# Task Overview:

You are to write two short programs, as described below.  
**WHEN YOU ARE FINISHED** you are to create a Google doc and answer the questions on the final page of this document.

For each program, a sample run will be shown. Your program should match this sample AS CLOSELY AS POSSIBLE. In the example, any **underlined text** represents information which was typed in on the keyboard and did NOT come from a print statement.

# First program:

You are tasked with creating a calculator program. To accomplish this, you will create the Calculator class.

Your program will print a menu of options, and then allow the user to make a selection from:

| Add | Subtract | Multiply | Divide | Square Root | Power | Quit |
| --- | --- | --- | --- | --- | --- | --- |

The user should be allowed to make this choice from a menu, at which point the program will perform the requested action, and then allow them to make another choice until they choose to quit.

The options should perform as follows:

* Add - Should prompt the user for two numbers and then call the add() method. This method will take the two numbers in as parameters and sum them, returning the answer which should then be printed to the user in an informative fashion.
* Subtract - Should work like the Add option, except that it calls the subtract() method. This method should work like add(), but instead subtracts the second parameter from the first parameter.
* Multiply - Should work like the Add option, except that it calls the multiply() method. This method should work like add(), but multiplies instead.
* Divide - Should work like the Add option, except that it calls the divide() method. This method should work like add(), but instead divides the first parameter by the second parameter.
* Square Root - Should prompt the user for a number and then calls the sqrt() method. It should then print the value returned by this method in an informative fashion.
* Power - Should prompt the user for two numbers and then find the value of the first number raised to the power of the second, by using the pow() method. The outcome of this method should then be printed in an informative fashion.

If a **method already exists in the Math class** to perform one of the menu options, your program should use that method instead. Otherwise, you should create your own methods as required. Your program should ensure that if the user enters an **incorrect data type** or an invalid selection, your program **continues to run**.

**Sample output follows on the next page.**

# Sample Output:

[A]dd

[S]ubtract

[M]ultiply

[D]ivide

Square [R]oot

[P]ower

[Q]uit

Please choose what you would like to do: **M**

First number: **3**

Second number: **5**

3 x 5 = 15

[A]dd

[S]ubtract

[M]ultiply

[D]ivide

Square [R]oot

[P]ower

[Q]uit

Please choose what you would like to do: **5**

Invalid selection, please try again

[A]dd

[S]ubtract

[M]ultiply

[D]ivide

Square [R]oot

[P]ower

[Q]uit

Please choose what you would like to do: **P**

First number: **I am not typing a number!**

Please enter a number value!

[A]dd

[S]ubtract

[M]ultiply

[D]ivide

Square [R]oot

[P]ower

[Q]uit

Please choose what you would like to do: **P**

First number: **3**

Second number: **4**

3 to the power of 4 is 81

[A]dd

[S]ubtract

[M]ultiply

[D]ivide

Square [R]oot

[P]ower

[Q]uit

Please choose what you would like to do: **Q**

Program ending!Second program:

You have been tasked with writing an application to help track the scores of players in a gaming league. To accomplish this, you are to write a program called **TeamCalculator**

Your program will ask the user for the name of a team captain which will be stored in a .txt file using their name. It will then open that file and print out the average score for each team member, and the average score of the entire team.

The text files used are designed so that a team member’s name is listed on one line, and then the score for each game they have played is listed on the following lines until the next team member has been reached. Team members may play any number of games, and they do not have to play the same amount.

An example of the .txt file might look as follows:

| Burns  43  2  Skuja  94  126  88  101  Dinkleberg  298  300  255  276  300  288  Carlson  236  248  213  252  287 |
| --- |

# Sample Output:

Please enter the name of the team captain: **Burns**

Opening Burns.txt

Burns average score: 22.50

Skuja average score: 102.25

Dinkleberg average score: 286.17

Carlson average score: 247.2

Team average: 200.41

# Google Doc Questions:

Create a Google Doc to answer the following questions. When completed, save it as a .pdf and submit it along with your program when you turn in your evaluation.

1. **In your first program:** Which methods did you use from the Math class?  
   **For each of these methods**, what would their method header look like (the line that begins the method, NOT the comment block) if you had to write that particular method instead of using it from the Math class?

I used the math class in my power and square root methods.

public static double pow(double num1, double num2)

public static double sqrt(double num1)

1. **In your second program:** Which class did you use to read data from the file?  
   Explain why you choose to use that particular class instead of one of the other classes for reading files that we learned about.

I used the scanner class to read because it reads all types of variables and I could easily turn it into an array list

1. **In your second program:** Did you create any additional methods?  
   **If you did:** Explain how you decided which methods to create.  
   **If you did not:** Explain how in general you would decide when you should create a method in a program you are writing.  
     
   I created a method to return if a string is numeric or not.
2. What types of exceptions did you have to handle in your two programs?  
   Give examples of at least two **significantly different** types of exceptions. For each exception, explain what kinds of situations might cause that exception to occur, and what steps your program will take when that exception happens.

I had to catch the InputMismatchException in order to check if its a numerical value or not

I also catched FileNotFoundException to check if the team captain name exists