I took about 8 hour time to complete this test, breakdown as follow:

* 1 hour on pseudo code
* 4 hours on writing and testing code
* 2 hours to review and re-factor codes
* 1 hour on summary preparation

There is only 1 class in the project, i.e. com.ideagen.Calculator. Javadoc is prepared for each method.

The calculation is based on basic mathematical calculation precedence, like follow:

1. Brackets
2. Multiply (\*) and divide (/)
3. Add (+) and minus (-)

Brackets:

The codes will first look for expression in brackets, with open and close brackets closest to each other and calculate the expression within first. For example, expression below has 2 sets of brackets, however, as ( 7 – 5 ) is nearest to each other, its expression is calculated first.

10 - ( 2 + 3 \* ( 7 - 5 ) )

Print out from program:

Expression to calculate: 10 - ( 2 + 3 \* ( 7 - 5 ) )

>> 7.0 - 5.0 = 2.0

>> 3.0 \* 2.0 = 6.0

>> 2.0 + 6.0 = 8.0

>> 10.0 - 8.0 = 2.0

10 - ( 2 + 3 \* ( 7 - 5 ) ) = 2.0

Multiply and divide

If no bracket found in expression, next precedence is multiply and divide. For example, expression below has \* operator, hence, 1 \* 3 will be calculated first.

1 + 1 \* 3

Print out from program:

Expression to calculate: 1 + 1 \* 3

>> 1.0 \* 3.0 = 3.0

>> 1.0 + 3.0 = 4.0

1 + 1 \* 3 = 4.0

Add and minus

+ and – are last expression to calculate in expression. Using same sample expression like above, we can see addition is performed last.

Print out from program:

Expression to calculate: 1 + 1 \* 3

>> 1.0 \* 3.0 = 3.0

>> 1.0 + 3.0 = 4.0

1 + 1 \* 3 = 4.0

Others:

If operators’ precedence is same, the calculation is performed left to right. For example, below expression will first calculate 15 / 3, then only (5 \* 4).

15 / 3 \* 4

Print out from program:

Expression to calculate: 15 / 3 \* 4

>> 15.0 / 3.0 = 5.0

>> 5.0 \* 4.0 = 20.0

15 / 3 \* 4 = 20.0