Midterm 3 Solution

CS 1323, Spring 2016

Name (printed legibly):

Student number:

Do not write anything on the back of any page. If you need extra space use the blank page at the back of the examination. If you pull the test apart, please put all of the pages in order when you hand your exam in.

Answer all programming questions in Java. Show your work to receive partial credit.

Pay careful attention to what is requested: a code fragment (a few lines of code), a method, or a program.

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. 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, unless asked to do so.

Relevant parts of the API are given in a separate document.

Integrity Pledge

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

Signature:

1. (20 points; 4 points each)

a) What does the String object command contain after the following code is executed?

String command = new String(“Right turn”);
command = command.substring(0,5); // Read the API on the attached handout

“Right”

b) What does the StringBuilder object command contain after the following code is executed?

StringBuilder command = new StringBuilder(“Left turn”);
command.substring(0, 5); // Read the API on the attached handout—do not assume you know it!

“Left turn”

c) Suppose that there were a method in the ArrayList class that creates/changes an ArrayList<String> object so that it contains n copies of the given String source. For the method signature below, determine whether the method is a class method (static) or an instance method and ***circle the appropriate answer***

ArrayList<String> copies (int n, String source)

class method instance method

d) What will be printed out by the code fragment below?

```
ArrayList<Integer> scores = new ArrayList<Integer>(20);
scores.add(new Integer(5));       scores.add(new Integer(7));
scores.add(new Integer(5));       scores.add(new Integer(9));
System.out.println(Collections.frequency(scores, new Integer(5));
```

2

e) What will be printed out by the code fragment below?

```
ArrayList<String> names = new ArrayList<String>();
names.add(“Bob”);
names.add(“Carmine”);  names.add(0, “Angel”);
names.add(“Jamal”);      names.add(“Rui”);
if (Collections.binarySearch(names,”Angel”) >0) // Read API and code carefully
        System.out.println(“Angel was not found”);
else
        System.out.println(“Angel was found”);
```

Angel was found

2. (20 points; 5 points each variable) Find the value of the variables below at the end of execution of the main program. You may do this using a memory diagram in the space provided, however the question will be graded only on the correctness of the answers.

```
import java.util.ArrayList;

public class Question2 {
	public static void main(String[] args) {
		String first = new String("XYZ");
		StringBuilder second = new StringBuilder("MNO");
		ArrayList<String> third = new ArrayList<String>(4);
		third.add("GHI");
		ArrayList<String> fourth;
		fourth = method(first, second, third);
		// What are the values here?
	}
	public static ArrayList<String> method(String s, StringBuilder sb,
			ArrayList<String> list) {
		s = s.toLowerCase();
		sb.append(s);
		list.add("JKL");
		list = new ArrayList<String>(4);
		list.add("ABC");
		list.add("DEF");
		return list;
	}
}
```

```
first contains: "XYZ"
second contains:  "MNOxyz"
third contains: {"GHI","JKL"}
fourth contains: {"ABC", "DEF"}
```

main stack frame

Identifier	Address	Contents
first	100	1000
second	101	1001
third	102	1003
fourth	103	1009
	104	
	105	
	106	

stack frame for method

Identifier	Address	Contents
s	200	~~1000~~ 1008
sb	201	1001
list	202	~~1003~~ 1009
	203	
	204	
	205	

Heap

Identifier	Address	Contents
	1000	XYZ
	1001	MNOx
	1002	yz
0	1003	GHI
1	1004	JKL
2	1005	
3	1006	
size	1007	~~1~~ 2
	1008	xyz
0	1009	ABC
1	1010	DEF
2	1011	
3	1012	
size	1013	2

3. (20 points; 4 points for a), b) and c); 4 points for d) and e))

Use the documentation for the TimeStamp class to answer the following questions and perform the following tasks.

In parts a) to c), if there are no methods of the required type, say "None." If there is only one, list that method name.

a) List two accessor methods in the TimeStamp class.

after, before, compareTo, equals, getTime, toString,

b) List two mutator methods in the TimeStamp class.

setTime

c) List two class methods in the TimeStamp class.

valueOf

d) Given two TimeStamp object references (finish and deadline), write a code fragment that prints out to the console "Finished before deadline" or "Finished after deadline" depending on whether the time finish came before or after the time deadline.

```
if (finish.before(deadline))
{
    System.out.println("Finished before deadline");
}
else
{
    System.out.println("Finished after deadline");
}
```

e) Write a code fragment that creates a TimeStamp object from a String object timeString that contains "2016-04-21 13:30:00"

```
TimeStamp t = TimeStamp.valueOf(timeString);
```

4. (40 points; 10 points each for a) and b), 20 points for c))

You've been asked to write a program to help the City of Flint, Michigan analyze lead in their drinking water. The data is stored in files where each line is in the following format:

<amount of lead in water sample> <name of resident—may be any number of words>

A sample file is shown below:

```
50.5 Ed Kurtz
71.2 Dayne Walling
NaN Jenna Walling Kurtz
29.3 Rick Snyder
42.9 Dayne Kurtz
```

The amount of lead is generally stored as a floating point value. However, sometimes the testing equipment fails and returns NaN (for Not a Number).

