Start

Declare the following int variables:

len, i, pl, mi, mul, di, S. String S1,

string arr[] & following int arrays:

arrPlus[], arrMin[], arrMul[], arrDiv[]

Create constructor and initialize

int variables to default value.

S1 & len, we will receive argument.

Initialize all arrays to size len.

Create function void.eval()

which will have

the following components:

Check if first &last letter is

+, -,\*, /. If yes "Invalid Expression".

If in other indices +, -, /

is on 2 consenting indices,

then also "Invalid Expression"

Convert String expression to

String array whose individual

members are numbers

in the expression & operators.

For every index that has "/"operator,

divide it's operands & replace ‘/’

with result in the array.

Operands will become null.

Repeat the same logic of

Division for Multiplication, Addition &

Subtraction .If the place before operator

is null, then we go left till we find an

operand. Similarly, if place after operator is

null, we go right to do the same.

Print the final result

Create scanner object for taking input

Input Arithmetic expression

Find length of expression

Pass exp & its length to create Calculator object

Call eval function

End

Pseudo-code:

1.class Calculator:

S1: String

arr: String[]

len: int

arrPlus: int[]

arrMin: int[]

arrMul: int[]

arrDiv: int[]

i: int

pl: int

mi: int

mul: int

di: int

s: int

2.function \_\_init\_\_(self, S1: String, len: int):

self.S1 = S1

self.i = 0

self.pl = 0

self.mi = 0

self.mul = 0

self.di = 0

self.s = 0

self.len = len

self.arr = new String[len]

self.arrPlus = new int[len]

self.arrMin = new int[len]

self.arrMul = new int[len]

self.arrDiv = new int[len]

3.function main(args: String[]):

sc = new Scanner(System.in)

print("Note:- This calculator works only on unsigned numbers. ie No +8 or -7 as integers.\nThe operators allowed are + - \* /")

print("Enter Expression")

S1 = sc.nextLine()

S1 = S1.trim()

len = S1.length()

c1 = new Calculator(S1, len)

c1.eval()

sc.close()

4.function eval(self):

ch = S1.charAt(len - 1)

okay = 0

switch(ch):

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

okay = 1

if okay == 0:

print("Invalid Expression")

return

okay = 0

ch = S1.charAt(0)

switch(ch):

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

okay = 1

if okay == 0:

print("Invalid Expression")

return

5.for i = 1 to len - 2:

chn = S1.charAt(i)

if chn == '+' or chn == '-' or chn == '\*' or chn == '/':

chm = S1.charAt(i+1)

if chm == '+' or chm == '-' or chm == '\*' or chm == '/':

print("Invalid Expression")

return

try:

6.for i = 0 to len:

pos1 = S1.indexOf("+", i)

pos2 = S1.indexOf("-", i)

pos3 = S1.indexOf("\*", i)

pos4 = S1.indexOf("/", i)

minPos = len

if pos1 != -1 and pos1 < minPos:

minPos = pos1

if pos2 != -1 and pos2 < minPos:

minPos = pos2

if pos3 != -1 and pos3 < minPos:

minPos = pos3

7. Initialize arrPlus, arrMin, arrMul, arrDiv, and arr to null or empty values

8. Read input string S1

9. Set len = S1.length() and i = 0

10. While i < len

a. Set pos1 = S1.indexOf("+", i)

b. Set pos2 = S1.indexOf("-", i)

c. Set pos3 = S1.indexOf("\*", i)

d. Set pos4 = S1.indexOf("/", i)

e. Set minPos = the minimum non-negative value among pos1, pos2, pos3, and pos4

f. If minPos is -1, set minPos = len

g. If pos4 != -1 and pos4 < minPos, set minPos = pos4

h. Set num = S1.substring(i, minPos)

i. Set arr[s] = num and increment s

j. If minPos == len, break out of the loop

k. Switch on S1.charAt(minPos) and perform the corresponding operation on arr and arrPlus, arrMin, arrMul, or arrDiv

i. Case '+':

1. Set arrPlus[pl] = s

2. Set arr[s] = "+"

3. Increment pl and s

ii. Case '-':

1. Set arrMin[mi] = s

2. Set arr[s] = "-"

3. Increment mi and s

iii. Case '\*':

1. Set arrMul[mul] = s

2. Set arr[s] = "\*"

3. Increment mul and s

iv. Case '/':

1. Set arrDiv[di] = s

2. Set arr[s] = "/"

3. Increment di and s

l. Set i = minPos

11. Perform division on arrDiv and update arr

12. Perform multiplication on arrMul and update arr

13. Perform addition on arrPlus and update arr

14. Perform subtraction on arrMin and update arr

15. Print the non-null values of arr

16. If any exception occurs, print "Something Invalid" and the stack trace