Midterm 2 Solution

CS 1323, Spring 2016

Name (printed legibly):

Student number:

Do not write anything on the back of any page. Tests are scanned for grading. Work on the back of the page will be lost.

If you need additional space, use the blank pages at the end of the examination. If you pull your examination apart, put your examination pages in order when you turn them in.

Answer all programming questions in Java.

Pay careful attention to what is requested: a code fragment (a few lines of code), a method, or a program. Students who write a whole program when a code fragment is requested will waste so much time that they may not complete the examination.

Unless otherwise indicated, each part of a problem is worth the same number of points. Show your work to receive partial credit.

Since you do not have the whole API available during the examination, it is acceptable to guess at method names, parameters and return values. If your guesses are reasonable—even if not perfect—you will receive credit. For example, if you forget that the String class has a length() method and call it size(), that is fine. If, however, you make up new methods that are not reasonable for the class or that magically solve problems the class cannot solve, you will not get credit.

You do not need to import packages or throw exceptions in any methods.

Integrity Pledge

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.

Signature:

1. (15 points; 3 points each) Find the logical value (true or false) of each of the statements below. If the code is not legal, say so. Assume the variables are declared and initialized as follows:

```
int[] data = {5, 9, 1, 2, 4};
int size = 200;
int maxSize = 300;
```

a) size > 100 && size < maxSize

200 > 100 && 200 < 300
true && true
true

b) data[1] < data[2] && ! data[2] < data[3]

9 < 1 && ! 1 < 2
Illegal because ! cannot be applied to an int

c) 3 <= data.length || data[1] == 9

3 <= 5 || 9 == 0
true || true
true

d) ! (size > data.length)

!(200 > 5)
! false
true

e) data[1] < data[2] && data[2] < data[3] || data[3] < data[4]

9 < 1 && 1 < 2 || 2 < 4
false && true || true
false || true
true

2. (20 points) Trace the loop below in the table.

```
int[] data = {1, 3, 6, 4, 2};
int sum = 0;
for (int index=0; index<data.length; ++index)
{
            for (int target = index; index < data.length && data[index] <= data[target]; ++target)
            {
                              sum = sum + 1;
            }
}
```

index	**data[index]**	**target**	**data[target]**	**sum**
				0
0	1	0	1	1
		1	3	2
		2	6	3
		3	4	4
		4	2	5
		5		
1	3	1	3	7
		2	6	8
		3	4	9
		4	2	
2	6	2	6	10
		3	4	
3	4	3	4	11
		4	2	
4	2	4	2	12
		5		
5				

3. (10 points; 5 points each) The paragraphs below describe static methods. Determine the signature of the method **ONLY**. ***<u>Do not write the method(s).</u>***

The signature of a method is the return type, the method name, and the parameters.

Write the signature for a method that takes two arrays of Strings and joins them together into a single array. If the first array contained “a”, “b”, and “c” and the second contained “d”, “b” the resulting array would contain “a”, “b”, “c”, “d”, “b”.

a) Assume all of the String arrays are perfect size.

String[] concatenate(String[] first, String[] second)

b) Assume the original arrays are oversized, but the array result is perfect size

String[] concatenate(String first, int firstSize, String second, int secondSize)

4. (10 points) Show the contents of the three data elements in the main program when the program finishes execution. You may use a memory diagram to trace the code if you wish to, or may use your knowledge of passing by value and/or sharing. Show or explain your work to get partial credit.

```
public class TraceMe {

      public static void main(String[] args) {
            int[] array = {9, 8, 7, 6, 5};
            int value = 5;
            int size = 3;
            array = method(size, value);

      }
      public static int[] method(int s, int v){
            s = s + 1;
            v = v - 1;
            int[] data = new int[s];
            for (int j=0; j< data.length; ++j)
                  data[j] = 2*v;
            return data;
      }
}
```

```
array contains (6 points):{8, 8, 8, 8}
value contains (1 point):5
size contains (3 points):3
```

The tables below are for your use if you wish to trace the method to determine what is in array. You do not have to use these tables if you do not wish to, although it is one way to gain partial credit.

