Key to CSE142 Midterm, Winter 2018

1.

Expression	Value
12 + 3 / 5 + 3 % 2	13
7 + 1 + "4 + 2" + 1 + 7	"84 + 217"
15 / 4 / 3.0 - 18 / 5 + (15 / 10.0)	-0.5
!(7 * 2 != 42 && !(5 / 2 == 2))	true
6 % 4 + 4 % 6 + 6 % 6	6

2. The program produces the following output:

```
to be or ophelia to or
not be or or to be
or be or to to or
to be or not to be?
```

3.

Method Call	Output Produced
ifElseMystery(12, 45);	12 44
ifElseMystery(5, 15);	8 5
ifElseMystery(64, 8);	13 8
ifElseMystery(12, 12);	1 11
ifElseMystery(10, 3);	4 3
ifElseMystery(120, 6);	122 6

4.

Method Call	Output Produced
mystery(1);	1 1
mystery(4);	2 3
mystery(6);	2 8

5.

	x % 2 == 1	y > x	y % 2 == 1
Point A	sometimes	sometimes	never
Point B	sometimes	always	sometimes
Point C	never	always	never
Point D	never	always	never
Point E	sometimes	never	sometimes

6. One possible solution appears below.

```
public static int dogHears(String name, int numWords, Scanner console) {
    int count = 0;

    for (int i = 0; i < numWords; i++) {
        System.out.print("word? ");
        String input = console.next();

        System.out.print("dog hears: \"");
        if (input.equals(name)) {
            System.out.print(name);
            count++;
        } else {
            System.out.print("blah");
        }
        System.out.println("\"");
    }

    return count;
}
```

7. One possible solution appears below.

```
public static void walkHome(int start, Random rand) {
   int distance = start;
   int total = 0;

   System.out.println("starting at " + start);

   while (distance > 0) {
      System.out.print("*");
      for (int i = 0; i < distance; i++) {
         System.out.print("-");
      }
      System.out.println("|^|");

      int steps = rand.nextInt(5) - 2;
      if (steps > distance) {
         steps = distance;
      }
      distance -= steps;
      total += Math.abs(steps);
      System.out.println("moving " + steps + " step(s)");
   }

   System.out.println("*|^|");
   System.out.println("made it home in " + total + " step(s)");
}
```

8. Three possible solutions appear below.

```
    // get one digit, remove it, compare it to the new value of n % 10
    public static int digitsInARow(int n) {
        int max = 1;
        int count = 1;
        while (n > 0) {
            int next = n % 10;
            n = n / 10;
            if (n % 10 == next) {
                count++;
            } else {
                count = 1;
            }
            if (count > max) {
                max = count;
            }
        }
        return max;
    }

    // keep track of a prev digit and compare it to the next digit
    public static int digitsInARow(int n) {
        int max = 1;
        int count = 0;
        int prev = -1;
        while (n != 0) {
            int next = n % 10;
            n = n / 10;
            if (next == prev) {
                count++;
            } else {
                count = 1;
            }
            max = Math.max(max, count);
            prev = next;
        }
        return max;
    }

    // count how many times the last 2 digits are divisible by 11
    public static int digitsInARow(int n) {
        int max = 1;
        int count = 1;
        while (n > 0) {
            if (n % 100 % 11 == 0) {
                count++;
            } else {
                count = 1;
            }
            max = Math.max(max, count);
            n = n / 10;
        }
        return max;
    }
```