a syntax error.
- 7) An empty statement is defined as a loop that runs forever.
- 8) In a method invocation, there must be exactly the same number of arguments in parentheses as there are formal parameters in the method definition heading.
- 9) The modifier private means that an instance variable can be accessed by name outside of the class definition.
- 10) The modifier private means that an instance variable can be accessed by name outside of the class definition.
- 11) In a static method, you may use the **this** parameter either explicitly or implicitly.
- 12) Wrapper classes provide a class type corresponding to each of the primitive types so that you can have class types that behave somewhat like primitive types.
- 13) Wrapper classes are provided for all primitive Java types except **Boolean**.
- 14) An array is a collection of variables all of the same type.
- 15) An array has only one public instance variable, which is named length.
- 16) An array of **chars** is the same as a **String** in Java.
- 17) An array name references a memory address.
- 18) You can only use array indexed variables as arguments to methods.
- 19) A method cannot change the values stored in the indexed variables of an array argument.
- 20) A one dimensional array is also called an array of arrays.

**Q2. [40 marks] Program Understanding II**

Answer each of the following questions in the table below:

- 1) Identify the invalid Java identifier.
  - (a) 1Week
  - (b) Week1
  - (c) amountDue
  - (d) amount\_due
  
- 2) What is the Java expression for  $4a^2 + 2b * c$ ?
  - (a)  $(4 * a) + (2 * b) * c$
  - (b)  $(4 * a * a) + ((2 * b) * c)$
  - (c)  $((4 * a * a) + (2 * b)) * c$
  - (d)  $(4 + a * a) + ((2 + b) * c)$
  
- 3) What is the value of the variable c in the statements that follow?

**String phrase = "Make hay while the sun is shining.;"**  
**char c = phrase.charAt(10);**

  - (a) w
  - (b) h
  - (c) i
  - (d) None of the above
  
- 4) Valid arguments to the **System.out** object's **println()** method include:
  - (a) "Anything with double quotes"
  - (b) **String** variables
  - (c) Variables of type **int**
  - (d) All of the above

- 5) The statement: `System.out.printf("%6.2f", 597.7231);` displays:
- (a) 597.723
  - (b) 597.72
  - (c) 000597.72
  - (d) None of the above
- 6) What does the following code output?
- ```
DecimalFormat percent = new DecimalFormat("0.00%");
System.out.println(percent.format(0.308));
```
- (a) 3.080%
  - (b) 30.80%
  - (c) .0308%
  - (d) 308.0%
- 7) A multi-way **if-else** statement
- (a) allows you to choose one course of action.
  - (b) always executes the else statement.
  - (c) allows you to choose among alternative courses of action.
  - (d) executes all Boolean conditions that evaluate to true.
- 8) The controlling expression for a **switch** statement includes all of the following types except:
- (a) char
  - (b) int
  - (c) byte
  - (d) double
- 9) When using a compound Boolean expression joined by an **&& (AND)** in an **if** statement:
- (a) Both expressions must evaluate to true for the statement to execute.
  - (b) The first expression must evaluate to true and the second expression must evaluate to false for the statement to execute.
  - (c) The first expression must evaluate to false and the second expression must evaluate to true for the statement to execute.
  - (d) Both expressions must evaluate to false for the statement to execute.
- 10) The \_\_\_\_\_ operator has the highest precedence.
- (a) \*
  - (b) dot
  - (c) +=
  - (d) decrement
- 11) A \_\_\_\_\_ statement terminates the current iteration of a loop.
- (a) Break
  - (b) Continue
  - (c) Switch
  - (d) Throw

