CS 312 – Exam 1 – Fall 2019

Name: ______________________________ UTEID: ___________________

Circle your TA's Name: Andrew Anthony Carla Fatima Kate

Larry Rebecca Tim Varshinee

Problem Number	Topic	Points Possible	Points Off
1	Expressions	16	
2	Code Tracing	24	
3	Conditionals	12	
4	Programming	16	
5	Programming	12	
6	Graphics	20	
TOTAL POINTS OFF:			
SCORE OUT OF 100:			

Instructions:

1. Please turn off or silence your mobile devices.
2. You have 2 hours to complete the test.
3. Place your final answers on this test, not scratch paper. Answers must be in pencil.
4. You may not use a calculator.
5. When code is required, write Java code. You may use only features we discussed in topics 1-14, including those covered in the textbook chapters 1 through 5.
6. Follow the restrictions of the question. For example, if a question does not allow the use of Strings, do not use any Strings in your answer.
7. You may write additional methods to provide structure and remove redundancy.
8. The exam proctors will not answer questions regarding the content of the exam. If you believe a question has an error or is ambiguous, state your assumption and answer based on your assumption.
9. If you finish early bring your exam and scratch paper to the proctor and show them your UTID.

1. Expressions: 1 point each, 16 points total. For each Java expression in the left hand column, indicate the result of the expression in the right hand column.

You must show a value of the appropriate type. For example, 7.0 rather than 7 for a double and "7" instead of 7 for a String. Answers that do not indicate the data type correctly are wrong.

A. `6 * 2 - 3 * (5 - 10)` ________________________________

B. `10 / 4 * 20 / 15` ________________________________

C. `0.5 + 5 / 2 * 2.5` ________________________________

D. `100 / (20 * 20) * 703.0` ________________________________

E. `50 % 15 + 33 % (10 * 6)` ________________________________

F. `5.125 % 0.25` ________________________________

G. `10 * -5 + 100 + "GDC"` ________________________________

H. `8 - 2 + "8-2" + 8 * 2 + 8` ________________________________

I. `"" + 8 % 16 * 10 + 2` ________________________________

J. `(int) (-1.9 * 2)` ________________________________

K. `(int) 3.75 * 3 + 13 % 5` ________________________________

L. `(double) (10 / 4 * 3)` ________________________________

M. `"Java" + 3 * 4` ________________________________

N. `Math.pow(2.0, 3.0)` ________________________________

O. `Math.floor(100.5 / 30.5)` ________________________________

P. `(int) Math.random() * 4 + 2` ________________________________

2. Code tracing: 2 points each, 24 points total. Place you answer in the box to the right of the code. If the code results in a syntax error, answer **syntax error.** If the code results in a runtime error, answer **runtime error. For output show exactly what is output to the screen when the code is run.**

A. What is output by the following code when it is run?

```
int x1 = 10;
int y1 = 2 + x1 / 3;
x1 += 2;
y1--;
y1 = 3 + y1;
System.out.print(x1 + " " + y1);
```

B. What is output by the following code when it is run?

```
double a2 = 2.0;
double b2 = 3.0;
double c2 = 2.0;
b2 *= c2 * (a2 + b2);
System.out.print(b2 + " " + c2);
```

C. How many asterisks does the following code print out? **Don't show the output. Simply state the number of asterisks that are printed out when the code runs.**

```
for (int i = 0; i < 5; i++) {
     System.out.print("***");
     for (int j = 0; j < 3; j++) {
          System.out.print("*");
     }
     System.out.print("***");
     for (int j = 1; j <= 5 ; j++) {
          System.out.print("*");
     }
}
```

D. How many asterisks does the following code print out? **Don't show the output. Simply state the number of asterisks that are printed out when the code runs.**

```
for (int i = 1; i <= 10; i += 2) {
     for (int j = 0; j < i; j++) {
          System.out.print("*");
     }
}
```

E. How many asterisks does the following code print out? **Don't show the output. Simply state the number of asterisks that are printed out when the code runs.**

```
a();
b();
c();

public static void a() {
      for (int i = 1; i <= 8; i++) {
            System.out.print("*");
      }
}

public static void b() {
      for (int i = 0; i < 3; i++) {
            System.out.print("*");
            a();
            System.out.print("*");
      }
}

public static void c() {
      a();
      b();
      for (int i = 1; i <= 3; i++) {
            a();
            b();
      }
}
```

