1. **Stack approach pseudo code** : (Assuming all the numbers, braces and operators are separated by space and the operator present within the braces applies to all the values present within the braces)

Note: Expression to be evaluated should be declared as class level variable or a method level variable. Also declare an integer variable and Stack of strings globally.

1. Given expression which is in string format is split into array of strings by using space as split criteria.
2. Array of strings is iterated (outer for loop), keep pushing the strings into global stack (only if the string happens to be a number/integer or an operator (\*,+,-,/)).
3. During iteration, if close braces ( “)” ) encounters, Create a temporary Stack of Integers and start popping the values from global stack and keep pushing those values into temporary stack until and unless you come across an operator.
4. During above operation, if the popped string happens to an operator, call a method which takes operator and Stack of integers as arguments and performs the operation based on the operator passed to it on all the numbers/Integers present in the temporary stack and returns the total value.
5. Once the method returns the calculated value, push the value into the global Stack and also store the value into the global integer variable.
6. Once the outer for loop is completed, the global integer should have the total sum of the expression.
7. **Stack approach java working code :**

**package** com.prop.test.main;

**import** java.util.Stack;

**public** **class** MathCalc {

**public** **static** **void** main(String[] args) {

String needTocalc = "( + 3 10 ( \* 5 2 ) 10 ( - 6 3 ) )";

**int** totalSum = 0;

System.*out*.println("calculating : " + needTocalc);

String[] strArray = needTocalc.split(" ");

Stack<String> strStack = **new** Stack<String>();

**for** (**int** i=0;i<strArray.length;i++) {

**if** (strArray[i].equalsIgnoreCase(")")) {

Stack<Integer> valuesStack = **new** Stack<Integer>();

**while** (!strStack.empty()) {

String str = strStack.pop();

**if** (str.equalsIgnoreCase("\*") || str.equalsIgnoreCase("+") || str.equalsIgnoreCase("-") || str.equalsIgnoreCase("/")) {

totalSum = *calculate*(str.trim(), valuesStack);

strStack.push(""+totalSum);

**break**;

} **else** **if** (!str.equalsIgnoreCase("(")){

valuesStack.push(Integer.*parseInt*(str));

}

}

} **else** {

strStack.push(strArray[i].trim());

}

}

System.*out*.println("SUM - " + totalSum);

}

**public** **static** **int** calculate(String op, Stack<Integer> values) {

**int** total = 0;

**if** (op.equalsIgnoreCase("\*") || op.equalsIgnoreCase("-")) {

total = values.pop();

}

**while** (!values.empty()) {

**if** (op.equalsIgnoreCase("\*")) {

total = total \* values.pop();

} **else** **if** (op.equalsIgnoreCase("+")) {

total = total + values.pop();

} **else** **if** (op.equalsIgnoreCase("-")) {

total = total - values.pop();

} **else** **if** (op.equalsIgnoreCase("/")) {

total = total / values.pop();

}

}

**return** total;

}

}