- 12) A \_\_\_\_\_ statement terminates the execution of a loop.
- (a) Break
  - (b) Continue
  - (c) Switch
  - (d) Throw
- 13) The new operator:
- (a) allocates memory
  - (b) is used to create an object of a class
  - (c) associates an object with a variable that names it.
  - (d) All of the above.
- 14) A variable whose meaning is confined to an object of a class is called:
- (a) instance variable
  - (b) local variable
  - (c) global variable
  - (d) none of the above
- 15) A variable whose meaning is confined to a method definition is called an/a
- (a) instance variable
  - (b) local variable
  - (c) global variable
  - (d) none of the above
- 16) In Java, call-by-value is only used with:
- (a) objects
  - (b) primitive types
  - (c) this
  - (d) all of the above
- 17) The parameter this refers to
- (a) instance variables
  - (b) local variables
  - (c) global variables
  - (d) the calling object
- 18) Two methods that are expected to be in all Java classes are:
- (a) getName and setName
  - (b) toString and equals
  - (c) compareTo and charAt
  - (d) toLowerCase and toUpperCase

- 19) Java has a way of officially hiding details of a class definition. To hide details, you mark them as \_\_\_\_\_.  
(a) public  
(b) protected  
(c) private  
(d) all of the above
- 20) A static method is one that can be used with a \_\_\_\_\_.  
(a) instance variable  
(b) local variable  
(c) global variable  
(d) the class name as a calling object
- 21) Static variables are often used:  
(a) in arithmetic expressions  
(b) to communicate between objects  
(c) within looping structures  
(d) all of the above
- 22) Only \_\_\_\_\_ copy/copies of a static variable are available to objects of a class.  
(a) one  
(b) two  
(c) three  
(d) none of the above
- 23) All of the following are wrapper classes except:  
(a) String  
(b) Integer  
(c) Character  
(d) Double
- 24) Converting from a value of primitive type to a corresponding object of its associated wrapper class is called:  
(a) Boxing  
(b) Converting  
(c) Unboxing  
(d) Reinstantiating
- 25) The conversion from an object of a wrapper class to a value of its associated primitive type is called:  
(a) Boxing  
(b) Converting  
(c) Unboxing  
(d) Reinstantiating

- 26) A copy constructor has \_\_\_\_\_ parameters.
- (a) zero
  - (b) one
  - (c) two
  - (d) three
- 27) The correct syntax for accessing the length of an array named Numbers is:
- (a) Numbers.length()
  - (b) Numbers.length
  - (c) both A and B
  - (d) none of the above
- 28) An **ArrayIndexOutOfBoundsException** error is a:
- (a) compiler error
  - (b) syntax error
  - (c) logic error
  - (d) all of the above
- 29) Which of the following initializer lists correctly initializes the indexed variables of an array named myDoubles?
- (a) double myDoubles[double] = {0.0, 1.0, 1.5, 2.0, 2.5};
  - (b) double myDoubles[5] = new double(0.0, 1.0, 1.5, 2.0, 2.5);
  - (c) double[] myDoubles = {0.0, 1.0, 1.5, 2.0, 2.5};
  - (d) array myDoubles[double] = {0.0, 1.0, 1.5, 2.0, 2.5};
- 30) The base type of an array may be all of the following but:
- (a) String
  - (b) boolean
  - (c) long
  - (d) all of these may be a base type of an array.
- 31) The correct syntax for passing an array as an argument in a method is:
- (a) a[]
  - (b) a()
  - (c) a
  - (d) a[0]..a[a.length]
- 32) Partially filled arrays require:
- (a) a variable to track the number of array positions used
  - (b) the use of local variables
  - (c) the use of global variables
  - (d) all of the above

