

Templates and Inheritance

Templates & Inheritance 1

Options:

- a template may be derived from another template.
- a non-template class may be derived from a template.
- a template may be derived from a non-template.
- templates may use multiple inheritance.

We will briefly examine the first two cases here; the remaining cases are left to the reader.

Computer Science Dept Va Tech April 2001 OO Software Design and Construction ©2001 McQuain WD & Keller BJ

A Base Template...

Templates & Inheritance 2

Recall the queue template QueueT from earlier notes:

```
const int Size = 100;

template <class Foo> class QueueT {

private:
    Foo buffer[Size];
    int Head, Tail, Count;

public:
    QueueT();
    void Enqueue(Foo Item);
    Foo Dequeue();
    int getSize() const;
    bool isEmpty() const;
    bool isFull() const;
    ~QueueT();
};
```

Computer Science Dept Va Tech April 2001 OO Software Design and Construction ©2001 McQuain WD & Keller BJ

... and a Derived Template

Templates & Inheritance 3

We can derive an extended queue template that adds the ability to “peek” at the element at the front of the queue:

```
template <class Foo>
class InspectableQueue : public Queue<Foo> {

public:
    InspectableQueue();
    Foo Inspect();
    ~InspectableQueue();
};
```

Computer Science Dept Va Tech April 2001 OO Software Design and Construction ©2001 McQuain WD & Keller BJ

Using InspectableQueue Template

Templates & Inheritance 4

```
InspectableQueue<Location> Path;
Location loc1(...);
Location loc2(...);
...
Path.Enqueue(loc1); // base class method
Path.Enqueue(loc2); // base class method
...
Location Front = Path.Dequeue(); // base class method
Location newFront = Path.Inspect(); // derived class method
```

Computer Science Dept Va Tech April 2001 OO Software Design and Construction ©2001 McQuain WD & Keller BJ

NonTemplate Derived from Template

Templates & Inheritance 5

Recalling the linked list template, `LinkedListT`, discussed earlier, we can derive a `Polygon` class from it:

```
class Polygon : public LinkedListT<Location> {
public:
    Polygon();
    void WidderShins();           // sort points
    void Print(ostream& Canvas); // draw itself
};
```

Note that the template `LinkedListT` is elaborated with a specific type.

Computer Science Dept Va Tech April 2001

OO Software Design and Construction

©2001 McQuain WD & Keller BJ

Using the Polygon Class

Templates & Inheritance 6

```
Polygon Hex;
...
Hex.AppendNode(Location(20,20)); // inherited from
Hex.AppendNode(Location(30,30)); // LinkedListT template

// insert other locations

Hex.Print(cout); // derived class method
```

Computer Science Dept Va Tech April 2001

OO Software Design and Construction

©2001 McQuain WD & Keller BJ