

CS 312 – Exam 2 – Fall 2017

Your Name _____

Your UTEID _____

Circle your TA's Name: Aish Anthony Bri Carla Chris
 CK Dayanny Fatima Hailey Omer

Problem Number	Topic	Points Possible	Points Off
1	code trace	28	
2	program logic	8	
3	strings	10	
4	strings	10	
5	arrays	10	
6	programming	16	
7	scanners	18	
TOTAL POINTS OFF:			
SCORE OUT OF 100:			

Instructions:

1. You have 2 hours to complete the exam.
2. You must use a pencil on the exam.
3. You may not use a calculator or any other electronic devices.
4. When code is required, write Java code. Limit yourself to the features from chapters 1 - 7 of the book and topics 1 - 23 in class.
5. Ensure you follow the restrictions of the question.
6. You may write and call your own helper methods.
7. 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.
8. When you finish, show the proctor your UTID, turn in the exam and all scratch paper.

1. Evaluating Code. 28 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?

```
String a1 = "Longhorns";  
String a2 = a1.substring(5);  
System.out.print(a1.length() + " " + a2);
```

Output: _____

B. What is output by the following code?

```
String b1 = "ABBA";  
String b2 = "" + b1.charAt(1) + b1.charAt(3) + b1.charAt(3);  
System.out.print(b1 + " " + b2);
```

Output: _____

C. What is output by the following code?

```
String c1 = "REM_SLEEP";  
String c2 = c1.substring(0, c1.indexOf("_"));  
c1 = "REM";  
System.out.print(c1 == c2);
```

Output: _____

D. What is output by the following code?

```
String d1 = "THEORY_SYSTEMS";  
System.out.print(d1.indexOf("E") + " " + d1.indexOf("P") + " "  
                + d1.contains("STEM"));
```

Output: _____

E. For the following code what are the unique possible values x can store after the code completes?
List each one in ascending order. Do not list repeat values.

```
Random re = new Random();  
int x = ((re.nextInt(5) - 3) * 2) / 3;
```

Answer: _____

F. 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? p , q , and r are boolean variables.

Expression 1: $!(p \ || \ !q) \ \&\& \ !r$

Expression 2: $q \ \&\& \ !(p \ || \ r)$

Answer: _____

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

17 0.5 1.5

```
Scanner sc = new Scanner(System.in);
double a = sc.nextDouble();
sc.nextDouble();
double b = a + sc.nextDouble();
System.out.print(a + " " + b);
```

Output: _____

H. What is output by the following code?

```
String h1 = "FORTY_ACRES";
String h2 = h1.substring(1, 7).substring(2, 4);
System.out.print(h2);
```

Output: _____

I. Given the following expression, how many of the 8 combinations of values for $i1$, $i2$, and $i3$ (all boolean variables) cause the expression to evaluate to true?

$i1 \ || \ i2 \ || \ i3$

Answer: _____

J. What is output by the following method assuming sc is connected to a file with the following data:

12 13 CS

UT

```
public static void j(Scanner sc) {
    String s1 = sc.next();
    String s2 = sc.next();
    String s3 = sc.next();
    System.out.print(s1 + s2 + s3);
}
```

Output: _____

K. 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.

```
double k1 = 2.718;
System.out.printf("e=%5.2fish", k1);
```

Output: _____

L. What is output by the following code?

```
String str1 = "BASIC";
methodL(str1);
System.out.print(str1);

public static void methodL(String str1) {
    str1 = str1.substring(1, 3);
    System.out.print(str1 + " ");
}
```

Output: _____

M. What is output by the following code?

```
int[] m = new int[3];
m[1] = 5;
methodM(m);
System.out.print(Arrays.toString(m));

public static void methodM(int[] m) {
    m[0]++;
    m[2] = m[1] + m[0];
}
```

Output: _____

N. What is output by the following code?

```
int[] n = {2, 5, 1, 3};
methodN(n);
System.out.print(Arrays.toString(n));

public static void methodN(int[] n) {
    n[2] += n[n[2]] + n.length;
    n = new int[2];
    n[1] = 3;
}
```

Output: _____

2. Program Logic - 8 Points. Consider the following method. For each of the five points labeled by comments and each of the three assertions in the table, write whether the assertion is *always* true, *sometimes* true and sometimes false, or *never* true at that point in the code. Abbreviate *always* with an A, *sometimes* with an S and *never* with an N.

```
public static int mystery(int z) {
    int y = 10;
    int x = 0;
    // POINT A
    while (z >= y) {
        // POINT B
        if (x <= y) {
            // POINT C
            x++;
        }
        z -= 10;
        // POINT D
    }
    // POINT E
    return z;
}
```

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

	$x \leq y$	$y > z$	$x == 11$
POINT A			
POINT B			
POINT C			
POINT D			
POINT E			

3. Strings - 10 Points. Write a method that determines if two Strings start with the same N characters.

The only String methods you may use are the `length()` and `charAt(int i)` methods.

Do not use any other methods from the String class or other Java classes. Do not use arrays.

Your method should return its answer as soon as possible.

