CS203 Java Programming and Applications
Fall 2019

Assignment 1 (7 Questions, 100 marks)

Assigned Date: September 17, 2019
Due Date: October 6, 2019 @ 23:59

QUESTION 1 (10 Marks) Formatting Console Output

Show the output of the following statements.

(a) `System.out.printf("amount is %f %e\n", 32.32, 32.32);`

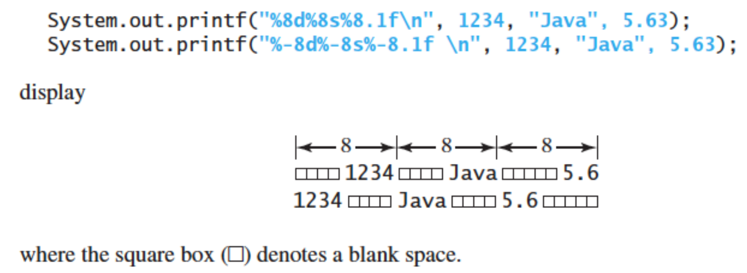
(b) `System.out.printf("amount is %5.2%% %5.4e\n", 32.327, 32.32);`

(c) `System.out.printf("%6b\n", (1 > 2));`

(d) `System.out.printf("%6s\n", "Java");`

(e) `System.out.printf("%-6b%s\n", (1 > 2), "Java");`

Write your answers as the two examples given below:



```
System.out.printf("%8d%8s%8.1f\n", 1234, "Java", 5.63);
System.out.printf("%-8d%-8s%-8.1f \n", 1234, "Java", 5.63);
```

display

```
|←—8—→|←—8—→|←—8—→|
□□□□1234□□□□Java□□□□□5.6
1234□□□□Java□□□□5.6□□□□□
```

where the square box (□) denotes a blank space.

Note: There is a missing "f" after "5.2" in (b).

QUESTION 2 (15 Marks) Methods

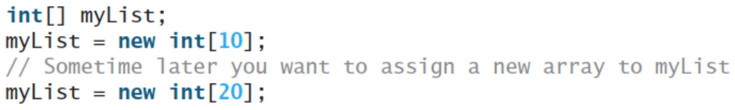
A pentagonal number is defined as $n(3n-1)/2$ for $n = 1, 2, \cdots$, and so on. Therefore, the first few numbers are 1, 5, 12, 22, $\cdots$. Write a method with the following header that returns a pentagonal number:

```
public static int getPentagonalNumber(int n)
```

Write a test program with the class name TestPentagonalNumber which uses the above method to display the first 100 pentagonal numbers with 10 numbers on each line. The method getPentagonalNumber is to be invoked in your main method. Each pentagonal number occupies 6 spaces and is left justified. Use proper CamelCase Notation for all identifiers in your program.

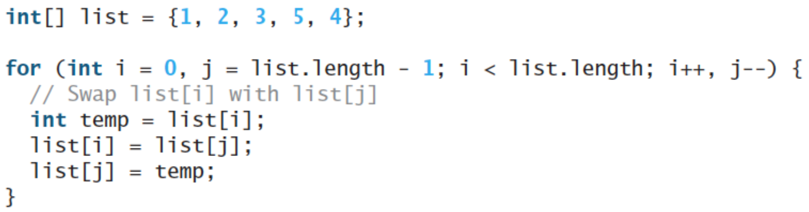
QUESTION 3 (10 Marks) Basics of One-Dimensional Arrays

(1) Once an array is created, its size cannot be changed. Does the following code resize the array? Explain your answer.



```
int[] myList;
myList = new int[10];
// Sometime later you want to assign a new array to myList
myList = new int[20];
```

