CSCI251/CSCI851 Autumn-2020
Advanced Programming (**LT11**)

Lecture Tutorial 11

From the Lab: The utility nm

- This isn't something you would be examined about but …
- … it's an example of a utility in the Linux environment that can help.
- Being familiar with the environment that you work in can make you more efficient…
- But, like Vidiu Plato said:
  "I don't care if it works on your machine! We are not shipping your machine!", …
- … don't assume users of your program/code have that same environment.
- Template functions won't be instantiated if they are not requested.

- Debug-A: The missing constructor can be something like this …

```
template<typename T>
CSL<T>::CSL(T *d, int s): data(d),size(s){}
```

A template exercise : doubled.cpp

```
template <typename T>
T doubled(T arg)
{
        return arg + arg;
}
```

- For this to work with a particular type, + needs to be defined for that type.
- So if it's an ADT (abstract data type), our only class most likely, we will need to overload operator+.

```
int main()
{
        int I=2;
        float F=3.3;
        char C=‘ ';
        string S="cat";
        X x(4);

        cout << doubled(I) << endl;
        cout << doubled(F) << endl;
        cout << doubled(C) << endl;
        cout << doubled(S) << endl;
        cout << doubled(x) << endl;

        return 0;
}
```

```
4
6.6
@
catcat
8
```

- The main to test can be something like the above.
  - We will get to X in a moment, that’s the ADT.
- A fair few people said this didn’t work with char, it does just maybe not in the way you expect.
  - Ascii space is 32 in decimal, doubles to 64 which is @.

```
class X{
friend ostream& operator<<(ostream& sOut, const X &x);
private:
        int data;
public:
        X(){};
        X(int arg): data(arg){}
        X operator+(const X& arg) const;
};

X X::operator+(const X& arg) const
{
        X temp;
        temp.data=this->data+arg.data;
        return temp;
}

ostream& operator<<(ostream& sOut, const X &x)
{
   sOut << x.data;
   return sOut;
}
```

- You don't have to overload `operator<<` for `X` for `doubled(X)` to work, but it means the use in main can be same for output.