Integer math q1

# Revision history

V 1.0: initial release

# usage history

# problem description

1. Have the user enter two integers. Output their quotient to the screen. Come up with test cases where:
   1. The result is correct according to the rules of division that you know
   2. The result is incorrect due to integer division errors
2. Have the user enter two integers. Output their actual quotient to the screen in as many ways as you can think of. Hint: you’ll have to convert something to a double or a float. How many ways are there to do that?

# Solution

Part a:

#include <iostream>

**using** **namespace** std**;**

int main**()**

**{**

int x**,** y**;**

cout **<<** "Enter two integers!" **<<** endl**;**

cin **>>** x >> y**;**

cout **<<** x/y**;**

**}**

Part b:  
There are many ways to write this code. Three suggestions are shown below.

#include <iostream>

**using** **namespace** std**;**

int main**()**

**{**

int x**,** y**;**

cout **<<** "Enter two integers!"**<<** endl**;**

cin **>>** x **>>** y**;**

//we can cast implicitly:

cout **<<** 1.0**\***x**/**y **<<** endl**;**

//we can cast explicitly

cout **<<** **(**double**)**x**/**y **<<** endl**;**

//we can store additional variables

double dX **=** x**,** dY **=** y**;**

cout **<<** dX**/**dY**;**

**}**

# suggested test cases

* Testing any case where x is a multiple of y will produce the correct result. For instance, x = 10, y = 2, will produce 5.
* Testing any case where x is NOT a multiple of y will induce integer division errors. Test x = 10, y = 3, for example, to get the answer 3.
* Code for part b should be re-tested using the same test cases as for part a, to compare answers.

# required topics

* Integer division
* Designing good test cases
* Casting rules