

# CSE 142, Autumn 2010 Midterm Exam Key

## 1. Expressions

<u>Expression</u>	<u>Value</u>
12 / 3 + 5 + 3 * -2	3
1 + 1 + "(1 + 1)" + 1 + 1	"2(1 + 1)11"
13 / 2 - 38 / 5 / 2.0 + (15 / 10.0)	4.0
11 < 3 + 4    !(5 / 2 == 2)	false
20 % 6 + 6 % 20 + 6 % 6	8

## 2. Parameter Mystery

```
3 + 5 = 1
10 + 3 = 5
7 + 4 = 7
2 + 5 = 2
```

## 3. If/Else Simulation

<u>Method Call</u>	<u>Output</u>
ifElseMystery(5, 5);	16 5
ifElseMystery(18, 4);	2 6
ifElseMystery(3, 6);	5 8

## 4. While Loop Simulation

<u>Method Call</u>	<u>Output</u>
whileMystery(7, 5);	2 -2 3
whileMystery(20, 4);	16 13 11 10 0
whileMystery(40, 10);	30 21 13 6 0 5

## 5. Assertions

	c > 3	d <= m	c == 0
<b>Point A</b>	NEVER	ALWAYS	ALWAYS
<b>Point B</b>	NEVER	SOMETIMES	SOMETIMES
<b>Point C</b>	SOMETIMES	ALWAYS	NEVER
<b>Point D</b>	NEVER	NEVER	ALWAYS
<b>Point E</b>	ALWAYS	ALWAYS	NEVER

## 6. Programming

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

```
public static void xo(int size) {
    for (int i = 1; i <= size; i++) {
        for (int j = 1; j <= size; j++) {
            if (j == i || j == size - (i - 1)) { // note the (i - 1), not i
                System.out.print("x");
            } else {
                System.out.print("o");
            }
        }
        System.out.println();
    }
}
```

```
public static void xo(int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (j == i || j == size - 1 - i) {
                System.out.print("x");
            } else {
                System.out.print("o");
            }
        }
        System.out.println();
    }
}
```

```
public static void xo(int size) {
    int x1 = 1;
    int x2 = size;
    for (int i = 1; i <= size; i++) {
        for (int j = 1; j <= size; j++) {
            if (x1 == j || x2 == j) {
                System.out.print("x");
            } else {
                System.out.print("o");
            }
        }
        System.out.println();
        x1++;
        x2--;
    }
}
```

```

public static void xo(int size) {
    for (int i = 0; i < size / 2; i++) {    // top half
        repeat("o", i);
        repeat("x", 1);
        repeat("o", size - 2*i - 2);
        repeat("x", 1);
        repeat("o", i);
        System.out.println();
    }

    if (size % 2 != 0) {    // middle line (when odd size)
        repeat("o", size / 2);
        repeat("x", 1);
        repeat("o", size / 2);
        System.out.println();
    }

    for (int i = size / 2 - 1; i >= 0; i--) {    // bottom half
        repeat("o", i);
        repeat("x", 1);
        repeat("o", size - 2*i - 2);
        repeat("x", 1);
        repeat("o", i);
        System.out.println();
    }
}

public static void repeat(String s, int n) {
    for (int i = 0; i < n; i++) {
        System.out.print(s);
    }
}

```

## 7. Programming

```
// nested ifs with || solution
public static boolean anglePairs(int a1, int a2, int a3) {
    if (a1 + a2 == 90 || a2 + a3 == 90 || a3 + a1 == 90) {
        if (a1 + a2 == 180 || a2 + a3 == 180 || a3 + a1 == 180) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

// boolean zen solution
public static boolean anglePairs(int a1, int a2, int a3) {
    return (a1 + a2 == 90 || a2 + a3 == 90 || a3 + a1 == 90) &&
        (a1 + a2 == 180 || a2 + a3 == 180 || a3 + a1 == 180);
}

// eliminate false cases first solution
public static boolean anglePairs(int a1, int a2, int a3) {
    if (a1 + a2 != 90 && a2 + a3 != 90 && a3 + a1 != 90) {
        return false;
    }
    if (a1 + a2 != 180 && a2 + a3 != 180 && a3 + a1 != 180) {
        return false;
    }
    return true;
}

// nest by which pair matches 90 vs. 180
public static boolean anglePairs(int a1, int a2, int a3) {
    if (a1 + a2 == 90) {
        if (a2 + a3 == 180 || a1 + a3 == 180) {
            return true;
        } else {
            return false;
        }
    } else if (a2 + a3 == 90) {
        if (a1 + a2 == 180 || a1 + a3 == 180) {
            return true;
        } else {
            return false;
        }
    } else if (a1 + a3 == 90) {
        if (a1 + a2 == 180 || a2 + a3 == 180) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

```

// nest by 90 check, zen version
public static boolean anglePairs(int a1, int a2, int a3) {
    if (a1 + a2 == 90) {
        return a2 + a3 == 180 || a1 + a3 == 180;
    } else if (a2 + a3 == 90) {
        return a1 + a2 == 180 || a1 + a3 == 180;
    } else if (a1 + a3 == 90) {
        return a1 + a2 == 180 || a2 + a3 == 180;
    }
    return false;
}

// nest by 90 check, very zen version
public static boolean anglePairs(int a1, int a2, int a3) {
    return (a1 + a2 == 90 && (a2 + a3 == 180 || a1 + a3 == 180))
        || (a1 + a3 == 90 && (a1 + a2 == 180 || a2 + a3 == 180))
        || (a1 + a3 == 90 && (a1 + a2 == 180 || a2 + a3 == 180));
}

// boolean flags version
public static boolean anglePairs(int a, int b, int c) {
    boolean supp = false;
    boolean comp = false;
    if (a + b == 90 || a + c == 90 || b + c == 90) {
        comp = true;
    }
    if (a + b == 180 || a + c == 180 || b + c == 180) {
        supp = true;
    }
    return supp && comp;
}

// zenned out && pairs
public static boolean anglePairs(int a1, int b, int c) {
    return a+b==90 && a+c==180 || a+b==90 && b+c==180
        || a+c==90 && a+b==180 || a+c==90 && b+c==180
        || b+c==90 && a+b==180 || b+c==90 && a+c==180;
}

```

## 8. Programming

```
public static int baseball(Scanner console) {
    int score1 = 0;
    int score2 = 0;
    int inning = 1;

    while ((inning <= 9 || score1 == score2) && Math.abs(score1 - score2) < 10) {
        System.out.print("Inning #" + inning + ": ");
        score1 += console.nextInt();
        score2 += console.nextInt();
        inning++;
    }

    System.out.println("Final score: " + score1 + " - " + score2);
    if (score1 > score2) {
        return 1;
    } else {
        return 2;
    }
}
```

```
public static int baseball (Scanner console) {
    int inning = 1;
    int team1 = 0;
    int team2 = 0;
    while (inning <= 9) {
        System.out.print("Inning #" + inning + ": ");
        team1 += console.nextInt();
        team2 += console.nextInt();
        if((team1-team2 >=10) || (team2 - team1 >= 10)) {
            System.out.println ("Final score: " + score1 + " - " + score2);
            if (team1 > team2) {
                return 1;
            } else {
                return 2;
            }
        } else {
            inning++;
        }
    }
    while (team2 == team1) {
        System.out.print ("Inning #" + inning + ": ");
        team1 += console.nextInt();
        team2 += console.nextInt();
    }
    System.out.println("Final score: " + score1 + " - " + score2);
    if (team1 > team2) {
        return 1;
    } else {
        return 2;
    }
}
```