CSE142 Sample Midterm
Summer 2018

1. Expressions, 10 points. For each expression in the left-hand column, indicate its value in the right-hand column. Be sure to list a constant of appropriate type (e.g., 7.0 rather than 7 for a double, Strings in quotes).

Expression	Value
`5 * 6 - (4 + 3) * 2 - 2 * 3`	_____
`208 / 20 / 4 + 12 / 10.0 + 0.4 * 2`	_____
`8 - 2 + "8 - 2" + 8 * 2 + 8`	_____
`4 * 5 % 6 + 297 % 10 + 4 % 8`	_____
`13 / 2 * 3.0 + 5.0 * 3 / 2`	_____

2. Parameter Mystery, 12 points. Consider the following program.

```
public class ParameterMystery {
   public static void main(String[] args) {
      String grace = "hopper";
      String george = "boole";
      String alan = george;
      String anita = "borg";
      String ada = "lovelace";

      mystery("turing", "godel", "grace");
      mystery(george, anita, anita);
      mystery(ada, "alan", grace);
      mystery(grace, alan, george);
   }

   public static void mystery(String x, String y, String z) {
      System.out.println(y + " wrote " + z + " with " + x);
   }
}
```

Write the output produced by this program.

3. If/Else Simulation, 12 points. Consider the following method.

```
public static void ifElseMystery(int a, int b) {
    if (a < b) {
        a++;
    }
    if (a < b) {
        a++;
    } else {
        b++;
    }
    if (a >= b) {
        b = b - 5;
    }
    System.out.println(a + " " + b);
}
```

For each call below, indicate what output is produced.

Method Call	Output Produced
ifElseMystery(1, 8);	_____
ifElseMystery(3, 5);	_____
ifElseMystery(4, 5);	_____
ifElseMystery(8, 6);	_____
ifElseMystery(7, 7);	_____
ifElseMystery(5, 7);	_____

4. While Loop Simulation, 12 points. Consider the following method:

```
public static void mystery(int n) {
    int x = 1;
    int y = 2;
    while (y < n) {
        if (n % y == 0) {
            n = n / y;
            x++;
        } else {
            y++;
        }
    }
    System.out.println(x + " " + n);
}
```

For each call below, indicate what output is produced:

Method Call	Output Produced
mystery(4);	_____
mystery(5);	_____
mystery(24);	_____
mystery(28);	_____

5. Assertions, 15 points. You will identify various assertions as being either always true, never true or sometimes true/sometimes false at various points in program execution. The comments in the method below indicate the points of interest.

```
public static int assertions(int n) {
    int x = 2;
    // Point A
    while (x < n) {
        // Point B
        if (n % x == 0) {
            n = n / x;
            x = 2;
            // Point C
        } else {
            x++;
            // Point D
        }
    }
    // Point E
    return n;
}
```

Fill in the table below with the words ALWAYS, NEVER or SOMETIMES.

	x > 2	x < n	n % x == 0
Point A			
Point B			
Point C			
Point D			
Point E			

6. Programming, 15 points. Write a static method named selfCheckout that totals a grocery bill for a customer. The method takes in three parameters: a Scanner connected to System.in, a String representing an item that is on sale, and a double representing the proportional discount on the sale item.

   Your method should first prompt the user for the number of items in their basket. Then, you should prompt the user to enter the name and price of each item in their basket, one at a time. The discount applies to items where the item name exactly matches the name of the sale item. After the user has entered the data for all of their items, print out a summary of the transaction, starting with the final total price, and then the discount (changed into a percentage) and total amount saved by purchasing sale items. You are to match the given output format exactly.

   For example, if the following call to the method is made:

```
Scanner console = new Scanner(System.in);
selfCheckout(console, "Bread", 0.1);
```

   we could expect the following interaction (user input bold and underlined):

```
How many items? 4
Item? Bread
Price? 4.50
Item? Milk
Price? 4.99
Item? Bread
Price? 2.50
Item? OrangeJuice
Price? 6.00
Final total (after discount): $17.29
The 10.0% discount on Bread saved you $0.7!
```

   You may assume that the user will input exactly one token for each item name, and will enter a non-negative number when prompted for the price. Do not round, or add extra 0s to, the double output values. All output should be printed in the given format, regardless of whether the discount is applied or impacts the final total.

7. Programming, 15 points. Write a static method noBigger that continues to generate a sequence of numbers until it generates a number larger than the previous one. Your method should print out the selected number, as well as the probability that the next number selected will continue the non-increasing streak. The method accepts an integer parameter, max, representing the range of numbers to choose from (1 - max, inclusive), and returns the count of numbers generated by the streak, not including the number that ends the streak.

You are to match the format of the output below exactly, though actual output from your program is likely to look different due to randomness:

call: int streak = noBigger(100)

output:

```
Picking numbers from 1 - 100
Number: 87
Probability to continue: 0.87
Number: 42
Probability to continue: 0.42
Number: 42
Probability to continue: 0.42
Number: 3
Probability to continue: 0.03
Number: 2
Probability to continue: 0.02
Number: 7, streak ends
```

value of streak after call: 5

call: int streak = noBigger(5)

output:

```
Picking numbers from 1 - 5
Number: 1
Probability to continue: 0.20
Number: 4, streak ends
```

value of streak after call: 1

You are to use a Random object to generate the numbers; use of Math.random() is not allowed. You may assume that the max passed is greater than 1.

8. Programming, 9 points. Write a static method filter that takes in two int parameters: num and d, and returns the value of num but with all occurrences of the digit d removed. Preserve the order of the remaining digits from num. If filtering removes all digits from the num, return 0.

Example call:	Return:
filter(12, 2);	1
filter(4400, 0);	44
filter(121212, 2);	111
filter(123212, 2);	131
filter(9777775, 7);	95
filter(56665565, 6);	5555
filter(11113, 1);	3
filter(11111, 1);	0
filter(0, 0);	0

You may assume that num and d are both non-negative, and that d is a single digit number. You are not allowed to use Strings to solve this problem.