Tutorial / Exercise – References

These practical exercises are designed to re-inforce the lecture material.

Exercises:

1. Basic Reference use:
    a. Create a new C++ project in Visual Studio
    b. In main() create an integer variable and initialize it to 10;
    c. Display the contents of the variable in the console.
    d. Create a reference to the integer using *int&*
    e. Display the contents of the reference in the console.
    f. Change the contents of the variable by directly changing the variable and using the reference
    g. Create another reference to the same integer and prove to yourself that both references are referring to the same int.
    h. Try creating a reference which doesn't refer to anything and try to make an existing reference refer to something else. Also try making a reference to an *int* refer to a float.
2. Functions and references:
    a. Create a new function which takes two integers as parameters. In the function body change the value of the integers.
    b. Call the function from main and display the values of the integers which are passed into the function before and after it's called.
    c. Replace the *int* parameters with *int&* and repeat step b. You should notice that the values now change
    d. Make sure you understand what is going on before continuing
3. Reference and *Structs:*
    a. Create a structure which has a number of member variables
    b. Modify the function you created earlier to take the structure as a parameter
    c. Modify the value of some of the structure members inside the function
    d. Display the values of the *structs* member values before and after the function is called
    e. Replace the function parameter types with a ref to the struct and repeat the previous step
4. Use of *const* with references
    a. In the previous example we passed a reference to a struct into a function. This is more efficient than passing in the entire struct (why?) but the downside is that we can accidentally modify the contents of the struct inside the function. Modify the function so that it can't change the value of the variables within the structure, there should be a compiler error if you try.

5. Questions:
    a. Why is it more efficient to pass a reference to a *struct* into a function rather than the structure itself?
    b. Is it more efficient to pass the reference to an *int* into a function than the int itself? Is there any advantage in (*const int*& foo) over (*int* foo)?