Your program should allow the following operations:

- Retrieve all of the lines that contain a given search String. The search String could be someone's full or partial name, a numeric value stored in a String, or NaN. The result of one search should be the input to the next search.
- Count the number of samples with values in a given range

You may choose to write this program using either an array of Strings, or an ArrayList<String>. You must be consistent—so if you use an ArrayList<String> in part a), you must also use that in parts b) and c).

The method(s) below are available for your use. You do not have to write either of these methods, and should use only one of the two methods. These methods take the filename and read the contents of the file into an oversized String[] or ArrayList<String>.

public static ArrayList<String> readData(String filename)
public static int readData(String[] data, String filename)

There also is a method that prints out the menu and accepts user input. You do not need to write this method either.

public int menu(Scanner keyboard)

A sample run of the program is given below. The file named “Data.txt” was shown on the previous page. The comment “// Menu is repeated” occurs whenever the four item menu would be shown. User data entries are underlined. I printed out the contents of the array or ArrayList after each operation so you could see the results as they progressed.

```
1. Choose a new file to analyze
2. Search the existing data
3. Count number of values in range
4. Quit
1
Enter the name of the file that contains the data
Data.txt
[50.5 Ed Kurtz, 71.2 Dayne Walling, NaN Jenna Walling Kurtz, 29.3 Rick
Snyder, 42.9 Dayne Kurtz]

// Menu is repeated
2
Enter the target
Kurtz
[50.5 Ed Kurtz, NaN Jenna Walling Kurtz, 42.9 Dayne Kurtz]

// Menu is repeated
3
Enter the minimum and maximum values (inclusive)
40.0 45.0
There were 1 values in that range

// Menu is repeated
1
Enter the name of the file that contains the data
Data.txt
[50.5 Ed Kurtz, 71.2 Dayne Walling, NaN Jenna Walling Kurtz, 29.3 Rick
Snyder, 42.9 Dayne Kurtz]

// Menu is repeated
2
Enter the target
Dayne
[71.2 Dayne Walling, 42.9 Dayne Kurtz]

// Menu is repeated
2
Enter the target
Kurtz
[42.9 Dayne Kurtz]

// Menu is repeated
3
Enter the minimum and maximum values (inclusive)
40.0 50.0
There were 1 values in that range

// Menu is repeated
4
Goodbye
```

a) Write one of the two methods below.

public static ArrayList<String> fullOrPartialMatch(ArrayList<String> source, String target)

public static int fullOrPartialMatch(String[] source, int sourceSize, String target)

```
public static ArrayList<String> fullOrPartialMatch(ArrayList<String> source,
String target)
	{
		ArrayList<String> result = new ArrayList<String>();

		for (int i=0; i<source.size(); ++i)
		{
			if (source.get(i).contains(target))
				result.add(source.get(i));
		}
		return result;
	}
```

b) Write one of the two methods below.

public static int countRange(ArrayList<String> list, double min, double max)

public static int countRange(String[] list, int listSize, double min, double max)

```
public static int countRange(ArrayList<String> list, double min, double max)
	{
		int count = 0;
		for (int i=0; i<list.size(); ++i)
		{
			String line = list.get(i);
			Scanner s = new Scanner(line);
			double value;
			if (s.hasNextDouble())
			{
				value = s.nextDouble();

				if (value >= min && value <= max)
					++count;
			}
			s.close();
		}
		return count;
	}
```

c) Write the main program. You must use the methods you wrote in a) and b). The complete set of method signatures is below.

public static ArrayList<String> readData(String filename)
public static int readData(String[] data, String filename)
public int menu(Scanner keyboard)
public static ArrayList<String> fullOrPartialMatch(ArrayList<String> source, String target)
public static int fullOrPartialMatch(String[] source, int sourceSize, String target)
public static int countRange(ArrayList<String> list, double min, double max)
public static int countRange(String[] list, int listSize, double min, double max)

```
public static void main(String[] args) throws FileNotFoundException
	{
		Scanner keyboard = new Scanner(System.in);

		int choice = menu(keyboard);
		ArrayList<String> current = new ArrayList<String>();

		while (choice != 4)
		{
			if (choice == 1)
			{
				System.out.println("Enter the name of the file "
					+ " that contains the data");
				String fileName = keyboard.nextLine();
				current = readFile(fileName);
				System.out.println(current);
			}
			else if (choice == 2)
			{
				System.out.println("Enter the target");
				String target = keyboard.nextLine();
				current = fullOrPartialMatch(current, target);
				System.out.println(current);
			}
			else if (choice == 3)
			{
				System.out.println("Enter the minimum and maximum "
					+ " values (inclusive)");
				double min = keyboard.nextDouble();
				double max = keyboard.nextDouble();
				keyboard.nextLine(); // get rid of that pesky newline
				int range = countRange(current, min, max);
				System.out.println("There were " + range
					+ " values in that range");
			}

			choice = menu(keyboard);
		}

		System.out.println("Goodbye");

	}
```