**YouTube tutorial 61 – Template Specialization**

https://www.youtube.com/watch?v=8kjVFp-Y4GA&index=61&list=PLAE85DE8440AA6B83

#include "stdafx.h"

#include <iostream>

using namespace std;

template <class T>

class Spunky { //no data type is written on this line

public:

Spunky(T x) {

cout << x << " is not a character!" << endl;

}

};

template <>

class Spunky<char> { //<char> is written in this case, to specify the data type

public:

Spunky(char x) {

cout << x << " is a character!" << endl;

}

};

int main()

{

Spunky <int>spunk(5);

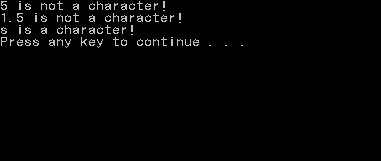
Spunky <double>spunk2(1.5);

Spunky <char>spunk3('s');

return 0;

}

**Result:**



**YouTube tutorial 62 – Exception**

https://www.youtube.com/watch?v=mFAaqmj399I&list=PLAE85DE8440AA6B83&index=62

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

try {

int momsAge = 30;

int sonsAge = 34;

if (sonsAge > momsAge)

throw 99;

cout << "hello" << endl; //This won’t print

}

catch (int x) {

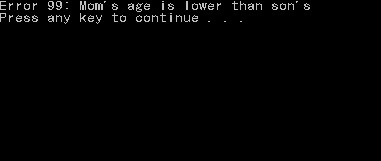
cout << "Error " << x << ": Mom's age is lower than son's" << endl;

}

return 0;

}

**Result:**



**Important notes:**

* One difference is that unlike in java, the exception’s takes int as its data type.
* Writing *“throw”* partly means getting out of the “*try*” block.