**YouTube tutorial 60 – Class Templates**

https://www.youtube.com/watch?v=U2QvTsMvWmM&t=387s

#include "stdafx.h"

#include <iostream>

using namespace std;

template <class T> //Before every single function, you need to have this line

class Bucky {

T first, second;

public:

Bucky(T a, T b) {

first = a; second = b;

}

T bigger();

};

template <class T>

T Bucky<T>::bigger() { //<-----To let the compiler know that we'll be using the

// class' same data type as the class

return (first>second?first:second);

}

int main()

{

Bucky <int>buck(10, 32); //When working with classes, we explicitly tell

//which data type we're gonna be using

cout << buck.bigger() << endl;

return 0;

}

**Result:**

32

**Important notes:**

* Keep in mind that you don’t need to explicitly write the data type for normal functions
* Only in this case, to tell the compiler we’ll be using the same data type as in class, T Bucky<T>::bigger() was used