

CSE 142, Autumn 2011 Midterm Exam Key

1. Expressions

<u>Expression</u>	<u>Value</u>
<code>5 + 2 * 4 / 3 + 5</code>	12
<code>5 + 5 + "23.0" + 5 + 2 * 5</code>	"1023.0510"
<code>!(5 > 2 && -2 > 2) 5 / 2 == 0</code>	true
<code>15 % 9 % 4 + 4 % 6 % 3</code>	3
<code>12 / 5 / 2.0 + 2 * 4</code>	9.0

2. Parameter Mystery

Twas **snack** and the **jubjub** toves did **vorpal**
 Twas **gyre** and the **mutl** toves did **jubjub**
 Twas **vorpal** and the **mut** toves did **jubjubsnack**
 Twas **jubjub** and the **snack** toves did **tumtum**

3. If/Else Mystery

<u>Method Call</u>	<u>Values Printed</u>
<code>ifElseMystery(4, 1);</code>	2 1 2
<code>ifElseMystery(-10, 100);</code>	-10 100 0
<code>ifElseMystery(18, 4);</code>	9 -1 1
<code>ifElseMystery(-12, 5);</code>	-12 5 0

4. While Loop Mystery

<u>Method Call</u>	<u>Values Printed</u>
<code>mystery(2, 3);</code>	3, 2, -1
<code>mystery(3, 5);</code>	5, 4, -1, -5
<code>mystery(4, 7);</code>	7, 6, -1, -7, -6

5. Assertions

	h == 0	f >= 5	num == 0
Point A	ALWAYS	SOMETIMES	SOMETIMES
Point B	SOMETIMES	NEVER	SOMETIMES
Point C	ALWAYS	SOMETIMES	ALWAYS
Point D	SOMETIMES	NEVER	NEVER
Point E	ALWAYS	ALWAYS	SOMETIMES

6. Programming

There are many ways to solve any programming problem. Here are some common correct solutions we saw:

```
public static boolean twoConsecutive(int a, int b, int c) {
    return Math.abs(a - b) == 1 || Math.abs(a - c) == 1 ||
        Math.abs(b - c) == 1;
}

public static boolean twoConsecutive(int a, int b, int c) {
    if (Math.abs(a - b) == 1) {
        return true;
    } else if (Math.abs(a - c) == 1) {
        return true;
    } else if (Math.abs(b - c) == 1) {
        return true;
    }
    return false;
}

public static boolean twoConsecutive(int a, int b, int c) {
    if (a - 1 == b || a - 1 == c || b - 1 == a || b - 1 == c || c - 1 == a || c - 1 ==
b) {
        return true;
    } else {
        return false;
    }
}

public static boolean twoConsecutive(int a, int b, int c) {
    if (b == a + 1 || a == b + 1) {
        return true;
    } else if (c == b + 1 || b == c + 1) {
        return true;
    } else if (a == c + 1 || c == a + 1) {
        return true;
    } else {
        return false;
    }
}

public static boolean twoConsecutive(int a, int b, int c) {
    boolean test = false;
    if (a - b == 1 || b - a == 1) {
        test = true;
    }
    if (a - c == 1 || c - a == 1) {
        test = true;
    }
    if (b - c == 1 || c - b == 1) {
        test = true;
    }
    return test;
}

public static boolean twoConsecutive(int a, int b, int c) {
    if (b == a + 1) {
        return true;
    }
    if (a == b + 1) {
        return true;
    }
    if (c == b + 1) {
        return true;
    }
    if (b == c + 1) {
        return true;
    }
    if (a == c + 1) {
        return true;
    }
    if (c == a + 1) {
        return true;
    }
    return false;
}
```

7. Programming

```
public static void stitching(int w, int h) {
    for (int line = 1; line <= h; line++) {
        for (int i = 0; i < w; i++) {
            if (line % 2 == 0) {
                System.out.print((i + line) + "-"); // parens matter
            } else {
                System.out.print("-" + (i + line));
            }
        }
        System.out.println();
    }
}

public static void stitching(int w, int h) {
    for (int line = 1; line <= h; line++) {
        for (int i = line; i < line + w; i++) {
            if (line % 2 == 0) {
                System.out.print(i + "-");
            } else {
                System.out.print("-" + i);
            }
        }
        System.out.println();
    }
}

public static void stitching(int w, int h) {
    for (int line = 1; line <= h; line++) {
        if (line % 2 == 0) {
            for (int i = 0; i < w; i++) {
                System.out.print(i + line + "-");
            }
        } else {
            for (int i = 0; i < w; i++) {
                System.out.print("-" + (i + line)); // parens matter
            }
        }
        System.out.println();
    }
}

public static void stitching(int w, int h) {
    for (int line = 1; line <= h; line += 2) { // +=2 is important here (or go up to
        h/2, etc.)
        for (int i = line; i < line + w; i++) {
            System.out.print("-" + i);
        }
        System.out.println();
        if (line + 1 <= h) {
            for (int i = line + 1; i < line + 1 + w; i++) {
                System.out.print(i + "-");
            }
            System.out.println();
        }
    }
}

public static void stitching(int w, int h) {
    int counter = 1;
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            if (counter % 2 == 0) {
                System.out.print(counter + j + "-");
            } else {
                System.out.print("-" + (counter + j)); // parens matter
            }
        }
        counter++;
        System.out.println();
    }
}
```

8. Programming

```
public static void sameFlip(Random r) {
    int prev = -1;
    int curr = -2;

    while (curr != prev) {
        prev = curr;
        curr = r.nextInt(2);
        if (curr == 0) {
            System.out.print("H");
        } else {
            System.out.print("T");
        }
    }
}

public static void sameFlip(Random r) {
    int heads = 0;
    int tails = 0;
    while (heads < 2 && tails < 2) {
        int flip = r.nextInt(2);
        if (flip == 0) {
            System.out.print("H");
            heads++;
            tails = 0;
        } else {
            System.out.print("T");
            tails++;
            heads = 0;
        }
    }
}
```