CS 1400-03 Introduction to Programming and Problem Solving
Project #1
(Due: 11:59 PM, Monday, 2/15/2021)

First make a new directory named `project` in `cs1400` and then change your working directory to `project`.

```
$cd cs1400
$mkdir project
$cd project
```

Write a Java program, called `Calculations`, that receives data from the user and prints out calculations. Your program will first prompt the user to enter 2 integers, and it will show several calculations using them. Then it will prompt the user for 2 strings, and it will show several calculations on those as well.

You'll need to show difference between integer division and floating-point division. You'll need to cast types to make this work. Notice that when you do the division, if the second number is 0, there will be a `DivideByZeroException`. This is ok. You'll learn how to handle this problem later. We assume the user will not test for this case.

Tips:

You will likely find it easiest to work on only part of the program at a time! Try programming just the part with the numbers first. When you think you have that working, then try adding the strings. It's best to split the problem up into individual pieces. That way, if something goes wrong, you have an idea of where it might be.

When you compile your program, you may receive syntax errors. These are caught by the compiler and listed out with line number and error found. You will learn how to understand what they tell you with experience. All syntax errors must be corrected before the program will run. Examples of syntax errors are spelling mistakes in variable names, missing semicolon, unpaired curly braces, etc. Now, correct all the syntax errors (if any) and then execute the program.

If the output of the program is different from what you expect, this usually indicates a logic error. If the statements are out of order, if there are errors in a formula, or if there are missing steps, the program can still run and give you output, but it may be the wrong output. Examine the program and correct logic errors. Compile the program and execute it again. Repeat until all output matches what is expected.

When you run `Calculations`, you may get a `StringIndexOutOfBoundsException`. This means that you're attempting to access a character outside the bounds of the string. For example, "Hello" contains five characters indexed from 0 to 4, so "Hello".`charAt(5)` will generate `StringIndexOutOfBoundsException`. JVM will perform stack trace to determine what line of code this problem is occurring on. Read your code to find and fix the problem if it happens.

You should test against all the test cases given on the next pages. The bold text shows the user's input. The prompt is not part of your output, that is just to show you how to run the program. Additionally, you should also create your own different categories of tests to run against. However, do not test your program with empty string input (this will be covered later). Make sure your program generates the same output as shown below, including spacing.

Basic Test:

```
fcsang@costello ~/cs1400/project $ java Calculations
Enter a number: 2
Enter a second number: 3
Enter a string: Hello
Enter a second string: World
2 + 3 = 5
2 - 3 = -1
2 * 3 = 6
2 / 3 = 0
2 % 3 = 2
2 / 3 = 0.6666666666666666
First word, uppercase: HELLO
Second word, lowercase: world
First characters: H and W
Last characters: o and d
```

EvenDivisionAndAllUppercase:

```
fcsang@costello ~/cs1400/project $ java Calculations
Enter a number: 10
Enter a second number: 5
Enter a string: HI
Enter a second string: HELLO
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2
10 % 5 = 0
10 / 5 = 2.0
First word, uppercase: HI
Second word, lowercase: hello
First characters: H and H
Last characters: I and O
```

SameNumberAndSameWord:

```
fcsang@costello ~/cs1400/project $ java Calculations
Enter a number: 6
Enter a second number: 6
Enter a string: turtle
Enter a second string: turtle
6 + 6 = 12
6 - 6 = 0
6 * 6 = 36
6 / 6 = 1
6 % 6 = 0
6 / 6 = 1.0
First word, uppercase: TURTLE
Second word, lowercase: turtle
First characters: t and t
Last characters: e and e
```

ZeroFirstAndSingleCharacter:

```
fcsang@costello ~/cs1400/project $ java Calculations
Enter a number: 0
Enter a second number: 3
Enter a string: a
Enter a second string: z
0 + 3 = 3
0 - 3 = -3
0 * 3 = 0
0 / 3 = 0
0 % 3 = 0
0 / 3 = 0.0
First word, uppercase: A
Second word, lowercase: z
First characters: a and z
Last characters: a and z
```

<u>Submission</u>:

In your `cs1400/project` directory, create a Java Source Code file named `Calculations.java`. Your Java program must begin with the comments below and follow the naming and coding conventions posted on Blackboard.

```
      // your name
      // CS1400, section 03
      // Project 1 – calculations on integers and strings
      // date
```

Generate a `script` file `pj1.txt` with appropriate time stamps and the following steps visible:

1. a `pwd` to show the current working directory
2. a `ls -l` to show in long format the files in your `cs1400/project` directory
3. a `cat` to display `Calculations.java`
4. compile `Calculations.java`
5. run `Calculations` <u>five times</u> with the given four test cases plus a test case of your own

Submit the file `pj1.txt` on Gradescope.