

Program information in a separate file. This summarizes the project/lab without cluttering the program's source code. (It will also be nice when there is more than one source file later in the semester!)

math.script

```

Script started on Sat Aug 24 12:22:55 2002
$ cat math.info
/*****
 *
 * NAME: Jason James          CLASS: CSC121-00x
 *
 * Lab: MATH!!!              Level: 0
 *
 * Description:
 *
 * This program reads two numbers from the user. Once read in, we
 * perform the four basic math operations (addition, subtraction,
 * multiplication, and division) on them and display the results in a
 * traditional format.
 *
 *****/

$ cat math.cpp
// include input/output streams support code
#include <iostream>

// use identifiers (functions, etc.) from the standard name space
using namespace std;

// main program where it all happens!
int main(void)
{
    double number1, number2; // two numbers input by the user

    // greet user
    cout << "Welcome to the Math Calculation Program!!\n" << endl;

    // prompt for and read in two values
    cout << "Please enter your two numbers: ";
    cin >> number1 >> number2;

    // let the user know we got their values and are working on it
    cout << "Thank you!! You've entered " << number1
    << " and " << number2 << "! Calculating...\n\nDone.\n" << endl;

    // print resulting calculations in a nice format
    cout << number1 << " + " << number2 << " = " << (number1 + number2) << endl
    << number1 << " - " << number2 << " = " << (number1 - number2) << endl
    << number1 << " * " << number2 << " = " << (number1 * number2) << endl
    << number1 << " / " << number2 << " = " << (number1 / number2) << endl;

    // thank the user and say goodbye
    cout << "Thank you for using the MCP!!\n\n"
    << "Endeavor to have a advantageous day!" << endl;

    // return control to the Operating System
    return 0;
}

$ CPP math
math.cpp**

$ ../math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 12 42
Thank you!! You've entered 12 and 42! Calculating..

Done.

12 + 42 = 54
12 - 42 = -30
12 * 42 = 504
12 / 42 = 0.285714
Thank you for using the MCP!!

```

Use your own words -- don't just copy it.

Good indentation between the {} of main.

Not the best variable names...

Good wrapping of long statements.

Program compiles cleanly...no warnings or errors.

Whitespace eases eye strain and logically separates parts of the program's execution.

```

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: -12 42
Thank you!! You've entered -12 and 42! Calculating...

Done.

-12 + 42 = 30
-12 - 42 = -54
-12 * 42 = -504
-12 / 42 = -0.285714
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 12 -42
Thank you!! You've entered 12 and -42! Calculating...

Done.

12 + -42 = -30
12 - -42 = 54
12 * -42 = -504
12 / -42 = -0.285714
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 12
42
Thank you!! You've entered 12 and 42! Calculating...

Done.

12 + 42 = 54
12 - 42 = -30
12 * 42 = 504
12 / 42 = 0.285714
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: -12 -42
Thank you!! You've entered -12 and -42! Calculating...

Done.

-12 + -42 = -54
-12 - -42 = 30
-12 * -42 = 504
-12 / -42 = 0.285714
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: -42 -12
Thank you!! You've entered -42 and -12! Calculating...

```

Tests various combinations of sign and even input spacing.

Poor vertical spacing; program interface seems 'smooshy'.

```

Done.

-42 + -12 = -54
-42 - -12 = -30
-42 * -12 = 504
-42 / -12 = 3.5
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 42 12
Thank you!! You've entered 42 and 12! Calculating...

Done.

42 + 12 = 54
42 - 12 = 30
42 * 12 = 504
42 / 12 = 3.5
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 0 12
Thank you!! You've entered 0 and 12! Calculating...

Done.

0 + 12 = 12
0 - 12 = -12
0 * 12 = 0
0 / 12 = 0
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 12 0
Thank you!! You've entered 12 and 0! Calculating...

Done.

12 + 0 = 12
12 - 0 = 12
12 * 0 = 0
12 / 0 = inf
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: -12 0
Thank you!! You've entered -12 and 0! Calculating...

Done.

-12 + 0 = -12
-12 - 0 = -12
-12 * 0 = -0
-12 / 0 = -inf
Thank you for using the MCP!!

Endeavor to have a advantageous day!

```

Tests highlighting weird but correct situations/values are optional, but not unwelcome.

Make sure you answer all parts of the TPQs -- it can be easy to skip these, sometimes...

```

$ ./math.out
Welcome to the Math Calculation Program!!!

Please enter your two numbers: 0 0
Thank you!! You've entered 0 and 0! Calculating...

Done.

0 + 0 = 0
0 - 0 = 0
0 * 0 = 0
0 / 0 = nan
Thank you for using the MCP!!

Endeavor to have a advantageous day!
$ cat math.tpq
1) How many cin statements do you need in this program?

One. cin skips any intervening spaces and so we can read both variables values with a single cin statement.

2) Should the "welcome" and the "enter" be printed from the same cout statement?

They can be, but they probably shouldn't be. They are two separate logical ideas and so should be two separate statements.

3) What happens if the user types their numbers on separate lines (instead of simply separated by spacing as above)?

That's fine. Any spacing -- even a new-line -- will do to separate the two variables' values.

4) Obviously if your variables for the user's numbers are float or double, the division answer will have decimal places. Can you still get a floating point answer even if you are using whole valued variables for the user's numbers?

Yes. You can use a typecast to make at least one of them floating point for the duration of the division:

long number1, number2;

cout << number1 << " / " << number2 << " = "
    << ( static_cast<double>(number1) / number2 ) << endl;

5) How many variables do you need for this program at a minimum?

Two -- one for each input.

At most?

I couldn't see using more than 6. One for each input and one for each calculation.

6) What happens if the user's values are of different signs? Both negative? In descending order? The same?

All of the calculations work on any pair of real numbers -- just like in mathematics.

7) What happens if the user's first value is a 0?

Everything works fine when the first value is a 0.

Their second value?

All calculations work fine...except division. When dividing by 0, mathematics says the result is undefined. The computer,

```

Seemingly simple answer...

Expanded upon to clarify understanding. That's what TPQs are all about!

```
however, defines the result as either INFINITY (if the dividend  
is positive), -INFINITY (if the dividend is negative), or Not-A-  
Number (if the dividend is also 0).
```

```
$ exit
```

```
Script done on Sat Aug 24 12:25:06 2002
```

I'll take the most recent submission -- no matter how confusing. \*grin\*