

Polymorphism

Template functions

```
template <class Type>
Type max(Type i, Type j) {
    if (i >= j)
        return(i) ;
    else
        return(j) ;
}
```

**When is the above function invoked?
What if there are other max functions?**

Class Templates

An example from the textbook

```
template <class Elem> class Link {
public:
    Elem element;          // Value for this node
    Link *next;           // Pointer to next node in list
    Link(const Elem& elemval, Link* nextval =NULL)
        { element = elemval; next = nextval; }
    Link(Link* nextval =NULL) { next = nextval; }
};
```

Derived classes with templates

An example from the textbook

```
class AList : public List<Elem> {
private:
    int maxSize;          // Maximum size of list
    int listSize;         // Actual number of elements in list
    int fence;           // Position of fence
    Elem* listArray;      // Array holding list elements
public:
    AList(int size=DefaultListSize) { // Constructor
        maxSize = size;
```

Abstract Base Class

```
template <class Elem> class List {
public:
    // Reinitialize the list. The client is responsible for
    // reclaiming the storage used by the list elements.
    virtual void clear() = 0;
    // Insert an element at the front of the right partition.
    // Return true if successful, false if the list is full.
    virtual bool insert(const Elem&) = 0;
    // Append an element at the end of the right partition.
    // Return true if successful, false if the list is full.
    virtual bool append(const Elem&) = 0;
```

Template instantiation

When does the compiler generate the code?
C++ tries to do the work at compile time!

Virtual functions

Use in derived classes
Pure virtual functions
Used in abstract interfaces

[STL – Standard Template Library](#)