*Notes 10/1*

*Int* can hold numbers from -2billion to 2billion. Double can hold 0, and positive/negative 10^-308 to 10^308, but can only hold about 15 significant digits. Floats can hold 10^-38 to 10^38 with 7 sig figs.

Usually use double rather than float, because 7 sig figs aren’t enough for many applications and the memory saved with float is negligible.

*cout.setf(ios::fixed)  
cout.precision(2)*

^This bit first sets the program to display numbers in standard floating point notation, rather than scientific notation. The second line sets the decimal precision to 2 decimal places.

C++ uses standard arithmetic precedence – parenthesis before multiplication/division before addition/subtraction. Be careful – 6 + 1 – 3 isn’t the same as 6 + (1 – 3)

14.3 / 5.0 🡺 2.86. Double / double = answer is a double.  
14.3 / 5 🡺 2.86. Double / int = convert int to double, divide, answer is a double.  
14 / 5.0 🡺 2.86. Int / double = convert int to double, divide, answer is double.  
14 / 5 🡺 2. Int / int = answer in an int. Will *drop off* decimal points. Be careful with this

Another operator is the % (remainder.)

14 % 5 🡺 4. The remainder when dividing 14 by 5 is 4, so the answer is the int 4.

You can initialize and declare a variable in one line in C++. For example, instead of  
  
*double totalPay  
totalPay = amountWorked \* payRate*

you can type  
  
*double totalPay = amountWorked \* payRate*

for the same effect.

Compilers can sometimes give you warnings about uninitialized variables that you use in equations. If you declare a variable and then use it without assigning a value to it, you will get garbage values and the program may crash; sometimes the compiler will catch you trying to do this and slap you for being dumb, but a lot of times it won’t catch it and your program will bug out. Be careful about using variables and make sure you initialize them before invoking them later on.