**Problem Statement**

Write a calculator program in Java that evaluates expressions in a very simple integer expression language. The program takes an input on the command line, computes the result, and prints it to the console.  For example:

Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| **add(1, 2)** | 3 |
| **add(1, mult(2, 3))** | 7 |
| **mult(add(2, 2), div(9, 3))** | 12 |
| **let(a, 5, add(a, a))** | 10 |
| **let(a, 5, let(b, mult(a, 10), add(b, a)))** | 55 |
| **let(a, let(b, 10, add(b, b)), let(b, 20, add(a, b))** | 40 |

**Implementation**

1. Build a Java project as Maven project therefore all the dependencies are in pom.xml file
2. We are using Java 8 as our default enviorment
3. It takes input as a string
4. Dependencies used are
   1. Slf4j for logging
   2. Junit for testing
5. Clone the directory from GitHub

***git clone*** [***https://github.com/Haricharanpanjwani/calculator-challenge.git***](https://github.com/Haricharanpanjwani/calculator-challenge.git)

1. Clean and build the code

***mvn clean install***

1. Compile the code

***mvn compile***

1. To run the main file

***mvn -X exec:java -Dexec.mainClass="com.codiscope.calculator.App" -Dexec.args=" let(a, 5, add(a, a))”***

1. There are 8 Junit test cases to test the code
   1. To test the input string
   2. To test the divide by zero
   3. To test the negative numbers
   4. To test the functionality of add, subtract, multiply, division and let function
2. To run the test file

***mvn –Dtest=com.codiscope.test.TestSuiteRunner test***

1. Logging

Logging layer is implemented to log all relevant information. 3 layers of verbosity: INFO, ERROR and DEBUG is implemented. For this assignment purpose INFO & ERROR logging is included in the code. DEBUG logging is implemented while running Junit test cases.

**Continuous Integration**

For Continuous Integration, I have used Amazon Web Service. It was my first attempt to do this. I have used services like AWS CodePipeline, AWS CodeBuild, AWS CodeDeploy.

In the AWS CodePipeline, it has a hook to GitHub repository, as soon as the code is updated on the github the pipeline will fetch the latest code and it will commit the code in the AWS S3 bucket. After that CodeBuild will build the code from the Amazon S3 and try to push it on the AWS EC2 instance. At EC2 instance you can go the folder and run the application. The only requirement on the EC2 instance is jdk1.8 installed and maven installed.

**Assumption**

1. In Java, when we divide the number by zero, it give +/- Inf when the result is assigned to int but when it is assigned to double it writes Infinity because does not have +/- Inf in it. Therefore, our calculator will return Infinity as output when we divide the number by zero, as we are considering all our input as infinity.
2. In this assignment, we are considering negative numbers as invalid argument. But it can be easily extended for negative numbers by changing our parsing input function