

## CSE 142, Summer 2010 Midterm Exam Key

### 1. Expressions

For each expression at left, indicate its value in the right column. List a value of appropriate type and capitalization. e.g., 7 for an int, 7.0 for a double, "hello" for a String, true or false for a boolean.

<u>Expression</u>	<u>Value</u>
<code>11 - 7 * 2 + 3 * 4</code>	9
<code>10 / (5 / 2) + 1.5 * 6 / 2</code>	9.5
<code>115 / 10 + 115 % 10 + 11 / 7 * 3.5</code>	19.5
<code>"0" + 10 * 5 + "0" + 5 + 6</code>	"050056"
<code>(4 &lt;= 9 / 2) &amp;&amp; !(2 != 10)</code>	false

### 2. Parameter Mystery

peacock in the study with the dagger  
study in the peacock with the mustard  
lounge in the mustard with the pipe  
pipe in the dagger with the ballroom  
miss mustard in the ballroom with the study

### 3. If/Else Simulation

<u>Method Call</u>	<u>Output</u>
<code>ifElseMystery(20, 8);</code>	30 80
<code>ifElseMystery(30, 30);</code>	40 300
<code>ifElseMystery(4, 3);</code>	4 4
<code>ifElseMystery(30, 4);</code>	40 41
<code>ifElseMystery(7, 7);</code>	7 8

## 4. While Loop Simulation

<u>Method Call</u>	<u>Output</u>
<code>whileMystery(3, 5);</code>	3, 4, 2
<code>whileMystery(5, 3);</code>	0
<code>whileMystery(-3, 6);</code>	-3, -2, 0, 3
<code>whileMystery(2, 12);</code>	2, 3, 5, 8, 4
<code>whileMystery(-9, 10);</code>	-9, -8, -6, -3, 1, 5

## 5. Assertions

	<code>num &lt; 0</code>	<code>first &lt; second</code>	<code>num &gt;= second</code>
<b>Point A</b>	SOMETIMES	NEVER	ALWAYS
<b>Point B</b>	NEVER	SOMETIMES	SOMETIMES
<b>Point C</b>	NEVER	ALWAYS	NEVER
<b>Point D</b>	NEVER	SOMETIMES	SOMETIMES
<b>Point E</b>	ALWAYS	SOMETIMES	SOMETIMES

## 6. Programming

```
public static boolean speedingTicket(int speed, int mph, boolean donut) {
    if (donut && (speed - mph >= 10)) {
        return true;
    } else if (!donut && ((speed - mph >= 5) || (mph - speed >= 5))) {
        return true;
    } else if (speed >= 100) {
        return true;
    } else {
        return false;
    }
}
```

```
public static boolean speedingTicket(int speed, int mph, boolean donut) {
    if ((donut && (speed - mph >= 10))
        || (!donut && (Math.abs(speed - mph) >= 5))
        || (speed >= 100)) {
        return true;
    } else {
        return false;
    }
}
```

```
public static boolean speedingTicket(int speed, int mph, boolean donut) {
    return ((donut && (speed - mph >= 10))
        || (!donut && (Math.abs(speed - mph) >= 5))
        || (speed >= 100));
}
```

## 7. Programming

```
public static void dogHears(String dogName, int lines) {
    Random rand = new Random();

    for (int i = 1; i <= lines; i++) {
        int wpl = rand.nextInt(9) + 2;
        for (int j = 1; j <= wpl; j++) {
            if (rand.nextInt(4) == 0) {
                System.out.print(dogName + " ");
            }
            else {
                System.out.print("blah ");
            }
        }
        System.out.println();
    }
}
```

## 8. Programming

```
// boolean flag solution
public static boolean highLow(int number) {
    int digit = number % 10;
    number = number / 10;
    boolean lookForHigh = (digit < 5);

    while (number != 0) {
        digit = number % 10;
        number = number / 10;
        if (lookForHigh && (digit >= 5)) {
            lookForHigh = false;
        } else if (!lookForHigh && (digit < 5)) {
            lookForHigh = true;
        } else {
            return false;
        }
    }

    return true;
}

// "prev and curr" solution
public static boolean highLow(int n) {
    int prev = 3; // 3 to start; then 1 for 'low' and 2 for 'high'
    while (n != 0) {
        int curr;
        if (n % 10 <= 4)
            curr = 1;
        else
            curr = 2;
        if (prev == curr) {
            return false;
        }
        n /= 10;
        prev = curr;
    }
    return true;
}

// "peek ahead" solution
public static boolean highLow(int n) {
    while (n != 0) {
        int digit = n % 10;
        n = n / 10;
        if (digit <= 4 && n > 0 && n % 10 <= 4) {
            return false;
        }
        if (digit >= 5 && n > 0 && n % 10 >= 5) {
            return false;
        }
    }
    return true;
}
```