Generic Programming

Definition: Generic Programming

Programming style in which specific Types are not specified initially, but instantiated when needed.

Templates

In C++ generic programming is implemented using Templates and instantiated at compile time.

SciNet compute • calcul CANADA

Templates

Syntax

Functions

```
template < typename T > funcname (T &a)
```

Classes

```
template < class T >
class classname {
   private:
      T a, b;
   public:
      classname ();
      ...
      memberfunction(T &c, T &d);
};
```

Templates

Example (Double Matrix Class)

```
class matrix {
   private:
      int rows, cols;
      double *elements;
   public:
      matrix(matrix& m);
      ...
      void fill(double value);
      matrix& operator= (const matrix &m)
};
```

Templates

Example (Templated Matrix Class)

```
template <typename T>
class matrix {
   private:
      int rows, cols;
      T *elements;
   public:
      matrix(matrix<T>& m);
      ...
      void fill(T value);
      matrix<T>& operator= (const matrix<T>& m);
};
```

```
template <typename T>
matrix<T>::matrix() {
   rows = 0; cols = 0;
   elements = new T[0];
}
```

Templates

Example (Calling Templated Class)

```
matrix<double> A(5,5);
A.fill(0.0);
A(0,0) = 1.3;
A(4,3) = -5.2;

matrix<int> B(5,5);
B.fill(33);
B(0,0) = 1;
B(4,3) = 2;
```

Templates

Explicit Instantiation

- Can override generic template abstract-type instantiation for a specific concrete-type.
- Similar to derived class override of base class member function.

```
template<> matrix<double>::fill(double value){
   for (int i=0; i < rows; i++) {
      for (int j=0; j < cols; j++){
         std::cout<<" I'm used for doubles ";
         (*this)(i,j)=value;
      }
   }
}
```

Templates

Gotcha:

- Compile times can be significantly longer.
- Large header files.
- Debugging can be a pain.
- Syntax can get complicated.

HANDS-ON:

Template your matrix class.