

CSE 142, Spring 2010

Midterm Exam Key

1. Expressions

<u>Expression</u>	<u>Value</u>
$8 - 2 * 3 + (5 - 1)$	6
$10 - 2 + "6" + 3 * 25 + (33 - 3) + 9$	"8675309"
$19 \% 8 == 3 \ \&\& \ 2 == 7 / 3$	true
$1.0 / 2 + (4.5 - 1.5) - 7 / 2$	0.5

2. Parameter Mystery

butters is kenny with friends
butters is stan with cartman
cartman is ike with stan
kyle is friends with butters

3. If/Else Simulation

<u>Method Call</u>	<u>Output</u>
<code>ifElseMystery(12, 12);</code>	12 13
<code>ifElseMystery(7, 4);</code>	14 5
<code>ifElseMystery(5, 8);</code>	10 8
<code>ifElseMystery(3, 42);</code>	5 43

4. Assertions

	<code>a == 0</code>	<code>a % 10 == digit</code>	<code>count > 0</code>
Point A	SOMETIMES	SOMETIMES	NEVER
Point B	NEVER	SOMETIMES	SOMETIMES
Point C	NEVER	ALWAYS	ALWAYS
Point D	NEVER	NEVER	SOMETIMES
Point E	ALWAYS	SOMETIMES	SOMETIMES

5. Programming

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

```
public static boolean dominant(int a, int b, int c) {
    return a > b + c || b > a + c || c > a + b;
}
```

```
public static boolean dominant(int a, int b, int c) {
    if (a + b < c) {
        return true;
    } else if (a + c < b) {
        return true;
    } else if (b + c < a) {
        return true;
    } else {
        return false;
    }
}
```

```
public static boolean dominant(int a, int b, int c) {
    if (c > a + b) {
        return true;
    }
    if (b > a + c) {
        return true;
    }
    if (a > b + c) {
        return true;
    }
    return false;
}
```

```
public static boolean dominant(int a, int b, int c) {
    int max = Math.max(a, Math.max(b, c));
    int min = Math.min(a, Math.min(b, c));
    int mid = a + b + c - max - min;
    if (max > min + mid) {
        return true;
    } else {
        return false;
    }
}
```

```
public static boolean dominant(int a, int b, int c) {
    int sum = a + b + c;
    int max = Math.max(a, Math.max(b, c));
    return sum - max < max;
}
```

```
public static boolean dominant(int a, int b, int c) {
    int big1 = Math.max(a, b);
    int big2 = Math.max(big1, c);
    int small1 = Math.min(a, b);
    int small2 = Math.min(big1, c);
    return big2 > small1 + small2;
}
```

```
public static boolean dominant(int a, int b, int c) {
    int max = Math.max(a, Math.max(b, c));
    int min = Math.min(a, Math.min(b, c));
    int sum = Math.min(Math.min(a+b, a+c), b+c);
    return a > sum || b > sum || c > sum;
}
```

6. Programming

```
public static void invest(double amount, double rate, int years) {
    double total = amount;
    for (int i = 1; i <= years; i++) {
        total += rate / 100.0 * total;
        System.out.printf("After year %d: $%.2f\n", i, total);
    }
    System.out.printf("Total interest earned: $%.2f\n", total - amount);
}

public static void invest(double amount, double rate, int years) {
    double totalInterest = 0.0;
    for (int i = 1; i <= years; i++) {
        double interest = rate / 100.0 * amount;
        amount = amount + interest;
        totalInterest = totalInterest + interest;
        System.out.printf("After year %d: $%.2f\n", i, amount);
    }
    System.out.printf("Total interest earned: $%.2f\n", totalInterest);
}
```

7. Programming

```
public static void lucky(int min) {
    Random r = new Random();
    int count = 0;
    int total = 0;
    while (count < 4) {
        int die = r.nextInt(6) + 1;
        System.out.print(die + " ");
        if (die <= min) {
            count++;
        } else {
            count = 0;
        }
        total++;
    }
    System.out.println();
    System.out.println("Finished after " + total + " rolls.");
}

public static void lucky(int min) {
    Random r = new Random();
    int count = 0;
    int rolls = 0;
    do {
        rolls++;
        int die = r.nextInt(6) + 1;
        if (die <= min) {
            count++;
        } else {
            count = 0;
        }
        System.out.print(die + " ");
    } while (count != 4);
    System.out.println("\nFinished after " + rolls + " rolls.");
}
```