

CSE142 Midterm Key
Autumn 2018

| 1. | Expression | Value |
|----|--|----------|
| | <code>1 + 2 * 3 - 4 * (5 + 6) / 7 + 8 % 9</code> | 9 |
| | <code>20 / 7 * 2.0 + 5.0 / 2 - (1 / 4)</code> | 6.5 |
| | <code>3 * 4 + "3" + 9 * 5 + 6</code> | "123456" |
| | <code>(3 * 5 < 1 + 2) (6 != 5 && !(7 < 7))</code> | true |
| | <code>1 + 1 * (1 - 1) + (1 + 1 + 1) % (1 + 1)</code> | 2 |

2. The program produces the following output:

```
Colonel Mustard in the ballroom with the candlestick
conservatory in the study with the wrench
candlestick in the Colonel Mustard with the Mr. study
Professor Plum in the ballroom with the Mrs. Peacock
```

| 3. | Method Call | Output Produced |
|----|------------------------------------|-----------------|
| | <code>ifElseMystery(3, 1);</code> | 13 1 2 |
| | <code>ifElseMystery(7, 10);</code> | 17 3 5 |
| | <code>ifElseMystery(5, -1);</code> | 5 9 5 |
| | <code>ifElseMystery(1, 2);</code> | 1 12 4 |

| 4. | Method Call | Output Produced |
|----|-------------------------------|-----------------|
| | <code>mystery(5, 20);</code> | 3 5 |
| | <code>mystery(30, 48);</code> | 4 6 |
| | <code>mystery(17, 4);</code> | 7 1 |
| | <code>mystery(9, 9);</code> | 0 9 |

| 5. | a == 0 | a % 10 == digit | count > 0 |
|---------|-----------|-----------------|-----------|
| 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 |

6. Two possible solutions are shown below.

```
public static void generateGrades(Random r, int num) {
    int grade = r.nextInt(21);
    int total = grade;
    int min = grade;

    System.out.print("grades: " + grade);
    for (int i = 1; i < num; i++) {
        grade = r.nextInt(21);
        total += grade;
        if (grade < min) {
            min = grade;
        }
        System.out.print(" " + grade);
    }
    System.out.println();

    System.out.println("min grade: " + min);
    System.out.println("total points: " + total);
    System.out.println("average grade: " + (double)total / num);
}
```

```
public static void generateGrades(Random r, int num) {
    int total = 0;
    int min = 21;

    System.out.print("grades: ");
    for (int i = 0; i < num; i++) {
        int grade = r.nextInt(21);
        total += grade;
        min = Math.min(grade, min);
        System.out.print(grade + " ");
    }
    System.out.println();

    System.out.println("min grade: " + min);
    System.out.println("total points: " + total);
    System.out.println("average grade: " + (double) total / num);
}
```

7. Two possible solutions are shown below.

```
public static double countVotesFor(String candidate, Scanner console) {
    double myVotes = 0;
    int totalVotes = 0;

    System.out.print("Candidate? ");
    String vote = console.nextLine();
    while (!vote.equalsIgnoreCase("done")) {
        if (vote.equalsIgnoreCase(candidate)) {
            myVotes++;
        }
        totalVotes++;

        System.out.print("Candidate? ");
        vote = console.nextLine();
    }

    double pct = myVotes / totalVotes * 100;
    System.out.println(candidate + " received " + pct + "% of the " +
        totalVotes + " votes cast");
    return pct;
}
```

```
public static double countVotesFor(String candidate, Scanner console) {
    int myVotes = 0;
    int notMyVotes = 0;
    String vote = "";

    while (!vote.equalsIgnoreCase("done")) {
        System.out.print("Candidate? ");
        vote = console.nextLine();

        if (vote.equalsIgnoreCase(candidate)) {
            myVotes++;
        } else if (!vote.equalsIgnoreCase("done")) {
            notMyVotes++;
        }
    }

    double pct = 100.0 * myVotes / (myVotes + notMyVotes);
    System.out.println(candidate + " received " + pct + "% of the " +
        (myVotes + notMyVotes) + " votes cast");
    return pct;
}
```

8. Five possible solutions are shown below.

```
public static boolean sameDashes(String str1, String str2) {
    for (int i = 0; i < str1.length(); i++) {
        if (str1.charAt(i) == '-') {
            if (i >= str2.length() || str2.charAt(i) != '-') {
                return false;
            }
        }
    }
    for (int i = 0; i < str2.length(); i++) {
        if (str2.charAt(i) == '-') {
            if (i >= str1.length() || str1.charAt(i) != '-') {
                return false;
            }
        }
    }
    return true;
}
```

```
public static boolean sameDashes(String str1, String str2) {
    String longer = str1;
    String shorter = str2;
    int length = str1.length();

    if (str2.length() > str1.length()) {
        longer = str2;
        shorter = str1;
        length = str2.length();
    }

    for (int i = 0; i < length; i++) {
        if (longer.charAt(i) == '-') {
            if (i >= shorter.length() || shorter.charAt(i) != '-') {
                return false;
            }
        } else if (i < shorter.length() && shorter.charAt(i) == '-') {
            return false;
        }
    }

    return true;
}
```

```

public static boolean sameDashes(String s1, String s2) {
    int length = Math.min(s1.length(), s2.length());
    boolean bad = s1.substring(length).contains("-") ||
        s2.substring(length).contains("-");
    for (int i = 0; i < length; i++) {
        if (s1.charAt(i) == '-' && s2.charAt(i) != '-') {
            bad = true;
        } else if (s1.charAt(i) != '-' && s2.charAt(i) == '-') {
            bad = true;
        }
    }
    return !bad;
}

```

```

public static boolean sameDashes(String str1, String str2) {
    int i1 = str1.indexOf('-');
    int i2 = str2.indexOf('-');
    while (i1 != -1 || i2 != -1) {
        if (i1 != i2) {
            return false;
        }
        str1 = str1.substring(i1 + 1, str1.length());
        str2 = str2.substring(i2 + 1, str2.length());
        i1 = str1.indexOf('-');
        i2 = str2.indexOf('-');
    }
    return true;
}

```

```

public static boolean sameDashes(String s1, String s2) {
    String r1 = "", r2 = "";
    for (int i = 0; i < s1.length(); i++) {
        if (s1.charAt(i) == '-') {
            r1 += "." + i;
        }
    }
    for (int i = 0; i < s2.length(); i++) {
        if (s2.charAt(i) == '-') {
            r2 += "." + i;
        }
    }
    return r1.equals(r2);
}

```