The method header is

```
/* Returns true if s1 and s2 start with the same n characters, false  
   otherwise. n shall be > 0. */  
public static boolean sameStart(String s1, String s2, int n)
```

Examples of calls to the method with expected results:

```
sameStart("", "AAA", 2) -> false
```

```
sameStart("AAA", "", 2) -> false
```

```
sameStart("BABBA", "AAA", 2) -> false
```

```
sameStart("AABBBA", "AAA", 2) -> true
```

```
sameStart("AA", "AAAAAAAA", 3) -> false
```

```
sameStart("aaa", "AAA", 2) -> false
```

```
sameStart("AaaaaBB", "AAA", 1) -> true
```

```
sameStart("8272302", "eight_two_seven", 3) -> false
```

Complete the method on the next page.

```
/* Returns true if s1 and s2 start with the same n characters, false  
   otherwise. n shall be > 0. */  
public static boolean sameStart(String s1, String s2, int n)
```

4. Strings 10 Points. Write a method `reverseAndStretch` that accepts one `String` parameter. The method creates and returns a new `String` that is the reverse of the original `String` and each character is repeated based on the pattern shown below.

Examples of `reverseAndStretch(String str)`

`reverseAndStretch("")` -> returns ""

`reverseAndStretch("A")` -> returns "A"

`reverseAndStretch("it")` -> returns "tti"

`reverseAndStretch("cat")` -> returns "tttaac"

`reverseAndStretch("dads")` -> returns "ssssdddaad"

`reverseAndStretch("CS312")` -> returns "222221111333SSC"

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

You may not use any other Java classes or methods.

Complete the method on the next page.


```
public static String reverseAndStretch(String str) {
```

5. Arrays 10 Points. Write a method `getDifferenceArray`. The method accepts two parameters, both arrays of `ints`. The method returns an array of `ints` equal in length to the parameter with the smallest length. The elements in the returned array are equal to the corresponding element in the first parameter minus the corresponding element in the second parameter.

Examples of results given various arrays:

```
[12, 5, 3, 12, 6, 2] first array  
[10, 10, -5] second array  
[2, -5, 8] resulting array
```

```
[] first array (length 0)  
[10, 10, -5] second array  
[] resulting array (length 0)
```

```
[5, 6, -5, 10, 100] first array  
[5, 6, -5, 13, 50] second array  
[0, 0, 0, -3, 50] resulting array.
```

You may not use any other Java classes or methods in your answer. Not even the `Math` class,

Neither of the parameters is altered as a result of this method call.

Complete the method on the next page.

```
public static int[] getDifferenceArray(int[] ar1, int[] ar2) {
```

6. Programming. 16 points. Write a method that determine if a given digit occurs exactly a given number of times in an int. The method is named `digitPresent` and it returns a boolean. The method header is

```
public static boolean digitPresent(int num, int digit, int times)
```

You may not use any other Java or classes in your answer. Specifically, you cannot use Strings or arrays of any kind.

Your method should return its answer as soon as possible.

You may assume the given int is ≥ 0 .

You may assume the target number of times is ≥ 0 .

Here are examples of call to the method and the expected return value. Recall parameters are number, target digit, required number of times

```
digitPresent(0, 1, 1) -> returns false
```

```
digitPresent(30, 3, 1) -> returns true
```

```
digitPresent(303, 3, 1) -> returns false (3 occurs 2 times in 303)
```

```
digitPresent(0, 0, 1) -> returns true
```

```
digitPresent(0, 0, 2) -> returns false (we do not consider possible leading zeros)
```

```
digitPresent(5627292, 1, 1) -> returns false
```

```
digitPresent(5627292, 1, 2) -> returns false
```

```
digitPresent(5627292, 2, 1) -> returns false (2 occurs 3 times in 5627292)
```

```
digitPresent(5627292, 2, 2) -> returns false (2 occurs 3 times in 5627292)
```

```
digitPresent(5627292, 2, 3) -> returns true
```

```
digitPresent(5627292, 2, 4) -> returns false
```

```
digitPresent(5627292, 0, 1) -> returns false
```

```
digitPresent(70809, 0, 2) -> returns true
```

```
digitPresent(70809, 5, 0) -> returns true (5 occurs 0 times in 70809)
```

Complete the method on the next page.

```
/* Return true if digit is present exactly times times in num, false  
   otherwise. */  
public static boolean digitPresent(int num, int digit, int times) {
```

7. Scanners. 18 points. Write a complete method `linesWithWord`. The method accepts a `Scanner` already connected to a file and a target `String`.

The method prints:

- the number of times the target word appears in each line
- the total number of times the word appears
- the maximum number of times the word appeared in a single line.

Consider this example. The `Scanner` is connected to the following file and the target `String` is `"line"`. Note, the occurrences of the target `String` `"line"` are bolded here for ease of reading.

```
This is the first line and the next line is blank. How many lines? 6

Love yourself first and everything else falls into line
That's a quote by Lucille Ball. Steven Wright said:
There is a fine line between fishing and standing on the shore...
It declines from there. Line of text. Last line.
```

Given the file above and a target of `"line"` the output would be:

```
1: 2
2: 0
3: 1
4: 0
5: 1
6: 0
total: 4
max times in line: 2
```

Note, in the example the tokens `"lines"`, `"decline"`, `"line."` and `"Line"` are not matches with `"line"`.

You may assume the file has at least one line of text in it.

You may assume the target `String` has a length greater than 0.

You may use the `hasNextLine()`, `nextLine()`, `hasNext()`, and `next()` methods from the `Scanner` class and the `equals()` method from the `String` class.

You may create and use new `Scanner` objects.

Do not use any other Java classes or methods.

Complete the method on the next page.

```
public static void linesWithWord(Scanner sc, String tgt) {
```