Template Usage Example

Files for example of template instantiation ¶

SConstruct

```
# -*- python -*-
#
# Setup our environment
#
import lsst.SConsUtils as scons

env = scons.makeEnv("Foo",
                    r"$HeadURL$",
                    [])

env.SharedLibrary('Foo', "Foo.cc")
env.Program(["main.cc"], LIBS="Foo", LIBPATH=".")
```

Foo.h

```
#if !defined(FOO_H)
#define FOO_H 1

#include <cstddef>

template<typename T>
class Foo {
public:
    explicit Foo();
    explicit Foo(T x);
    ~Foo();

    inline T getVal() const;
    void setVal(T val);
private:
    T _val;
};
//
// Inlined functions need to be in the .h file (maybe in the class definition itself)
//
template<typename T>
inline T Foo<T>::getVal() const { return _val; }
//
// Declare a templated function
//
template<typename T>
size_t mySizeof(void);
//
// Provide a declaration for explicitly-instantiated version[s] of mySizeof()
// This is only needed if we provide enough information here for the compiler
// to instantiate the function/class
//
// It's non-compliant code (but see C++ standards proposal N1897), and stops
// the compiler instantiating the function in each file, and leaving it up
// to the linker to clean up the mess
//
#if 0
extern template size_t mySizeof<float>(void);
#endif

#endif
```

Foo.cc

```
 /// Implementation of templated class Foo
#include "Foo.h"

template<typename T>
Foo<T>::Foo() : _val(0) {}

template<typename T>
Foo<T>::Foo(T val) : _val(val) {}

template<typename T>
Foo<T>::~Foo() {}

#if 0
template<typename T> T Foo<T>::getVal() const; // inline in Foo.h
#endif

template<typename T>
void Foo<T>::setVal(T val) { _val = val; }
//
// And a templated function
//
template<typename T>
size_t mySizeof() { return sizeof(T); }

//
// Explicit instantiations
//
template class Foo<float>;
template class Foo<int>;

template size_t mySizeof<float>();
```

main.cc

```
 #include <iostream>
#include "Foo.h"

int main() {
    Foo<int> i;
    Foo<float> x(1.234);

    std::cout << "i = " << i.getVal() << ", " << x.getVal() << std::endl;

    i.setVal(10);
    std::cout << "i = " << i.getVal() << std::endl;

    std::cout << "mySizeof(float) = " << mySizeof<float>() << std::endl;
}
```