**YouTube tutorial 58-59 – Function templates + with Multiple Parameters**

https://www.youtube.com/watch?v=W0aoAm6eYSk&index=58&list=PLAE85DE8440AA6B83

#include "stdafx.h"

#include <iostream>

using namespace std;

template <class bucky> //Can take any variable type only once

bucky addNumbers(bucky a, bucky b) {

return a + b;

}

template <class A, class T> //Can take 2 different any variable type

A addDiffNumbers(A a, T b) {

return a + b;

}

int main()

{

int a = 2;

int b = a;

double c = 2.5;

double d = c;

cout << "Sum of doubles is: " << addNumbers(c, d) << endl;

cout << "Sum of integers is: " << addNumbers(a, b) << endl;

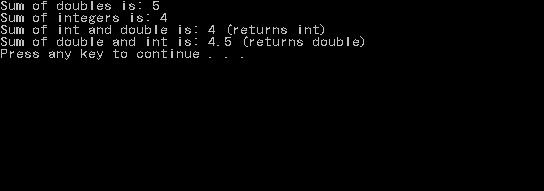
cout << "Sum of int and double is: " << addDiffNumbers(a, c) << " (returns int)" << endl;

cout << "Sum of double and int is: " << addDiffNumbers(c, a) << " (returns double)" << endl;

return 0;

}

**Result:**



**Important notes:**

* A good programming habit is using one character for each generic variable(T, V, H etc.).