- 33) An array with more than one index is called a/an:
- (a) partially filled array
  - (b) multidimensional array**
  - (c) bidirectional array
  - (d) one dimensional array
- 34) A array in which different rows can have different number of columns \_\_\_\_\_:
- (a) is a valid Java array**
  - (b) is a partially filled array
  - (c) will cause a runtime error
  - (d) will cause a compile error
- 35) Try blocks contain code that could possibly:
- (a) handle an exception
  - (b) throw an exception
  - (c) catch an exception
  - (d) display an exception
- 36) If a method does not catch an exception, then it must at least warn programmers that any invocation of the method might possibly throw an exception. This warning is called a/an:
- (a) Exception handler
  - (b) Throws clause
  - (c) Try block
  - (d) Catch block
- 37) If a method throws an exception, and the exception is not caught inside the method, then the method invocation:
- (a) terminates
  - (b) transfers control to the catch block
  - (c) transfers control to the exception handler
  - (d) none of the above
- 38) Methods that process String arguments as if they were numbers could possibly throw a/an \_\_\_\_\_ exception.
- (a) NumberFormatException
  - (b) NullPointerException
  - (c) both a and b
  - (d) none of the above
- 39) Exceptions that are subject to the catch or declare rule are called:
- (a) Checked exceptions
  - (b) Unchecked exceptions
  - (c) Fatal exceptions
  - (d) All of the above



- 40) A runtime exception is a/an:
- (a) checked exception
  - (b) unchecked exception
  - (c) offending exception
  - (d) none of the above

**Q3. [30 Points] Programming I**

Write a static method *rotateMatrix* which takes as inputs:

- 1- A 2D array *m*
- 2- A 1D array *a*.

The method should then return the 2D array where each row has been rotated *x* positions to the right.

The amount of rotation positions *x* is provided by the 1D array values.

**Example:**

if the 1D array *a* is :

|   |   |   |
|---|---|---|
| 1 | 2 | 4 |
|---|---|---|

And the 2D array *m* is :

|   |   |    |    |   |
|---|---|----|----|---|
| 1 | 4 | 9  | 16 | 9 |
| 5 | 7 | 1  | 3  | 8 |
| 2 | 4 | 12 | 6  | 3 |

The first row [1,4,9,16,9] should be rotated 1 position to the right, the second row [5,7,1,3,8] should be rotated 2 positions to the right, and the third row [2,4,12,6,3] should be rotated 4 positions to the right.

**returned array:**

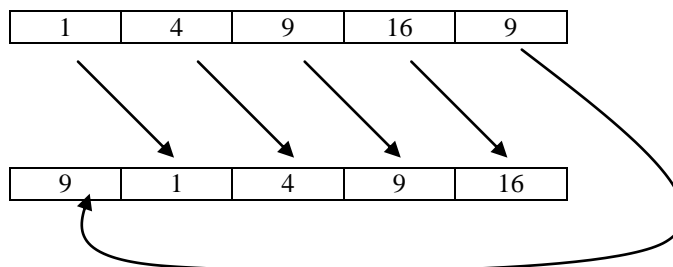
|   |    |   |   |    |
|---|----|---|---|----|
| 9 | 1  | 4 | 9 | 16 |
| 3 | 8  | 5 | 7 | 1  |
| 4 | 12 | 6 | 3 | 2  |

**Important Note:**

The method should check that all values are valid:

- The number of elements in *a* is equal to the number of rows in *m*.
- The maximum value for a rotation value is the number of columns of *m*.

In case there is a problem, the method should print an appropriate message and return **null**.

**Graphical illustration of 1 position rotation:**

**Q4. [30 Points] Programming II**

Design a java class **DynamicArray** that has the following instance variables:

- **data** – a 1D array of integers.
- **size**- number of filled elements in the array **data**.
- **MAX\_SIZE**- an integer constant representing the maximum capacity of the **DynamicArray**

- Write the code to declare the class **DynamicArray** and the above members
- Provide a constructor that takes as parameter an integer **n** and creates an empty array of size **n** and assigns it to **data**.

**Important Note:** Make sure that the **n** value is valid.

- Provide a copy constructor for **DynamicArray**
- Implement a method *equals()* for **DynamicArray** class. Two **DynamicArray** objects are equals if they have exactly the same list of elements in the same positions.
- Provide a method *addElement()* to add a new integer value into the **DynamicArray**.

**Important Note:** If all the cells in the array **data** are filled, it should be expanded by doubling its number of cells. But of course, this should not exceed **MAX\_SIZE**.