F. What is output by the following code when it is run?

```
int x6 = -5;
int y6 = 5;
f(x6, y6);
f(y6, x6);
System.out.print(x6 + " " + y6);

public static void f(int x6, int y6) {
      y6 -= 3;
      x6 *= 2;
      System.out.print(x6 + " " + y6 + " ");
}
```

G. What is output by the following code when it is run?

```
String s1 = "CS312";
String s2 = s1.substring(1, 3);
System.out.print(s1 + " " + s2);
```

H. What is output by the following code when it is run?

```
String s4 = "TOWER\\";
s4 = 1 + s4 + 1 + 2;
s4 = s4.substring(6);
System.out.print(s4 + s4.toLowerCase());
```

I. What is output by the following code when it is run?

```
int x9 = 3;
System.out.print(ii(x9 * 2) + " " + (x9 - 4) + ii(x9));

public static int ii(int x) {
     x = x * -3;
     System.out.print(x);
     x++;
     return x;
}
```

J. What is output by the following code when it is run?

```
int x5 = 2;
int y5 = 30;
while (y5 >= 4) {
     x5 *= 2;
     y5 = y5 / 2;
}
System.out.print(x5 + " " + y5);
```

K. What is output by the following code when it is run?

```
double a3 = -2.75;
double b3 = 3.751;
double c3 = -5.0;
System.out.print(Math.ceil(a3) + " " +
     Math.floor(b3) + " " + Math.ceil(c3));
```

L. What is output by the following code when it is run?

```
String s8 = "CS1";
while (s8.length() < 50) {
     s8 = s8 + s8;
}
System.out.print(s8.length());
```

3. Programming with Conditionals: 12 points Write a static method, `getSemester`. The method header is

```
public static String getSemester(int month, int day)
```

The parameters month and day represent a day in the year. For example, `month` = 6 and `day` = 30 for June 30th or `month` = 3 and `day` = 16 for March 16th.

Assume 1 <= `month` <= 12 and 1 <= `day` <= 31.

The method returns a String indicating which academic term (semester) a day falls in.

If the given day is between 1/15 and 5/20 inclusive the method returns the `String "Spring"`.

If the given day is between 5/21 and 8/25 inclusive the method returns the `String "Summer"`.

If the given day is between 8/26 and 1/14 inclusive the method returns the `String "Fall"`.

Examples of call to the method:

```
getSemester(8, 28) -> returns "Fall"

getSemester(10, 31) -> returns "Fall"

getSemester(1, 1) -> returns "Fall"

getSemester(12, 31) -> returns "Fall"

getSemester(1, 14) -> returns "Fall"

getSemester(1, 15) -> returns "Spring"

getSemester(2, 28) -> returns "Spring"

getSemester(5, 2) -> returns "Spring"

getSemester(5, 26) -> returns "Summer"

getSemester(7, 14) -> returns "Summer"
```

Do not use any other Java classes or methods.

Complete the method on the next page:

```
public static String getSemester(int month, int day) {
```

4. Programming: 16 points - Complete the `countTensPlace` method. The method accepts a `Scanner` object that is already connected to `System.in`. The method counts the number of values entered by the user that have the given digit, also entered by the user, as the second to the last digit in the number. In other words, the method counts the number of values entered by the user that have the given digit in the ten's place.

The method asks the user which digit, 0 through 9, they want to count. You may assume the user always enters a positive integer between 0 and 9 inclusive.

The method then asks the user how many numbers they want to enter. You may assume the user always enters a positive integer in response.

The method prompts the use for the given number of numbers. You may assume the user always enters a valid integer.

Finally, the method prints out the percentage of numbers that have the given digit in the ten's place

Here is an example of the methods output. The user input is shown in bold. Of course the user input could be different than what is shown in the example.

```
Ten's place digit to count: 7

How many numbers? 6

Enter 1: 1201
Enter 2: 355
Enter 3: 1570
Enter 4: -176
Enter 5: 0
Enter 6: 120

33.333333333333336% values with 7 in the ten's place
```

The only method you may use in your answer is the `nextInt()` method from the `Scanner` class.

Do not use any other Java classes or methods.

Specifically, do not use `Strings` or any methods from the `Math` class.

