CS 312 – Exam 2 – Fall 2016

Your Name____________________________________

Your UTEID ______________________________________

Circle your TA's Name: Aish Linh Shelby

Sonika CK Omer

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Topic</th>
<th>Points Possible</th>
<th>Points Off</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>code trace</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>program logic</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>scanners</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>strings</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>arrays</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>arrays and strings</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>strings</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL POINTS OFF:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCORE OUT OF 120:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instructions:
1. You have 2 hours to complete the test.
2. You must use a pencil on the exam.
3. You may not use a calculator or any other electronic device.
4. When code is required, write Java code. Ensure you follow the restrictions of the question. Limit yourself to the features from chapters 1 - 7 of the book and topics 1 - 25 in class.
5. You may break problems up into smaller methods. (In other words you can add helper methods.)
6. The proctors will not answer questions. If you believe there is an error or a question is ambiguous, state your assumptions and answer based on those assumptions.
7. When you finish, show the proctor your UTID, turn in the exam and all scratch paper.
1. Evaluating Code. 30 points, 2 points each. Assume all necessary imports have been made. If the snippet contains a syntax error or compiler error, answer syntax error. If the snippet results in a runtime error or exception answer runtime error. If the code results in an infinite loop answer infinite loop.

A. What is output by the following code?

```java
String a1 = "Chelsea";
String a2 = a1.substring(1, 4);
System.out.print(a2 + " " + a1.length());
```

Output: __________________________

B. What is output by the following code?

```java
String b1 = "longhorns";
String b2 = b1.substring(4); // original version said a1.substring(4)
System.out.print(b2 + " " + b1.charAt(2));
```

Output: __________________________

C. What is output by the following code?

```java
String c1 = "cs";
String c2 = "ut";
String c3 = "utcs";
String c4 = c2 + c1;
System.out.print(c3 == c4);
```

Output: __________________________

D. Are the two boolean expressions below logically equivalent? In other words given the same inputs do the two expressions always evaluate to the same boolean result? x, y, and z are int variables.

Expression 1: (x < y) || (y == z)
Expression 2: !((x < y) && (y != z))

Answer: __________________________

E. Given the following expression, how many of the 8 combinations of values for e1, e2, and e3 (all boolean variables) cause the expression to evaluate to false?

```java
(e1 || !e2) && e3
```

Answer: __________________________
F. What is output by the following code?

```java
String f1 = "BDM8";
f1.toLowerCase();
f1 += "***";
System.out.print(f1);
```

Output: __________________________

G. What is output by the following code assuming the user types in the following:

```
12  ONE
```

```java
Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
int y = sc.nextInt();
System.out.print(x + " " + y);
```

Output: __________________________

H. What is output by the following code? For this question only use an underscore character, _, to indicate any spaces in the output. One underscore per space.

```java
double h1 = 12.37;
System.out.printf("val = %6.3f", h1);
```

Output: __________________________

I. What is output by the following code?

```java
int[] i1 = new int[5];
System.out.print(i1.length + " " + i1[3]);
```

Output: __________________________

J. What is output by the following code?

```java
int[] data1 = {5, -3, 7, 2, 4};
int j1 = 3;
data1[j1] += data1[j1] + j1;
System.out.println(Arrays.toString(data1));
```

Output: __________________________
K. What is output by the following code?

```java
int[] k1 = {5, 3, 2, -1};
int x2 = 3;
int y2 = x2 * 2;
if (k1[y2] < 5 && y2 < k1.length) {
    k1[1] = 12;
}
System.out.print(Arrays.toString(k1));
```

Output: __________________________

L. What is output by the following code?

```java
int[] ar = {5, 3, 6, 2};
ar[1] += 2;
methodL(ar);
ar[2] -= 3;
System.out.print(Arrays.toString(ar));
```

```java
public static void methodL(int[] data) {
    data[1] += 2;
}
```

For parts M, N, and O consider the following method:

```java
public static int[] create(int val) {
    int[] result = new int[val + 2];
    for (int i = 0; i < result.length; i++) {
        result[i] = val * i + i * i;
    }
    return result;
}
```

```
// Output for L: ______________________________
```

M. What is output by the following code?

```java
int[] m = create(-2);
System.out.print(Arrays.toString(m));
```

Output: ______________________________

N. What is output by the following code?

```java
int[] n = create(1);
System.out.print(Arrays.toString(n));
```

Output: ______________________________

O. What is output by the following code?

```java
System.out.print(Arrays.toString(create(3)));
```

Output: ______________________________
2. Program Logic - 15 Points. Consider the following method. For each of the four points labeled by comments and each of the four assertions in the table, write whether the assertion is *always* true, *sometimes* true, or *never* true at that point in the code. Abbreviate *always* with an A, *sometimes* with an S and *never* with an N.

```java
public static void assertionPractice() {
    Random r = new Random();
    int y = 10;
    int z = 0;
    // POINT A
    while (y != 0) {
        // POINT B
        y = r.nextInt(23);
        if (y % 4 == 0) {
            z++;
            // POINT C
            y--;
        }
        // POINT D
        y--;
    }
    // POINT E
    System.out.println("z = " + z);
}
```

Abbreviate *always* with an A, *sometimes* with an S and *never* with an N.

