# Integer, floating-point division

**The goal for this exercise** is to extend what was done in the first lesson, adding some more detail to how math is done in C#

Prior to this exercise, you should read about the two types of division in C# (and most other programming languages) – integer division, and the normal, floating-point division.

**What you need to do for this exercise:**  Download the starter project for the PCEs (if you haven’t already), and add code to the Integer\_Vs\_Real\_Division. RunExercise() method that prints out the result of doing **integer division** where

1. the result is exact (i.e., 6/3)
2. the result will drop the remainder (i.e., 7/3)

Next, **do the same divisions using floating-point division**, and print out those results. Do this using both a float and then again using double.

Include a short comment explaining why your code doesn’t always generate an exact answer (e.g., why does 7/3 not produce a number like 2.3333, but 7.0/3.0 does?). Focus your answer on explaining why (syntactically) C# decides to give you the rounded-off answer.

Include a short comment explaining why using the float doesn’t get you exactly the same answer as using the double data type.

|  |
| --- |
| **Note For Studying:**  Please keep in mind that many of the C# language topics we'll be looking at tend to have a lot less stuff to memorize, but each item can be used in lots of different places – use this to optimize your study strategies for the C# language! |