Topic 3

1. Inheritance hierarchies
2. Implementing derived classes
3. Overriding member functions
4. Virtual functions and polymorphism

Overriding Member Functions

Recall that our design requires that the display member function be rewritten in the **`ChoiceQuestion`** class.

This is called *overriding* a member function.

`ChoiceQuestion`'s **`display`** method needs to:

1. Display the question text.
2. Display the answer choices.

Overriding Member Functions (2)

The second part is easy – just a loop to print out the derived-class data members

```
int ChoiceQuestion::display() const
{
   // Display the question text
   ...
   // Display the answer choices
   for (int i = 0; i < choices.size(); i++)
   {
      cout << i + 1 << ": "
         << choices[i] << endl;
   }
}
```

Overriding Member Functions (3)

The first part seems easy too – call the **`display`** in Question:

```
int ChoiceQuestion::display() const
{
   // Display the question text
   display(); /* attempting to call Question's
   display function */
   // Display the answer choices
   ...
```

UNFORTUNATELY, this results in a recursive function with `display()` called from `ChoiceQuestion` by itself, instead of the base class `display()`.

Overriding Member Functions (4)

The solution is to do what you said in the first place:

Call **Question**'s `display` by prefixing the function name with `Question::`

```
int ChoiceQuestion::display() const
{
   // Display the question text
   Question::display();
   // Display the answer choices
   ...
}
```

Note that the override function does not necessarily have to call the base class's function, unless it must use the base's data members.

A Program With `ChoiceQuestion` (1)

```
// sec03/demo.cpp
#include <iostream>
#include <sstream>
#include <vector>
#include "question.h"

class ChoiceQuestion : public Question
{
public:
   /**
      Constructs a choice question with no choices.
   */
   ChoiceQuestion();
```

A Program With `ChoiceQuestion` (2)

```
   /**
      Adds an answer choice to this question.
      @param choice the choice to add
      @param correct true if this is the correct choice,
      false otherwise
   */
   void add_choice(string choice, bool correct);
   void display() const;
private:
   vector<string> choices;
};

ChoiceQuestion::ChoiceQuestion()
{
}
```

A Program With `ChoiceQuestion` (3)

```
void ChoiceQuestion::add_choice(string choice, bool correct)
{
   choices.push_back(choice);
   if (correct)
   {
      // Convert choices.size() to string
     string num_str = to_string(choices.size());
     set_answer(num_str);
   }
}
void ChoiceQuestion::display() const
{
   // Display the question text
   Question::display();

   // Display the answer choices
   for (int i = 0; i < choices.size(); i++)
   {
      cout << i + 1 << ": " << choices[i] << endl;
   }
}
```

A Program With `ChoiceQuestion` (4)

```
int main()
{
   string response;
   cout << boolalpha;

   // Ask a basic question
   Question q1;
   q1.set_text("Who was the inventor of C++?");
   q1.set_answer("Bjarne Stroustrup");

   q1.display();
   cout << "Your answer: ";
   getline(cin, response);
   cout << q1.check_answer(response) << endl;
```

A Program With `ChoiceQuestion` (5)

```
   // Ask a choice question

   ChoiceQuestion q2;
   q2.set_text("In which country was the inventor of C++ born?");
   q2.add_choice("Australia", false);
   q2.add_choice("Denmark", true);
   q2.add_choice("Korea", false);
   q2.add_choice("United States", false);

   q2.display();
   cout << "Your answer: ";
   getline(cin, response);
   cout << q2.check_answer(response) << endl;

   return 0;
}
```

A Program With `ChoiceQuestion (6)`

Program Run

```
Who was the inventor of C++?
Your answer: Bjarne Stroustrup
true
In which country was the inventor of C++ born?
1: Australia
2: Denmark
3: Korea
4: United States
Your answer: 2
true
```

Common Error: Forgetting the Base-Class Name

Don’t forget to use the

```
BaseClass::memberFunction
```

notation when you want to call a member function from the base class, when you are writing:

```
DerivedClass::memberFunction
```