```
/*****************************************************************************
EGRE 531: Multi-threaded Programming
Lab 1
Programmed by: Luis Barquero
Purpose: Program will calculate the value of an RPN expression.
*****************************************************************************/
#include <iostream>
#include <stack>
#include "Lab1.h"
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include <algorithm>
using namespace std;

Calculator::Calculator() //Default constructor.
{

};

void Calculator::add(double x, double y) //function that will add 2 numbers.
{
    this -> answer = x + y; //this -> answer to access private member answer. Adds up
    two numbers.
};


void Calculator::add(double x) //function that will add a number to previously marked
answer.
{
    stack<double> answer_stack;
    answer_stack.push(this -> answer); //places the answer at the previous answer at
    the top of the stack.
    answer_stack.pop(); //pops out the answer at the top of the stack.
    this -> answer = this -> answer + x; //performs the + operation and stores the
    answer in this -> answer.
    answer_stack.push(this -> answer); //pushes this -> answer back into the stack.
};

void Calculator::sub(double x) // function that will subtract a number from a previous
answer.
{
    stack<double> answer_stack;
    answer_stack.push(this -> answer); //places the answer at the previous answer at
    the top of the stack.
    answer_stack.pop(); //pops out the answer at the top of the stack.
    this -> answer = this -> answer - x; //performs the - operation and stores the
    answer in this -> answer.
    answer_stack.push(this-> answer); //pushes this -> answer back into the stack.
};

void Calculator::sub(double x, double y) // function that will subtract a number from a
previous answer.
{
    stack<double> answer_stack;
    answer_stack.push(this -> answer); //places the answer at the previous answer at
    the top of the stack.
    answer_stack.pop(); //pops out the answer at the top of the stack.
    this -> answer = x - x; //performs the - operation and stores the answer in this ->
    answer.
    answer_stack.push(this-> answer); //pushes this -> answer back into the stack.
};

void Calculator::div(double x, double y) // function that will subtract a number from a
previous answer.
{
    stack<double> answer_stack;
    answer_stack.push(this -> answer); //places the answer at the previous answer at
    the top of the stack.
```

```
58          answer_stack.pop(); //pops out the answer at the top of the stack.
59          this -> answer = x / y; //performs the - operation and stores the answer in this ->
            answer.
60          answer_stack.push(this-> answer); //pushes this -> answer back into the stack.
61      };
62
63      void Calculator::mult(double x, double y) // function that will subtract a number from
        a previous answer.
64      {
65          stack<double> answer_stack;
66          answer_stack.push(this -> answer); //places the answer at the previous answer at
            the top of the stack.
67          answer_stack.pop(); //pops out the answer at the top of the stack.
68          this -> answer = x * y; //performs the - operation and stores the answer in this ->
            answer.
69          answer_stack.push(this-> answer); //pushes this -> answer back into the stack.
70      };
71
72      void Calculator::mult(double x) // function that will multiply a number to a previous
        answer.
73      {
74          stack<double> answer_stack;
75          answer_stack.push(this -> answer);  //places the answer at the previous answer at
            the top of the stack.
76          answer_stack.pop(); //pops out the answer at the top of the stack.
77          this -> answer = this -> answer * x; //performs the * operation and stores the
            answer in this -> answer.
78          answer_stack.push(this-> answer); //pushes this -> answer back into the stack.
79      };
80
81      void Calculator::clear() //function that clears the inputs.
82      {
83          answer_stack.push(0); //pushes this -> answer into the stack.
84          cout << "CLEARED ANSWER" << endl;
85      };
86
87      void Calculator::enter(double x) //function that enters a number to the registry.
88      {
89          answer_stack.push(x); //places this -> answer at the top of the stack.
90      };
91
92      void Calculator::div(double x) // function that divides the answer and the number x.
93      {
94          stack<double> answer_stack;
95          answer_stack.push(this -> answer); //pushes this -> answer into the stack.
96          answer_stack.pop(); //pops out the answer at the top of the stack.
97          this -> answer = (this -> answer) / (x); //performs the / operator and stores the
            answer in this -> answer.
98          answer_stack.push(this-> answer); // pushes this -> answer into the stack.
99      };
100
101     void Calculator:: prt() //print function.
102     {
103         cout << "ANSWER: " << answer_stack.top() << endl;
104         cout << "\n";
105
106     };
107
108     void Calculator:: mult()
109     {
110         double a = 0;
111         double b = 0;
112         a = answer_stack.top(); // sets a = to the top of the stack.
113         answer_stack.pop();//pops out a;
114         b = answer_stack.top();  // sets b = to the top of the stack.
115         answer_stack.pop(); //pops out the top of the stack.
116         answer_stack.push(a*b);// pushes the expression a*b.;
117         cout << "\n" << b << " * " << a << endl;
118     };
```

```
119
120 void Calculator:: add()
121 {
122     double a = 0;
123     double b = 0;
124     a = answer_stack.top(); // sets a = to the top of the stack.
125     answer_stack.pop();//pops out a;
126     b = answer_stack.top(); // sets b = to the top of the stack.
127     answer_stack.push(a+b); // pushes the expression a+b.
128     cout << "\n" << b << " + " << a << endl;
129 };
130
131 void Calculator:: sub()
132 {
133     double a = 0;
134     double b = 0;
135     a = answer_stack.top(); // sets a = to the top of the stack.
136     answer_stack.pop(); //pops out a;
137     b = answer_stack.top(); // sets b = to the top of the stack.
138     answer_stack.push(b-a); // pushes the expression b-a.
139     cout << "\n" << b << " - " << a << endl;
140 };
141
142 void Calculator:: div()
143 {
144     double a = 0;
145     double b = 0;
146     a = answer_stack.top(); // sets a = to the top of the stack.
147     answer_stack.pop(); //pops out a;
148     b = answer_stack.top(); // sets b = to the top of the stack.
149     answer_stack.pop(); //pops out b;
150     answer_stack.push(b/a);// pushes the expression b/a.
151     cout << "\n" << b << " / " << a << endl;
152     if (a == 0) // condition where if a == 0, it will output an error message, since
    dividing by zero is not allowed.
153     {
154         cout << "Error." << endl;
155         exit(1);
156     }
157 };
```