

1. Expressions, 10 points. For each expression in the left-hand column, indicate its value in the right-hand column. Be sure to list a constant of appropriate type (e.g., 7.0 rather than 7 for a double, Strings in quotes).

Expression	Value
4 * (2 + 4) - 3 * 5.5	_____
6 % 4 + 4 % 6 + 6 % 6	_____
15 / 4 / 3.0 - 18 / 5 + (15 / 10.0)	_____
3 + 3 + "3 * 3" + (3 + 3) * 3 % 3	_____
(11 != 7 + 4 3 * 2 <= 39 % 10) == false	_____

2. Parameter Mystery, 12 points. Consider the following program.

```
public class Mystery {
    public static void main(String[] args) {
        String phone = "app";
        String app = "screen";
        String screen = "home";
        String person = "you";
        String me = "person";

        call(me, "you", person);
        call(app, phone, screen);
        me = "(" + me + ")";
        call(me, "me", me);
        call(screen, phone, app);
    }

    public static void call(String person, String app, String phone) {
        System.out.println(app + " calls " + phone + " with " + person);
    }
}
```

List below the output produced by this program.

3. If/Else Simulation, 12 points. Consider the following method.

```
public static void ifElseMystery(int a, int b) {
    if (a == b || a % 2 == 0) {
        a = a - b;
    }
    if (a < b) {
        a = a + 5;
    } else {
        b = 10;
    }
    if (a < b) {
        b--;
        a = b + 3;
    }
    System.out.println(a + " " + b);
}
```

For each call below, indicate what output is produced.

Method Call	Output Produced
ifElseMystery(4, 7);	_____
ifElseMystery(5, 5);	_____
ifElseMystery(8, 3);	_____
ifElseMystery(1, 12);	_____
ifElseMystery(9, 6);	_____
ifElseMystery(3, 4);	_____

4. While Loop Simulation, 12 points. Consider the following method:

```
public static void mystery(int x) {
    int y = 0;
    while (x > 9) {
        y++;
        int z = x % 10;
        x = x / 10 + z;
    }
    System.out.println(x + " " + y);
}
```

For each call below, indicate what output is produced.

Method Call	Output Produced
mystery(6);	_____
mystery(34)	_____
mystery(721);	_____
mystery(65289);	_____

5. Assertions, 15 points. You will identify various assertions as being either always true, never true or sometimes true/sometimes false at various points in program execution. The comments in the method below indicate the points of interest.

```

public static int mystery(int x) {
    if (x <= 0) {
        x = 42;
    }
    int y = 0;
    // Point A
    while (x != 1) {
        // Point B
        if (x % 2 == 0) {
            y++;
            x = x / 2;
            // Point C
        } else {
            x = 3 * x + 1;
            // Point D
        }
    }
    // Point E
    return y;
}

```

Fill in the table below with the words ALWAYS, NEVER or SOMETIMES.

	x == 1	x % 2 == 1	y == 0
Point A			
Point B			
Point C			
Point D			
Point E			

6. Programming, 15 points. Write a static method called `giveProblems` that takes a console scanner and an integer `n` as parameters and that makes up `n` multiplication problems for the user to solve. Each multiplication problem involves choosing two numbers that are each between 1 and 12 inclusive. The user should be prompted for the answer and should be told whether their answer was correct or incorrect. The method should also report the number of problems solved correctly and the number of problems given.

For example, if the following calls are made:

```
Scanner console = new Scanner(System.in);
giveProblems(console, 5);
```

We would expect interaction like the following (user input bold and underlined):

```
10 * 6 =? 60
correct
9 * 9 =? 82
incorrect...the answer was 81
4 * 10 =? 40
correct
3 * 4 =? 13
incorrect...the answer was 12
8 * 5 =? 40
correct
3 of 5 correct
```

In this log, the user entered 5 answers (60, 82, 40, 13, 40). All other text was produced as output by the method. Notice that there is a final line of output indicating the overall results. Your method must exactly reproduce the format of this log.

Your method should construct a `Random` object to use for making up random problems. You may assume that the integer passed to your method is greater than 0 and that the user always enters an integer when prompted.

7. Programming, 15 points. Write a static method called `printSequenceTo` that takes a target value as a parameter and that prints terms from a particular numerical sequence until they add up to a value greater than or equal to the target and that returns the number of terms that were included. For example, if the following calls are made:

```
int n1 = printSequenceTo(3.0);
int n2 = printSequenceTo(5.5);
```

The following output should be produced:

```
1/2 + 2/3 + 3/4 + 4/5 + 5/6 = 3.5500000000000003
1/2 + 2/3 + 3/4 + 4/5 + 5/6 + 6/7 + 7/8 + 8/9 = 6.171031746031746
```

The variable `n1` is set to 5 because it took 5 terms from the sequence to get a sum that is at least 3.0. The variable `n2` would be set to 8 because it took 8 terms to get a sum that is at least 5.5. You are to exactly reproduce the format of this output. You may assume that the target is greater than 0. Notice that the sum is not rounded.

8. Programming, 9 points. Write a static method called `acronym` that takes as a parameter a string containing a phrase and that returns a string that has the acronym for the phrase. For example, the following call:

```
acronym("self contained underwater breathing apparatus")
```

should return "SCUBA". The acronym is formed by combining the capitalized first letters of each word in the phrase. Words in the phrase will be separated by one or more spaces. There might be extra spaces at the beginning or end of the phrase. The string passed as a parameter will not contain any characters other than spaces and letters, and is guaranteed to contain at least one word. Below are several sample calls.

Method Call	Value Returned
<code>acronym(" automatic teller machine ")</code>	"ATM"
<code>acronym("personal identification number")</code>	"PIN"
<code>acronym("computer science")</code>	"CS"
<code>acronym("merry go round")</code>	"MGR"
<code>acronym(" All my Children ")</code>	"AMC"
<code>acronym(" grand old party ")</code>	"GOP"
<code>acronym(" loner ")</code>	"L"

You may use only the string methods included on the cheat sheet and you are not allowed to use a scanner to solve this problem.