

## Sample Midterm Exam #2 Key

Also check out *Practice-It* to test solving these problems or to type in your own solution to see if it works!

### 1. Expressions

<u>Expression</u>	<u>Value</u>
$8 + 5 * 3 / 2$	15
$1.5 * 4 * 7 / 8 + 3.4$	8.65
$73 \% 10 - 6 \% 10 + 28 \% 3$	-2
$4 + 1 + 9 + "." + (-3 + 10) + 11 / 3$	"14.73"
$3 / 14 / 7 / (1.0 * 2) + 10 / 6$	1.0
$10 > 11 == 4 / 3 > 1$	true

### 2. Parameter Mystery

happy and pumpkin were orange  
orange and happy were pumpkin  
orange and sleepy were y  
pumpkin and x were green  
green and pumpkin were vampire

### 3. If/Else Simulation

<u>Method Call</u>	<u>Value Returned</u>
mystery(-5)	-15
mystery(0)	6
mystery(7)	10
mystery(18)	22
mystery(49)	52

### 4. While Loop Simulation

<u>Method Call</u>	<u>Value Returned</u>
mystery(2, 9)	0
mystery(5, 1)	6
mystery(38, 5)	119
mystery(5, 5)	0
mystery(40, 10)	57

### 5. Assertions

	$next < 0$	$y > z$	$y == 0$
Point A	SOMETIMES	NEVER	ALWAYS
Point B	NEVER	SOMETIMES	SOMETIMES
Point C	NEVER	ALWAYS	NEVER
Point D	SOMETIMES	SOMETIMES	NEVER
Point E	ALWAYS	SOMETIMES	SOMETIMES

## 6. Programming (one solution shown)

```
public static void printMultiples(int n, int times) {
    System.out.print("The first " + times + " multiples of " + n + " are " + n);
    for (int i = 2; i <= times; i++) {
        System.out.print(", " + i * n);
    }
    System.out.println();
}
```

## 7. Programming (four solutions shown)

```
public static boolean monthApart(int m1, int d1, int m2, int d2) {
    if (m1 == m2) {
        return false;
    } else if (m1 <= m2 - 2) {
        return true;
    } else if (m1 >= m2 + 2) {
        return true;
    } else if (m1 == m2 - 1) {
        if (d1 <= d2) {
            return true;
        } else {
            return false;
        }
    } else if (m1 == m2 + 1) {
        if (d1 >= d2) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

```
public static boolean monthApart(int m1, int d1, int m2, int d2) {
    if (m1 < m2 - 1 || m1 > m2 + 1) {
        return true;
    } else if (m1 == m2 - 1 && d1 <= d2) {
        return true;
    } else if (m1 == m2 + 1 && d1 >= d2) {
        return true;
    } else {
        return false;
    }
}
```

```
public static boolean monthApart(int m1, int d1, int m2, int d2) {
    return (m2 - m1 > 1) || (m1 - m2 > 1) ||
        (m2 - m1 == 1 && d1 <= d2) ||
        (m1 - m2 == 1 && d1 >= d2);
}
```

```
public static boolean monthApart(int m1, int d1, int m2, int d2) {
    return Math.abs((m1 * 31 + d1) - (m2 * 31 + d2)) >= 31;
}
```

## 8. Programming (one solution shown)

```
public static void threeHeads() {
    Random r = new Random();
    int numHeads = 0;

    while (numHeads < 3) {
        int flip = r.nextInt(2); // flip coin
        if (flip == 0) { // heads
            numHeads++;
            System.out.print("H ");
        } else {
            numHeads = 0;
            System.out.print("T ");
        }
    }
    System.out.println();
    System.out.println("Three heads in a row!");
}
```