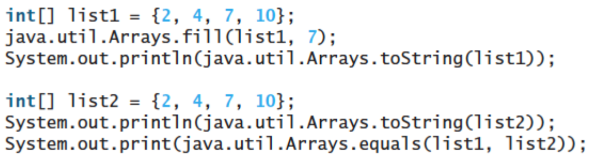
(2) Suppose the following code is written to reverse the contents in an array, explain why it is wrong. How do you fix it?



```
int[] list = {1, 2, 3, 5, 4};

for (int i = 0, j = list.length - 1; i < list.length; i++, j--) {
  // Swap list[i] with list[j]
  int temp = list[i];
  list[i] = list[j];
  list[j] = temp;
}
```

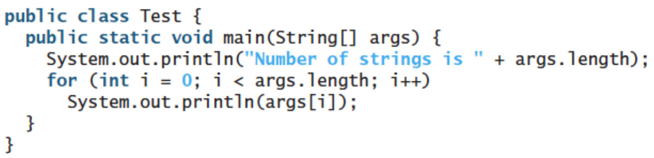
(3) Show the output of the following code:



```
int[] list1 = {2, 4, 7, 10};
java.util.Arrays.fill(list1, 7);
System.out.println(java.util.Arrays.toString(list1));

int[] list2 = {2, 4, 7, 10};
System.out.println(java.util.Arrays.toString(list2));
System.out.print(java.util.Arrays.equals(list1, list2));
```

(4) Compile and run the following program in command-line environment and make screenshots to show your results.



```
public class Test {
  public static void main(String[] args) {
    System.out.println("Number of strings is " + args.length);
    for (int i = 0; i < args.length; i++)
      System.out.println(args[i]);
  }
}
```

Use the following commands one by one:

a) ls command: The listed files and directories should include “Test.java” and no “Test.class”.

b) javac command: Use it to compile your “Test.java”, which produces “Test.class”.

c) ls command: The listed files and directories should now include both “Test.java” and “Test.class”.

d) java command: Use it three times to invoke the program as follows:

```
java Test I have a dream
java Test “1 2 3”
java Test
```

Your screenshots should include all the above commands, and the outputs of the three invocations. (Hint: You may need to import java.lang.String in the program.)

QUESTION 4 (15 Marks) Programming with One-Dimensional Arrays

Write two overloaded methods in a class named `AverageOfArray` that return the average of an array with the following headers:

```
public static int average(int[] array)
public static double average(double[] array)
```

Write a `main` method that prompts the user to enter ten double values, invokes this method, and displays the average value. Use CamelCase Notation for all identifiers in your program.

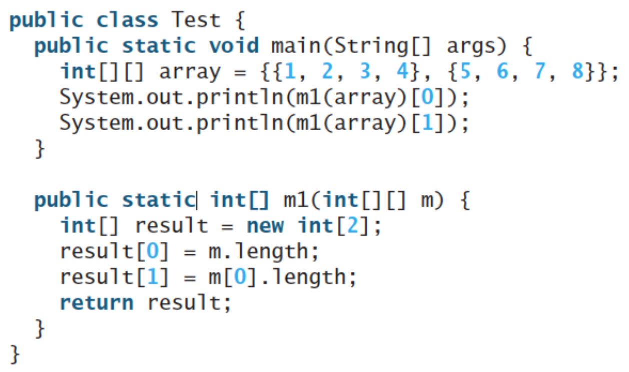
QUESTION 5 (15 Marks) Basics of Multidimensional Arrays

Show the output of the following codes:

(1)

```
int[][] array = {{1, 2}, {3, 4}, {5, 6}};
int sum = 0;
for (int i = 0; i < array.length; i++)
  sum += array[i][0];
System.out.println(sum);
```

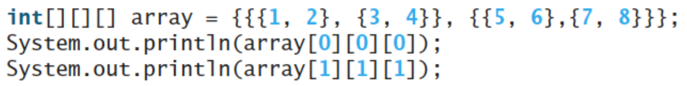
(2)



```java
public class Test {
  public static void main(String[] args) {
    int[][] array = {{1, 2, 3, 4}, {5, 6, 7, 8}};
    System.out.println(m1(array)[0]);
    System.out.println(m1(array)[1]);
  }

  public static int[] m1(int[][] m) {
    int[] result = new int[2];
    result[0] = m.length;
    result[1] = m[0].length;
    return result;
  }
}
```

