



**SOEN 6011 : SOFTWARE ENGINEERING PROCESSES
SUMMER 2021**

SUPER CALCULATOR

PROBLEM - 4
Error Handling, Debugger and Quality Attributes

Authors

Rokeya Begum Keya

Kyle Taylor Lange

Sijie Min

Manimaran Palani

<https://www.overleaf.com/project/610304de4e6b8d24f7c781b6>

PROBLEM 4 - F2: $\tan(x)$

SOEN 6011 - Summer 2021

Software Engineering Processes

Repository address : <https://github.com/Dakatsu/SOEN6011Calculator>

Rokeya Begum Keya

40183615

Error Handling

Error handler in a program works when there is any unexpected error condition in a program.

In our Super Calculator, for my function, if there is any input without integer(degree), the program will notify the user by sending an error message.

numericInputCheck(inputDataString) will provide an error message of type NumberFormatException when the input will not an integer(degree).

If there is a value of $\tan(90^\circ)$ as an input, the program will give an error message as an output "undefined".

Debugger

Description:

In this project I used the Eclipse debugger. When I run my program in Eclipse, it automatically open my program with debug form. In Eclipse debugging, It allows the function to run the code step by step and detect the debug.

PROBLEM 4 - F3: Hyperbolic Sine, $\sinh(x)$

SOEN 6011 - Summer 2021

Software Engineering Processes

<https://www.overleaf.com/project/610304de4e6b8d24f7c781b6>

<https://github.com/Dakatsu/SOEN6011Calculator>

Kyle Taylor Lange

27627696

Repository address :

PROBLEM 4 - F5

SOEN 6011 - Summer 2021

Software Engineering Processes

Repository address : <https://github.com/Dakatsu/SOEN6011Calculator>

Sijie Min

401*****

Team please add your content here

PROBLEM 4 - F7 : x^y

SOEN 6011 - Summer 2021

Software Engineering Processes

Repository address : <https://github.com/Dakatsu/SOEN6011Calculator>

Manimaran Palani

40167543

Problem 4 - Description

This section presents an overview of the source code of the Super Calculator application (F7-Power Function) and the practices followed during the development.

Error Handling

When an exception occurs in Super Calculator, Its considered that the exception is "thrown."

`numericInputCheck(inputDataString)` throws an exception of type `NumberFormatException` when the value entered in the text field is any character other than real numbers.

When an exception is thrown, it is possible to "catch" the exception and prevent it from crashing the program. This is done with a `try..catch` statement in the super calculator.

In simplified form, the syntax for a `try..catch` statement can be:

```
try {
    statements-1
    ...
    numericInputCheck(inputDataString)
    ...
}
catch ( NumberFormatException exception) {
    statements-2
}
```

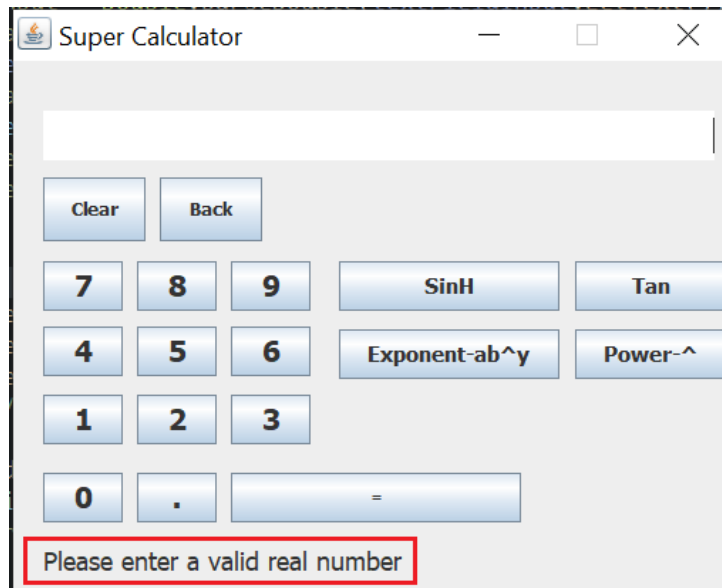


Figure 1: An Error message in the Super Calculator that displays if the input is not accepted.

Debugger

Eclipse has a standard debugger which allows the program to open in debug mode. It supports both step by step debugging and break point based debugging. It offers, breakpoints, checkpoints and multiple views which enhance the experience of debugging.

Advantages

1. Can add any variable that one want to monitor to watch list.
2. Eclipse debugger allows to remote debug a process on any other machine .
3. One can move the current execution while executing .
4. One can step in and out of the code base based on whether it matters or not.
5. The debug perspective offers additional views that can be used to troubleshoot an application like Breakpoints, Variables, Debug, Console etc.
6. The Eclipse Debugging Platform helps developers debug by providing buttons in the toolbar and key binding shortcuts to control program execution.

Disadvantages

1. Debugging with eclipse will become difficult when the execution of a particular function is time bound or if there is usage of sleep statements inside the program.
2. It is difficult to monitor the programs that uses mutli threading .

Quality Attributes

Quality attributes assessed while implementing the algorithm are :

- **Correctness:** Since I used Taylor series for approximating the power function, I had to test with different number of iterations ranging from 10 to 100 to come to a conclusion on the optimum number of iterations to ensure the correctness of the function. Based on my tests, I came to a conclusion that to have minimum difference between expected output and actual output, the number of iterations needed is 13.
- **Efficiency:** As number of iterations increase, it increases the time of execution of the program but less no of iterations have a big impact on the correctness of the program. Hence to have a above average efficiency and a decent correctness I have chosen the no of iterations as 13.
- **Maintainability:** I have refactored the code and included the comments to improve the maintainability and understand-ability of the code. Have a test file to ensure that any changes made doesn't have an effect on the existing functionality of the program.
- **Robustness:** I have handled the situation where the user might give a wrong input such as a string in the place of a double. Hence the program doesn't crash and notifies the user with an appropriate error message. This increases the robustness of the program.
- **Usability:** I am packaging my program in an executable jar file so that other users can use my file without any difficulties. This increases the usability of the program.

Code Quality Check

To check the quality of the Super Calculator's source code, Checkstyle tool is used. This section presents an overview of Checkstyle tool.

- Checkstyle is a development tool to help programmers write Java code that adheres to a coding standard.
- It automates the process of checking Java code to spare humans of this boring (but important) task.
- This makes it ideal for projects that want to enforce a coding standard.
- Checkstyle is highly configurable and can be made to support almost any coding standard.