CSE 142 Sample Midterm Exam #2

1. Expressions

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`, `String`s in quotes).

Expression	Value
`8 + 5 * 3 / 2`	_____
`1.5 * 4 * 7 / 8 + 3.4`	_____
`73 % 10 - 6 % 10 + 28 % 3`	_____
`4 + 1 + 9 + "." + (-3 + 10) + 11 / 3`	_____
`3 / 14 / 7 / (1.0 * 2) + 10 / 6`	_____
`10 > 11 == 4 / 3 > 1`	_____

2. Parameter Mystery

At the bottom of the page, write the output produced by the following program.

```
public class ParameterMystery {
    public static void main(String[] args) {
        String x = "happy";
        String y = "pumpkin";
        String z = "orange";
        String pumpkin = "sleepy";
        String orange = "vampire";

        orange(y, x, z);
        orange(x, z, y);
        orange(pumpkin, z, "y");
        z = "green";
        orange("x", "pumpkin", z);
        orange(y, z, orange);
    }

    public static void orange(String z, String y, String x) {
        System.out.println(y + " and " + z + " were " + x);
    }
}
```

3. If/Else Simulation

For each call of the method below, write the value that is returned:

```
public static int mystery(int n) {
    if (n < 0) {
        n = n * 3;
        return n;
    } else {
        n = n + 3;
    }
    if (n % 2 == 1) {
        n = n + n % 10;
    }
    return n;
}
```

Method Call	Value Returned
mystery(-5)	_____
mystery(0)	_____
mystery(7)	_____
mystery(18)	_____
mystery(49)	_____

4. While Loop Simulation

For each call of the method below, write the value that is returned:

```
public static int mystery(int i, int j) {
    int k = 0;
    while (i > j) {
        i = i - j;
        k = k + (i - 1);
    }
    return k;
}
```

Method Call	Value Returned
`mystery(2, 9)`	_____
`mystery(5, 1)`	_____
`mystery(38, 5)`	_____
`mystery(5, 5)`	_____
`mystery(40, 10)`	_____

5. Assertions

For the following method, identify each of the three assertions in the table below as being either ALWAYS true, NEVER true or SOMETIMES true / sometimes false at each labeled point in the code. You may abbreviate these choices as A/N/S respectively.

```
public static int mystery(Scanner console) {
    int y = 0;
    int z = 1;
    int next = console.nextInt();

    // Point A
    while (next >= 0) {
        // Point B
        if (y > z) {
            // Point C
            z = y;
        }
        y++;
        next = console.nextInt();
        // Point D
    }

    // Point E
    return z;
}
```

	next < 0	y > z	y == 0
Point A			
Point B			
Point C			
Point D			
Point E			

6. Programming

Write a static method named `printMultiples` that accepts an integer n and an integer m as parameters and that prints a complete line of output reporting the first m multiples of n. For example, the following calls:

```
printMultiples(3, 5);
printMultiples(7, 3);
```

should produce this output:

```
The first 5 multiples of 3 are 3, 6, 9, 12, 15
The first 3 multiples of 7 are 7, 14, 21
```

Notice that the multiples are separated by commas. You must exactly reproduce this format. You may assume that the number of multiples you will be asked to generate is greater than or equal to 1.

7. Programming

Write a static method named `monthApart` that accepts four integer parameters representing two calendar dates. Each date consists of a month (1 through 12) and a day (1 through the number of days in that month [28-31]). Assume that all dates occur during the same year. The method returns whether the dates are at least a month apart. For example, the following dates are all considered to be at least a month apart from 9/19 (September 19): 2/14, 7/25, 8/2, 8/19, 10/19, 10/20, and 11/5. The following dates are NOT at least a month apart from 9/19: 9/20, 9/28, 10/1, 10/15, and 10/18. Note that the first date could come before or after (or be the same as) the second date. Assume that all parameter values passed are valid.

Sample calls:

```
monthApart( 6, 14,  9, 21) should return true,   because June 14 is at least a month before September 21
monthApart( 4,  5,  5, 15) should return true,   because April 5 is at least a month before May 15
monthApart( 4, 15,  5, 15) should return true,   because April 15 is at least a month before May 15
monthApart( 4, 16,  5, 15) should return false,  because April 16 isn't at least a month apart from May 15
monthApart( 6, 14,  6,  8) should return false,  because June 14 isn't at least a month apart from June 8
monthApart( 7,  7,  6,  8) should return false,  because July 7 isn't at least a month apart from June 8
monthApart( 7,  8,  6,  8) should return true,   because July 8 is at least a month after June 8
monthApart(10, 14,  7, 15) should return true,   because October 14 is at least a month after July 15
```

8. Programming

Write a static method named `threeHeads` that repeatedly flips a coin until three heads *in a row* are seen. You should use a `Random` object to give an equal chance to a head or a tail appearing. Each time the coin is flipped, what is seen is displayed (H for heads, T for tails). When 3 heads in a row are flipped a congratulatory message is printed. Here are possible outputs of two calls to `threeHeads`:

```
T T T H T H H H
Three heads in a row!
```

```
T H T H T T T T T H H T H H H
Three heads in a row!
```