Main stack frame

Identifier	Address	Contents
array	100	~~1000~~ 1006
	101	
	102	
	103	

Heap

Identifier	Address	Contents
0	1000	9
1	1001	8
2	1002	7
3	1003	6
4	1004	5
length	1005	5
0	1006	~~0~~ 8
1	1007	~~0~~ 8
2	1008	~~0~~ 8
3	1009	~~0~~ 8
length	1010	4
	1011	
	1015	

Method stack frame

Identifier	Address	Contents
s	200	~~3~~ 4
v	201	~~5~~ 4
data	202	1006
	203	
	204	

5. (15 points) **Work either part a) or part b) of this problem.** You do not need to work both parts, and will not get extra credit for doing so. You must use the proper algorithm. If you fill in both parts and don't indicate which you want to count, I will grade part a).

a) Trace ***selection sort*** on the array below. ***Show each swap in a separate line of the table*** and show only the data that are moved.

2	**5**	**4**	**1**	**3**
1			2	
	2		5	
		3		4
			4	5

b) Trace ***insertion sort*** on the array below. ***Show each assignment of data on a separate line*** of the table, in the order that the assignment statements are done. Show only the data that are moved.

2	**5**	**4**	**1**	**3**	**aux**
					5
					4
		5			
	4				
					1
			5		
		4			
	2				
1					
					3
				5	
			4		
		3			

6. (40 points; 10 points each for a), b) and 20 points for c))

Write a program that helps the Oklahoma Sooners basketball team calculate personal statistics for a game. Lately everyone is basketball has been talking about a "triple double." Basketball statistics are kept in the following categories[1]: points (1, 2, or 3), rebound, steal, assist, block

A player has a triple double when there are three of more categories from those above that have two digits (i.e. 10 or more).

The activity of the game is stored in a file, with example contents shown below (events of the game, one to a line with points being a number (only 1, 2, and 3 are permitted scores in basketball)):

```
rebound
3
steal
2
3
block
3
rebound
2
steal
```

This would mean the summary of the game is as below:

```
13 point(s)
2 rebound(s)
2 steal(s)
0 assist(s)
1 block(s)
```

So the program should print out:

```
No triple double.
```

This program will use an oversized array that will contain 500 elements.

Do not write code on this page or on the back of any page.

<u>You must write the methods exactly as they are given. You may not change the methods, parameters, or return type. You must use the methods you write in a) and b) in the main program in c).</u>

[1] https://en.wikipedia.org/wiki/Double_%28basketball%29

a) This method should open the file stored in the main project directory and called fileName. The data in the file should be read and put in the array. The number of elements stored in the array should be returned.

```
public static int readFile(String[] result, String fileName) throws
FileNotFoundException
{
	Scanner file = new Scanner(new File(fileName));

	int index = 0;
	while (file.hasNextLine() && index < result.length)
	{
		result[index] = file.nextLine();
		++index;
	}
	file.close();
	return index;
}
```

b) This method should take the array of data that was read from the file and analyze it and return the number of times that the target String occurred in the first size positions in the array.

```
public static int countTarget(String[] data, int size, String target)
{
      int count = 0;
      for (int index = 0; index <size; ++index)
      {
            if (data[index].equals(target))
                  ++count;
      }
      return count;
}
```

a) Write the main program using the methods in a) and b).

The method signatures from a) and b) are:
public static int readFile(String[] result, String fileName)
public static int countTarget(String[] data, int size, String target)

You may also use the method with the signature below ***without writing it***. It returns whether or not the player scored a triple double based on the number of points, rebounds, etc. that were found in the file.
public static boolean tripleDouble(int points, int rebounds, int steals, int assists, int blocks)

```java
public static void main(String[] args) throws FileNotFoundException
{
	final int SIZE = 500;
	String[] data = new String[SIZE];
	int dataSize = readFile(data, "Basketball.txt");

	int points = countTarget(data, dataSize, "1");
	points += countTarget(data,dataSize,"2") * 2;
	points += countTarget(data,dataSize,"3") * 3;
	int rebounds = countTarget(data, dataSize, "rebound");
	int blocks = countTarget(data, dataSize, "block");
	int assists = countTarget(data, dataSize, "assist");
	int steals = countTarget(data,dataSize, "steal");

	if (tripleDouble(points, rebounds, blocks, assists, steals))
	{
		System.out.println("Triple double!");
	}
	else
	{
		System.out.println("No triple double.");
	}
}
```