

```
#include <iostream>
#include <typeinfo>
#include <iomanip>

using namespace std;

int main(){
    //Basado en "programa 2.6" (p64 del libro)
    cout<<"15.0 mas 2.0 es igual a: "<<(15.0+2.0)<<endl
        <<"15.0 menos 2.0 es igual a: "<<(15.0-2.0)<<endl
        <<"15.0 por 2.0 es igual a: "<<(15.0*2.0)<<endl
        <<"15.0 dividido 2.0 es igual a: "<<(15.0/2.0)<<endl<<endl;

    system("pause");
    cout<<endl<<endl;
    /*
    En operaciones mixtas el tipo de datos de cada operación se determina por las
    siguientes reglas:
        1. Si ambos operandos son enteros, el resultado de
        la operación es un entero.
        2. Si un operando es un valor real, el resultado de
        la operación es un valor de
        precisión doble.
    */

    /*
    Ahora bien. En los ejemplos anteriores algunos resultados no los mostraba con el
    punto decimal a pesar de ser un resultado de precision doble.
    Este pequeño codigo es solo para mostrar que internamente
    la computadora interpreta al resultado como un valor doble o entero suguiendo
    las reglas antes mencionadas
    */

    cout <<"Resultado: "<<15.0+2.0<<typeid(15.0+2.0).name()<<endl;
    cout <<"Resultado: "<<15+2<<typeid(15+2).name()<<endl<<endl;

    system("pause");
    cout<<endl<<endl;
    /*
    Podemos ver algo similar inicializando variables, y usando "fixed" para mostrar
    los decimales. (Basado en p65)
    */

    int a=2;
    double b=15;
    int c=15;

    cout<<setprecision(4)<<fixed<<"15/2= "<<b/a<<endl;
    cout<<setprecision(4)<<fixed<<"15/2= "<<c/a<<endl<<endl; //Notar que en este caso trunca.

    system("pause");
    return 0;
}
```