Complete the `countTensPlace` method on the next page.

```
// key is already connected to System.in
public static void countTensPlace(Scanner key) {
```

5. Programming with Strings: 12 points Complete the `replace` method.

The method accepts 3 parameters, a source `String`, a target `char`, and a replacement `String`.

The method returns a new `String` with all occurrences of the target character in the source `String` replaced with the given replacement `String`. Characters in the source `String` that are not equal to the target character appear unchanged in the resulting `String` and in the same relative order they appeared in the source `String`.

The only `String` methods you may use on the question are the `length()`, `charAt(int pos)`, and `String` concatenation, the **+** operator for `String`s.

Examples of calls to the method

```
public static String replace(String source, char tgt, String replace) {

replace("Larry", 'r', "great") -> returns "Lagreatgreaty"

replace("Tim", 't', "J") -> returns "Tim"

replace("Carla", 'a', "") -> returns "Crl"

replace("Varshinee_Rebecca***12_KatE", 'e', "**")
          -> Varshin****_R**b**cca***12_KatE"

replace("", 'S', "312") -> ""

replace("CS312_Fall_2019", '1', "??") -> "CS3??2_Fall_20??9"
```

Recall, the only `String` methods you may use on the question are the `length()`, `charAt(int pos)`, and `String` concatenation, the **+** operator for `String`s. **Do not use any other Java classes or methods.**

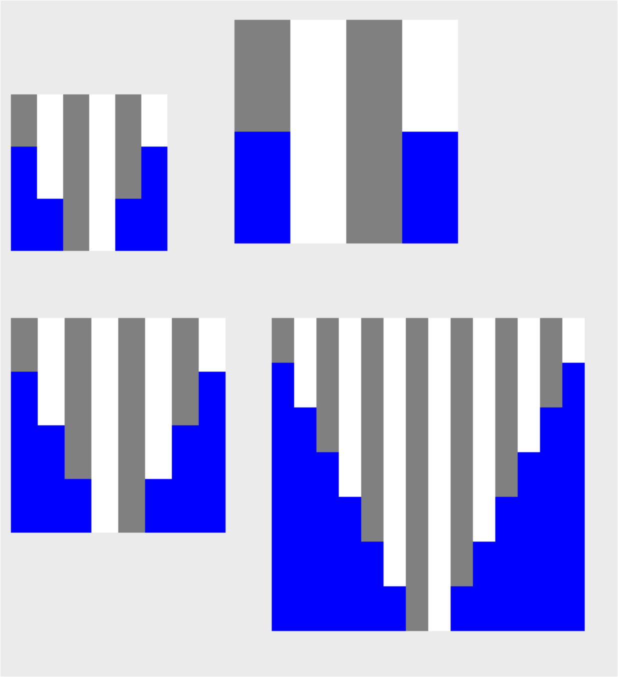
Complete the method on the next page.

```
public static String replace(String source, char tgt, String replace) {
```

6. Graphics Programming: 20 points Complete a method, `drawFigure` that produces the following figure using Java 2D graphics. The parameters for the method are:

- The `Graphics` object for the `DrawingPanel.` You do not know what the current `Color` is.
- The x and y coordinates of the upper left corner of the figure.
- The size of the figure. The figure is square.
- The total number of gray and white rectangles in the figure

The following image shows the results of various calls to `drawFigure`.



The background square is the BLUE constant from the `Color` class. (The background rectangle appears black on the examples to the left.)The rectangles in the figure alternate from left to right between GRAY and WHITE.

You may assume the number of rectangles will be an even number.

You may assume the `size` of a figure is an integer multiple of the number of rectangles.

Recall the following methods from the Graphics class:

```
fillRect(int x, int y, int width,
   int height)

setColor(Color c)
```

Recall the constants from the `Color` class: `Color.BLUE`, `Color.GRAY`, and `Color.WHITE`.

The figures above are produced by the following calls to `drawFigure(Graphics g, int x, int y, int size, int numRectangles)`:

```
drawFigure(g, 20, 200, 210, 6);    // upper left

drawFigure(g, 320, 100, 300, 4);   // upper right

drawFigure(g, 20, 500, 288, 8);    // lower left

drawFigure(g, 370, 500, 420, 14);  // lower right
```

Do not use any Java methods or classes except the `fillRect` and `setColor` methods and the `Color` constants mentioned above.

Complete the method on the next page.

```
public static void drawFigure(Graphics g, int x, int y, int size,
                                   int numRectangles) {
```