Key to CSE143X Midterm, Fall 2020

1.

| Expression | Value |
|---|---|
| 3 * (5 - 2) - 3 - 2 * 2 | 2 |
| 4 * 7 % 8 + 132 % 10 + 3 % 4 | 9 |
| 27 / 5 / 2 + 3.4 * 2 - 1.1 * 2 | 6.6 |
| 9 + 9 + "9 + 9" + 9 + 9 | "189 + 999" |
| 19 / 2 / 2.0 + 2.5 * 6 / 2 + 0.5 * 4 | 14.0 |

2. Parameter Mystery. The program produces the following output.

```
semi missing brace and a 42
semi missing 42 and a 8
brace missing literal and a semi
84 missing 1 and a cse
```

3.

| Method Call | Output Produced |
|---|---|
| ifElseMystery(2, 2); | 4 2 |
| ifElseMystery(3, 1); | 1 4 |
| ifElseMystery(4, 0); | 4 1 |
| ifElseMystery(5, 3); | 4 7 |
| ifElseMystery(1, 2); | 2 0 |
| ifElseMystery(7, 4); | 8 7 |

4.

| Method Call | Output Produced |
|---|---|
| mystery(1); | 1 1 |
| mystery(5); | 3 4 |
| mystery(10); | 4 8 |
| mystery(42); | 6 32 |

5.

| | x > 0 | x == 0 | y == 0 |
|---|---|---|---|
| Point A | always | never | always |
| Point B | always | never | sometimes |
| Point C | never | never | never |
| Point D | never | sometimes | never |
| Point E | never | always | never |

6. One possible solution appears below.

```
public static void spinWheel(Random r, int n) {
    int spin = r.nextInt(5) * 10 + 20;
    System.out.print("spins: " + spin);
    int count = 0;
    if (spin == 20) {
        count++;
    }
    int totalSpins = 1;
    while (count < n) {
        spin = r.nextInt(5) * 10 + 20;
        totalSpins++;
        System.out.print(", " + spin);
        if (spin == 20) {
            count++;
        } else {
            count = 0;
        }
    }
    System.out.println();
    System.out.println(n + " in a row after " + totalSpins + " spins");
}
```

7. One possible solution appears below.

```
public static void printFigure(Scanner input) {
    while (input.hasNextLine()) {
        String line = input.nextLine();
        Scanner data = new Scanner(line);
        while (data.hasNextInt()) {
            int count = data.nextInt();
            String text = data.next();
            for (int i = 0; i < count; i++) {
                if (text.equals("space")) {
                    System.out.print(" ");
                } else {
                    System.out.print(text);
                }
            }
        }
        System.out.println();
    }
}
```

8. Arrays. One possible solution appears below.

```
public static int numUnique(int[] list) {
    if (list.length == 0) {
        return 0;
    } else {
        int count = 1;
        for (int i = 1; i < list.length; i++) {
            if (list[i] != list[i - 1]) {
                count++;
            }
        }
        return count;
    }
}
```

9. Two possible solutions appear below.

```
public static int numWords(String s) {
    int count = 0;
    boolean inWord = false;
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) == ' ') {
            inWord = false;
        } else if (!inWord) {
            count++;
            inWord = true;
        }
    }
    return count;
}
```

```
public static int numWords(String s) {
    int count = 0;
    for (int i = 1; i < s.length(); i++) {
        if (s.charAt(i - 1) == ' ' && s.charAt(i) != ' ') {
            count++;
        }
    }
    if (s.length() > 0 && s.charAt(0) != ' ') {
        count++;
    }
    return count;
}
```