

CS& 141 Sample Midterm Exam #1 Key

1. Expressions

<u>Expression</u>	<u>Value</u>
$3 * 4 + 5 * 6 + 7 * -2$	28
$1.5 * 2.0 + (5.5 / 2) + 5 / 4$	6.75
$23 \% 5 + 31 / 4 \% 3 - 17 \% (16 \% 10)$	-1
<code>"1" + 2 + 3 + "4" + 5 * 6 + "7" + (8 + 9)</code>	"123430717"
$345 / 10 / 3 * 55 / 5 / 6 + 10 / (5 / 2.0)$	24.0
$1 / 2 > 0 \ \ 4 == 9 \% 5 \ \ 1 + 1 < 1 - 1$	true

2. Parameter Mystery

tyler and tv like java
java and tyler like tv
tv and donnie like rugby
hamburger and x like tyler
tyler and java like tyler

3. If/Else Simulation

<u>Method Call</u>	<u>Value Returned</u>
mystery(4, 2)	4
mystery(5, 4)	5
mystery(5, 13)	15
mystery(5, 17)	20
mystery(4, 8)	8

4. While Loop Simulation

<u>Method Call</u>	<u>Output</u>
mystery(5, 0);	5
mystery(3, 2);	1 0 1
mystery(16, 5);	3 2 1 0 1
mystery(80, 9);	8 4 2 1 2 0 2
mystery(1600, 40);	40 19 2 9 0 4 0

5. Assertions

	$y > x$	$z < 0$	$z > 0$
Point A	SOMETIMES	NEVER	NEVER
Point B	NEVER	NEVER	SOMETIMES
Point C	SOMETIMES	NEVER	ALWAYS
Point D	ALWAYS	NEVER	SOMETIMES
Point E	ALWAYS	SOMETIMES	SOMETIMES

6. Programming (five solutions shown)

```
public static boolean hasMidpoint(int a, int b, int c) {
    double mid = (a + b + c) / 3.0;
    if (a == mid || b == mid || c == mid) {
        return true;
    } else {
        return false;
    }
}

public static boolean hasMidpoint(int a, int b, int c) {
    double mid = (a + b + c) / 3.0;
    return (a == mid || b == mid || c == mid);
}

public static boolean hasMidpoint(int a, int b, int c) {
    return (a == (b + c) / 2.0 || b == (a + c) / 2.0 || c == (a + b) / 2.0);
}

public static boolean hasMidpoint(int a, int b, int c) {
    int max = Math.max(a, Math.max(b, c));
    int min = Math.min(a, Math.min(b, c));
    double mid = (max + min) / 2.0;

    return (a == mid || b == mid || c == mid);
}

public static boolean hasMidpoint(int a, int b, int c) {
    return (a - b == b - c || b - a == a - c || a - c == c - b);
}
```

7. Programming (one solution shown)

```
public static void sequenceSum(double limit) {
    if (limit >= 1) {
        System.out.print("1");
        int denominator = 1;
        double sum = 1.0;
        while (sum < limit) {
            denominator++;
            sum += 1.0 / denominator;
            System.out.print(" + 1/" + denominator);
        }
        System.out.printf(" = %.3f\n", sum);
    }
}
```

8. Programming (three solutions shown)

```
public static void favoriteLetter(Scanner console, String letter) {
    System.out.println("Looking for two \"" + letter + "\"" words in a row.");
    int count = 0;
    String word = "";
    while (count < 2) {
        System.out.print("Type a word: ");
        word = console.next();
        if (word.startsWith(letter)) {
            count++;
        } else {
            count = 0;
        }
    }
    System.out.println("\"" + letter + "\" is for \"" + word + "\"");
}

// uses two Strings instead of count, and uses forever/break loop
public static void favoriteLetter(Scanner console, String letter) {
    System.out.println("Looking for two \"" + letter + "\"" words in a row.");
    System.out.print("Type a word: ");
    String word1 = console.next();
    System.out.print("Type a word: ");
    String word2 = console.next();
    while (!(word1.startsWith(letter) && word2.startsWith(letter))) {
        word1 = word2;
        System.out.print("Type a word: ");
        word2 = console.next();
    }
    System.out.println("\"" + letter + "\" is for \"" + word2 + "\"");
}

// uses do/while loop
public static void favoriteLetter(Scanner console, String letter) {
    System.out.println("Looking for two \"" + letter + "\"" words in a row.");
    int count = 0;
    String word;
    do {
        System.out.print("Type a word: ");
        word = console.next();
        if (word.startsWith(letter)) {
            count++;
        } else {
            count = 0;
        }
    } while (count < 2);
    System.out.println("\"" + letter + "\" is for \"" + word + "\"");
}
```