

CSE142 Sample Midterm Key
Spring 2019

1. Expression	Value
<code>1 + 2 * 3 - 4 * (5 + 6) / 7 + 8 % 9</code>	9
<code>20 / 7 * 2.0 + 5.0 / 2 - (1 / 4)</code>	6.5
<code>3 * 4 + "3" + 9 * 5 + 6</code>	"123456"
<code>(3 * 5 < 1 + 2) (6 != 5 && !(7 < 7))</code>	true
<code>1 + 1 * (1 - 1) + (1 + 1 + 1) % (1 + 1)</code>	2

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

```
a snow and slippery for sleet
a storm and snow for sleetsnow
a snowsleet and sleetsnow for snowstorm
a sun and storm for sunny
```

3. Method Call	Output Produced
<code>ifElseMystery(6, 5);</code>	5 15
<code>ifElseMystery(2, 2);</code>	4 2
<code>ifElseMystery(3, 1);</code>	1 4
<code>ifElseMystery(4, 0);</code>	4 1
<code>ifElseMystery(5, 3);</code>	4 7
<code>ifElseMystery(1, 2);</code>	2 0
<code>ifElseMystery(7, 4);</code>	8 7

4. Method Call	Output Produced
<code>mystery(8);</code>	1 8
<code>mystery(32);</code>	2 5
<code>mystery(184);</code>	3 13
<code>mystery(8239);</code>	4 22

5.	x == 1	x % 2 == 1	y == 0
Point A	sometimes	sometimes	always
Point B	never	sometimes	sometimes
Point C	sometimes	sometimes	never
Point D	never	never	sometimes
Point E	always	always	sometimes

6. One possible solution appears below.

```
public static int generate(Scanner console) {
    System.out.print("number between -5 and 5? ");
    int target = console.nextInt();
    Random r = new Random();
    int next = r.nextInt(11) - 5;
    System.out.print("numbers are: " + next);
    int count = 1;
    while (next != target) {
        next = r.nextInt(11) - 5;
        System.out.print(", " + next);
        count++;
    }
    System.out.println();
    System.out.println("came up after " + count + " tries");
    return count;
}
```

7. One possible solution appears below.

```
public static int digitRange(int n) {
    int min = n % 10;
    int max = n % 10;
    while (n > 0) {
        int digit = n % 10;
        n = n / 10;
        if (digit < min) {
            min = digit;
        }
        if (digit > max) {
            max = digit;
        }
    }
    return max - min;
}
```

8. Two possible solutions appear below.

```
public static void printStripped(String s) {
    boolean inComment = false;
    for (int i = 0; i < s.length(); i++) {
        char next = s.charAt(i);
        if (next == '<') {
            inComment = true;
        } else if (inComment && next == '>') {
            inComment = false;
        } else if (!inComment) {
            System.out.print(next);
        }
    }
    System.out.println();
}
```

```
public static void printStripped(String s) {
    int start = s.indexOf('<');
    while (start != -1) {
        int stop = s.indexOf('>', start + 1);
        s = s.substring(0, start) + s.substring(stop + 1);
        start = s.indexOf('<');
    }
    System.out.println(s);
}
```