

EECS2030Z Test 2

Version C

GETTING STARTED

1. Start eclipse; use the workspace suggested by eclipse (or remember the directory name of your workspace).
2. Import the test project by doing the following:
 1. Under the **File** menu choose **Import...**
 2. Under **General** choose **Existing Projects into Workspace** and press **Next**
 3. Click the **Select archive file** radio button, and click the **Browse...** button.
 4. Select the file **test2C.zip** and click **OK**
 - If you do not see the file named **test2C.zip** in your directory, then open a terminal and copy and paste the following command:

```
cp /eecs/dept/www/course/2030/labtest/test2C.zip .
```

and re-import the project.

5. Click **Finish**.
 3. All of the files you need for this test should now appear in eclipse.
 4. Open a terminal. You will use this terminal to submit your work.
 5. Copy and paste the command `cd workspace/Test2C/src/test2` into the terminal and press enter.
-

Question 1 (20 marks total)

[SOLUTION](#)

Implement [the utility class described by this API](#). You do not have to include javadoc comments.

```
submit 2030 test2C Utility2C.java
```

Question 2 (10 marks total)

[SOLUTION](#)

Implement [the constructors for the class described by this API](#). Use constructor chaining where possible when implementing your constructors. You do not have to include javadoc comments in your code, but see Question 3C below.

```
submit 2030 test2C Range.java
```

Question 3 (20 marks total)

A.

State the definition of a method postcondition.

A method postcondition is a condition that the method promises will be true immediately after the method finishes running.

Grading scheme: 3 marks total.

B.

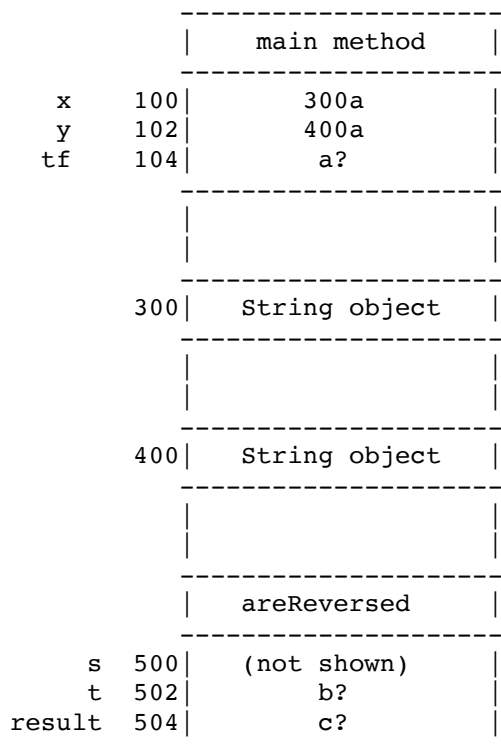
Suppose that the method named `areReversed` from Question 1 was implemented like so:

```
public static boolean areReversed(String s, String t) {
    boolean result;
    /* code not shown here that assigns the correct value to result */
    return result;
}
```

Suppose a client writes a main method that includes the following two lines of Java code:

```
String x = "xyz";
String y = "zyx";
boolean tf = Utility2C.areReversed(x, y);
```

The memory diagram illustrating the state of memory for the three lines of client code is shown below. What suitable values of a, b, and c would complete the memory diagram?



a = true, b = 400a, c = true

a and c are both equal to the return value of `areReversed`. b is equal to the address of the second string.

Grading scheme: 3 marks total (1 for each of a, b, and c).

C.

Provide the Javadoc necessary to *exactly* reproduce the API documentation for the constructor `Range(int min, int max)` from Question 2.

```
/**
 * Initializes the range to have the given minimum and maximum
 * values.
 *
 * @param min the minimum value of the range
 * @param max the maximum value of the range
 * @throws IllegalArgumentException if min is greater than
 * or equal to max
 */
```

Grading scheme: 4 marks total (1 for method description, 1 for the the first parameter description, 1 for the second parameter description, 1 for the exception). The answer must appear exactly as shown above (missing or extra white space is acceptable, and the leading * are not required.).

D.

Provide 3 test cases for the method `utility2C.areReversed`. Make sure that each test case tests a different feature of the method; try to include one boundary test case. For each test case, provide a one sentence explanation of what the test case is testing.

The boundary case occurs when both strings are empty.

```
s          : ""
t          : ""
expected return value: true
explanation  : tests strings of zero length
```

```
s          : "and"
t          : "dna"
expected return value: true
explanation  : tests two strings that are the reverse of each other
```

```
s          : "ate"
t          : "tea"
expected return value: false
explanation  : tests two strings that are not the reverse of each other
```

```
s          : "a"
t          : "and"
expected return value: IllegalArgumentException
explanation  : tests that an exception is thrown if strings have different lengths
```

Grading scheme: 6 marks total.

2 marks for each test case (1 mark for providing appropriate inputs, 0.5 mark for providing the expected return value or result, 0.5 mark for the explanation). Deduct 1 mark if there are 3 valid test cases that do include the boundary case.

E.

Consider the following implementation of `equals(Object)` for the class `Range` from Question 2:

```
public class Range {  
  
    private int min;  
    private int max;  
  
    // constructors not shown  
  
    @Override  
    public boolean equals(Object obj) {  
        if (this == obj) {  
            return true;  
        }  
        if (obj == null || this.getClass() != obj.getClass()) {  
            return false;  
        }  
        Range other = (Range) obj;  
        if (this.min - other.min == 0) {  
            return true;  
        }  
        if (this.max - other.max == 0) {  
            return true;  
        }  
        return false;  
    }  
}
```

Explain whether or not the implementation shown above is correct. If it is not correct then identify all of the errors in the implementation.

The implementation is not correct. The error is that the implementation returns `true` if the min of the two ranges are equal without also checking that the max of the two ranges are also equal.

It looks like there are also two overflow errors, but these don't actually affect the returned result (i.e., it is impossible to find two integer values that are not equal whose difference equals zero).

Grading scheme: 4 marks total.

2 marks for stating `x.equals(x) == true` is not always true. 2 marks for the explanation. Award 1 mark if overflow is mentioned.

submit 2030 test2C answers.txt