CSE 142, Winter 2008
Midterm Exam Key

1. Expressions

Expression	Value
`4 + 3 * 2 - 1`	`9`
`1 + ( 2 + " ( 3 + " + 4 + " 5 ) " ) + 6`	`"12 ( 3 + 4 5 ) 6"`
`100 % 30 % 4 + 5 % 2`	`3`
`!(1 + 1 >= 2) || 10 / 3 == 3`	`true`
`2.0 * 2.2 + -23 / 2 / 2.0`	`-1.1`
`10 > 5 && (2 > 3 || 5 <= 4 + 2)`	`true`

2. Parameter Mystery

A **knight** and a **rook** beats a **pawn**
A **pawn** and a **knight** beats a **rook**
A **queen** and a **pawn** beats a **queen**
A **rook** and a **bishop** beats a **knight**
A **king** and a **queen** beats a **king**

3. While Loop Simulation

Method Call	Value Returned
`whileMystery(2)`	2
`whileMystery(84)`	4
`whileMystery(4097)`	8
`whileMystery(388)`	6
`whileMystery(23980)`	10

4. Assertions

	count < 10	**even > 0**	**even < 3**
Point A	ALWAYS	NEVER	ALWAYS
Point B	ALWAYS	SOMETIMES	ALWAYS
Point C	SOMETIMES	SOMETIMES	ALWAYS
Point D	SOMETIMES	ALWAYS	SOMETIMES
Point E	SOMETIMES	SOMETIMES	SOMETIMES

5. Programming

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

```
public static boolean canMakeChange(int pennies, int nickels, int cents) {
    if (cents % 5 > pennies) {
        return false;
    } else if (pennies + 5 * nickels < cents) {
        return false;
    } else {
        return true;
    }
}

public static boolean canMakeChange(int pennies, int nickels, int cents) {
    while (cents >= 5 && nickels > 0) {
        cents -= 5;
        nickels--;
    }
    while (cents > 0 && pennies > 0) {
        cents--;
        pennies--;
    }
    return cents == 0;
}

public static boolean canMakeChange(int pennies, int nickels, int cents) {
    for (int p = 0; p <= pennies; p++) {
        for (int n = 0; n <= nickels; n++) {
            if (p + 5 * n == cents) {
                return true;
            }
        }
    }
    return false;
}

public static boolean canMakeChange(int pennies, int nickels, int cents) {
    if (cents > nickels * 5 + pennies) {
        return false;
    }
    if (nickels * 5 >= cents) {
        if (pennies >= cents % 5) {  // enough nickels to cover all except % 5 part
            return true;
        } else {
            return false;
        }
    } else {
        if (pennies >= cents - nickels * 5) {
            return true;  // not enough nickels; need pennies
        } else {
            return false;
        }
    }
}

public static boolean canMakeChange(int pennies, int nickels, int cents) {
    if (nickels * 5 >= cents) {
        return (pennies >= cents % 5);  // enough nickels to cover all except % 5 part
    } else {
        return (pennies >= cents - nickels * 5);   // not enough nickels; need pennies
    }
}

public static boolean canMakeChange(int pennies, int nickels, int cents) {
    cents -= Math.min(nickels * 5, cents - cents % 5);
    return cents - pennies <= 0;
}

public static boolean canMakeChange(int pennies, int nickels, int cents) {
    return (pennies >= cents % 5) && (pennies + 5 * nickels >= cents);
}
```

6. Programming

```
public static void meetings(int h, int m) {
    System.out.println(h + "h " + m + "m start time");
    Random rand = new Random();
    int count = 0;               // count makes it so we always enter loop
    while ((m > 5 && m < 55) || count == 0) {
        count++;
        int r = rand.nextInt(16) + 15;
        if (m + r < 60) {
            m += r;              // normal case, just add minutes
        } else {
            m = (m + r) - 60;    // minutes-wrapping case
            if (h == 12) {
                h = 1;           // hours-wrapping case
            } else {
                h++;
            }
        }
        System.out.println(h + "h " + m + "m after " + r + "-min meeting");
    }
}

public static void meetings(int h, int m) {
    System.out.println(h + "h " + m + "m start time");
    Random rand = new Random();
    do {
        int r = rand.nextInt(16) + 15;
        m += r;
        if (m >= 60) {         // wrapping case
            h = (h % 12) + 1;
            m = m % 60;
        }
        System.out.println(h + "h " + m + "m after " + r + "-min meeting");
    } while (m > 5 && m < 55);
}

public static void meetings(int h, int m) {
    System.out.println(h + "h " + m + "m start time");
    Random rand = new Random();
    do {
        int r = rand.nextInt(16) + 15;
        int totalTime = h * 60 + m + r;

        if (totalTime / 60 <= 12) {
            h = totalTime / 60;
        } else {
            h = 1;
        }

        m = totalTime % 60;
        System.out.println(h + "h " + m + "m after " + r + "-min meeting");
    } while (m > 5 && m < 55);
}
```