(3)



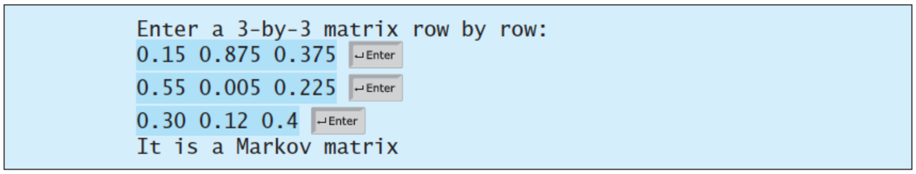
```java
int[][][] array = {{{1, 2}, {3, 4}}, {{5, 6},{7, 8}}};
System.out.println(array[0][0][0]);
System.out.println(array[1][1][1]);
```

QUESTION 6 (20 Marks) Programming with Multidimensional Arrays

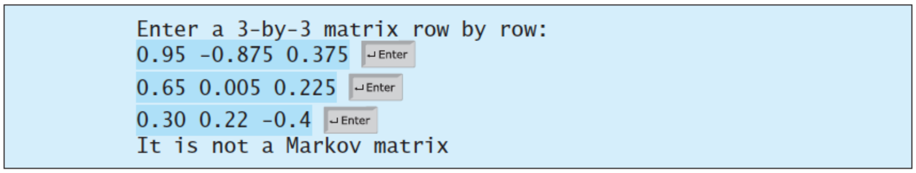
An $n \times n$ matrix is called a positive Markov matrix if each element is positive and the sum of the elements in each column is 1. Write the following method in a class named MarkovMatrix to check whether a matrix is a (positive) Markov matrix.

```java
public static boolean isMarkovMatrix(double[][] m)
```

Write a main method that prompts the user to enter a 3×3 matrix of double values and tests whether it is a Markov matrix. Here are sample runs:



```
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375 ↵Enter
0.55 0.005 0.225 ↵Enter
0.30 0.12 0.4 ↵Enter
It is a Markov matrix
```



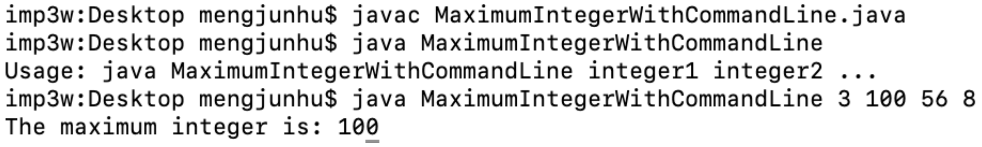
```
Enter a 3-by-3 matrix row by row:
0.95 -0.875 0.375 ↵Enter
0.65 0.005 0.225 ↵Enter
0.30 0.22 -0.4 ↵Enter
It is not a Markov matrix
```

QUESTION 7 (15 Marks) Strings and Arrays

Write a main method in a class named MaximumIntegerWithCommandLine to get integers from command line arguments and display the maximum of these integers. The user can arbitrarily input any number of integers. If no integer is input from user, your program displays the following message and terminates.

```
Usage:  java MaximumIntegerWithCommandLine integer1 integer2 ...
```

The following screenshot is a sample run.



```
imp3w:Desktop mengjunhu$ javac MaximumIntegerWithCommandLine.java
imp3w:Desktop mengjunhu$ java MaximumIntegerWithCommandLine
Usage: java MaximumIntegerWithCommandLine integer1 integer2 ...
imp3w:Desktop mengjunhu$ java MaximumIntegerWithCommandLine 3 100 56 8
The maximum integer is: 100
```

Hint: You need to invoke proper method to convert String to int.