/\*

CIS 200 Lab 10

Stacks

This program evaluates a postfix expression with a stack

\*/

#include<iostream>

#include<stdlib.h>

#include<string>

#include <iomanip>

#include<stack>

using namespace std;

class EmptyStack

{

public:

EmptyStack() { mess = "Attempt to top/pop an empty stack\n"; }

string what() { return mess; }

private:

string mess;

};

void evaluate(stack<double> & nums, char symbol)

{

EmptyStack except;

double a, b;

if (nums.empty())

throw except;

a = nums.top();

nums.pop();

if (nums.empty())

throw except;

b = nums.top();

nums.pop();

switch (symbol)

{

case '+': nums.push(a + b); break;

case '\*': nums.push(a \* b); break;

case '-': nums.push(b - a); break;

case '/': nums.push(b / a); break;

}

}

bool oper(string operand)

{

if (operand.size() == 1 &&

(operand[0] == '+' || operand[0] == '-' || operand[0] == '\*' || operand[0] == '/'))

return true;

else

return false;

}

int main()

{

string operand;

stack<double> values;

EmptyStack except;

cin >> operand;

try{

while (operand != "stop")

{

if (oper(operand))

evaluate(values, operand[0]);

else if (operand[0] == '=')

{

cout << fixed << showpoint << setprecision(5) << values.top() << endl;

if (values.empty())

throw except;

values.pop();

}

else

values.push(atof(operand.c\_str()));

cin >> operand;

}

}

catch (EmptyStack except)

{

cout << except.what() << endl;

}

return 0;

}

/\*

1 3 + =

10 5 / =

10 6 2 + 3 - / =

1.1 2.2 \* =

stop

4.00000

2.00000

2.00000

2.42000

\*/