<table>
<thead>
<tr>
<th></th>
<th>z == 0</th>
<th>y == 0</th>
<th>y % 4 == 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>POINT A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. **Scanners. 15 points.** Write a complete method `ratioOfIntsToDoubles`. The method accepts a `Scanner` already connected to a file. The method returns the ratio of the number of integers in the file to the number of doubles in the file.

If there are no doubles in the file the method shall return `-1.0`.

**A token that could be read as an int or a double shall be read as an int.** For example the token **12** could be read as an int or a double, but your method shall read it as an int.

For example, if the `Scanner` were connected to the following file:

```
Data data data 12 5       I need data
One cannot make bricks ( 0.3 ) without clay! 12
You are 0 (<- that's a zero, not an oh.) help.
17    -2.5     -12
Last line of data with two numbers  0.5    37.14
```

The file in the example has 6 integers (12 5 12 0 17 -12) and 4 doubles (0.3 -2.5 0.5 37.14).

Given that file the method would return 1.5. \( \frac{6}{4} = 1.5 \)

You may use the methods from the `Scanner` class. Do not use any other Java classes or methods.

Do not use arrays.

**Complete the method on the next page.**
public static double ratioOfIntsToDoubles(Scanner sc) {

4. Strings - 15 Points. Create a method `getStretchedString` that accepts two parameters, a String and an int. The method creates and returns a new String with each character repeated the given number of times.

Examples of calls to `stretchString`:

getStretchedString ("abc", 3) returns "aaabbbccc"
getStretchedString ("", 3) returns ""
getStretchedString ("Example 3", 2) returns "EExxaammppllee  33"
getStretchedString ("aaDa", 4) returns "aaaaaaaaDDDDaaaaa"

Assume the int parameter `num` is greater than 1. (`num > 1`)

You may use String concatenation, the `length()` and `charAt()` methods from the String class.

Do not use any other Java methods or classes.

Complete the method on the next page.
public static String getStretchedString (String str, int num) {

5. Arrays 14 Points. Write a method `numLessThanPrevious`. The method has one parameter: an array of `int`s. The method returns the number of elements in the array that are less than the element just preceding it in the array.

Examples of results given various arrays:

`[]` returns 0
`[1]` returns 0
`[2, 1]` returns 1 (1 < 2)
`[1, 5, 10, 12, 37]` returns 0
`[1, 1, 1, 1, 1]` returns 0
`[5, 2, 0, -12, -101, 10, 5]` returns 5
(2 < 5, 0 < 2, -12 < 0, -101, -12, 5 < 10)

You may not use any other Java classes or methods in your answer.

Do not create any new arrays.

The parameter is unaltered by this method.

Complete the method on the next page.
public static int numLessThanPrevious (int[] data) {

6. Arrays 15 Points. Write a method `noDuplicates` that accepts one parameters, an array of `String` variables. The method returns `true` if each `String` in the array appears exactly once in the array. In other words the method returns `true` if there are no duplicate `Strings` in the array.

Examples of results given various arrays:

[] returns true

["the"] returns true

["the", "The"] returns true

["a", "aa", "b", "the", ",", "REM"] returns true

[",", "FYC", ",", "the", "THE"] returns false, "," repeated

["the", "the", "the", "the", "the"] returns false

["a", "aa", "b", "the", ",", "REM", "aa", "ELO"] returns false, "aa" repeated

You may assume the array sent to the method DOES NOT contain any elements equal to `null`.

You may use the `equals` methods from the `String` class, but no other Java classes or methods.

Do not alter the original array.

Do not create any new arrays.

You method should not do any unnecessary work. In other words it should be as efficient as possible given the constraints of the question.

Complete the method on the next page.
public static boolean noDuplicates(String[] data) {


7. Strings  15 Points. Write a method `matchingChars`. The method accepts two Strings and a char as parameters. The method returns true if the two Strings contain the given character in the same place for all occurrences of the given character.

Examples of `matchingChars(String s1, String s2, char ch)`

<table>
<thead>
<tr>
<th>s1</th>
<th>s2</th>
<th>ch</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;dad day&quot;</td>
<td>&quot;cab sad&quot;</td>
<td>'a'</td>
<td>true</td>
</tr>
<tr>
<td>&quot;dad day&quot;</td>
<td>&quot;cab&quot;</td>
<td>'a'</td>
<td>false, no match for second 'a' in s1</td>
</tr>
<tr>
<td>&quot;dad&quot;</td>
<td>&quot;cab also&quot;</td>
<td>'a'</td>
<td>false, no match for second 'a' in s2</td>
</tr>
<tr>
<td>&quot;dad day&quot;</td>
<td>&quot;cab sad&quot;</td>
<td>'a'</td>
<td>true, there are no '*'s so all occurrences match</td>
</tr>
<tr>
<td>&quot;that old band ABBA&quot;</td>
<td>&quot;that new band abba??&quot;</td>
<td>'a'</td>
<td>false, case sensitive</td>
</tr>
<tr>
<td>&quot;example<strong>with some</strong><em>non</em>letters*&quot;</td>
<td>&quot;but has<strong>spacesins</strong><em>ome</em>places!*** !!&quot;</td>
<td>'*'</td>
<td>true</td>
</tr>
</tbody>
</table>

You may use the String `charAt()` and `length()` methods.

You may not use any other Java classes or methods.

Complete the method on the next page.
public static boolean matchingChars(String s1, String s2, char ch) {