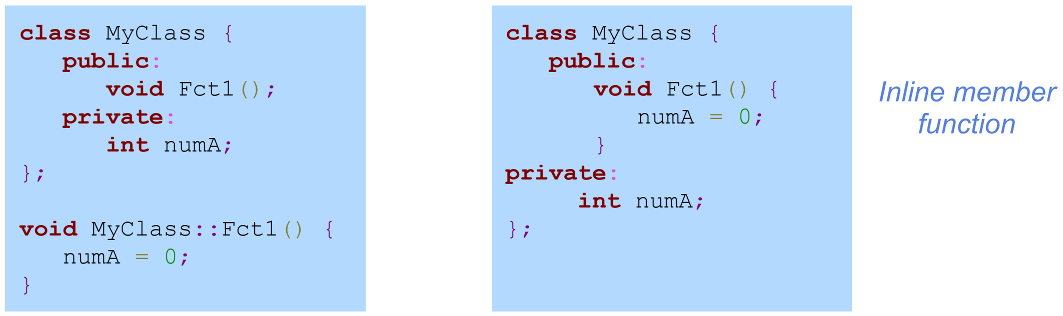
7.4 Inline member functions

Inline member functions

A member function's definition may appear within the class definition, known as an ***inline member function***. Programmers may inline short function definitions to yield more compact code, keeping longer function definitions outside the class definition to avoid clutter.

PARTICIPATION ACTIVITY 7.4.1: Inline member functions.

Start 2x speed



```
class MyClass {
   public:
      void Fct1();
   private:
      int numA;
};

void MyClass::Fct1() {
   numA = 0;
}
```

```
class MyClass {
   public:
      void Fct1() {
         numA = 0;
      }
   private:
      int numA;
};
```

Feedback?

Figure 7.4.1: A class with two inline member functions.

```
My favorite
restaurants:
Central Deli -- 4
Friends Cafe -- 5
```

```
#include <iostream>
#include <string>
using namespace std;

class Restaurant {                            // Info about a
restaurant
   public:
      void SetName(string restaurantName) { // Sets the
restuarant's name
         name = restaurantName;
      }
      void SetRating(int userRating) {      // Sets the rating (1-
5, with 5 best)
         rating = userRating;
      }
      void Print();                         // Prints name and
rating on one line

   private:
      string name;
      int rating;
};

// Prints name and rating on one line
void Restaurant::Print() {
   cout << name << " -- " << rating << endl;
}

int main() {
   Restaurant favLunchPlace;
   Restaurant favDinnerPlace;

   favLunchPlace.SetName("Central Deli");
   favLunchPlace.SetRating(4);

   favDinnerPlace.SetName("Friends Cafe");
   favDinnerPlace.SetRating(5);

   cout << "My favorite restaurants: " << endl;
   favLunchPlace.Print();
   favDinnerPlace.Print();

   return 0;
}
```

Feedback?

PARTICIPATION ACTIVITY | 7.4.2: Inline member functions.

Consider the example above.

1) Member function SetName() was defined _____.

[✓ inlined]

not inlined

Correct

The function's definition has just one statement, so fits easily in the class definition.

2) Inline member function SetRating() _____ a semicolon after the function name and parentheses, just like a function declaration.

- ○ has
- [✓ does not have]

Correct

A function declaration follows the name and parentheses with a semicolon. In contrast, a function definition has an opening brace, the function's statements, and a closing brace.

3) Member function Print() was _____.

- ○ inlined
- [✓ not inlined]

Correct

Member function Print() was only declared in the class definition. The function's definition appears outside the class definition. Note: This function was not inline to demonstrate that some functions may be inlined and some not; the function is short enough to be inlined.

4) A function with a long definition likely ____ be inlined.

- ○ should
- [✓ should not]

Correct

Short functions (having just a few statements) can be inlined without cluttering the class definition. Longer functions should be kept separate, else the many statements prevent a class user from easily seeing the list of available public member functions.

5) A function defined as an inline member function _____ also have a definition outside the class as well.

- ○ may
- [✓ may not]

Correct

A function can only have one definition. Else, the compiler doesn't know which definition to use. Having two definitions yields a compiler error.

Feedback?

Exception to variables being declared before used

Normally, items like variables must be declared before being used, but this rule does not apply within a class definition. Ex: Above, SetRating() accesses rating, even though rating is declared a few lines after. This rule exception allows a class to have the desired form of a public region at the top and a private region at the bottom: A public inline member function can thus access a private data member even though that private data member is declared after the function.

PARTICIPATION ACTIVITY 7.4.3: Inline member functions.

Consider the following class definition.

```
class PickupTruck {
   public:
      void SetLength(double fullLength);
      void SetWidth (double fullWidth) {
         widthInches = fullWidth;
      }
   private:
      double lengthInches;
      double widthInches;
};

void PickupTruck::SetLength(double fullLength) {
   lengthInches = fullLength;
}
```

1) Inside the class definition, SetLength() is declared but not defined.

[✓ True] False

Correct

`void SetLength(double fullLength);` is just the declaration; the definition appears further below, outside the class definition.

2) Inside the class definition, SetWidth() is declared but not defined.

True [✓ False]

Correct

SetWidth()'s definition is inside the class definition, having not just the function name and parameter, but also the statements that define the function, in this case `widthInches = fullWidth;`

3) SetWidth() is an inline member function.

[✓ True] False

Correct

Because SetWidth() is defined inside the class definition, SetWidth() is an inline member function. In contrast, SetLength() is defined outside the class definition (sometimes called an out-of-line member function).

4) SetWidth()'s use of widthInches is an error because widthInches is declared after that use.

True

Correct

Normally variables must be declared before being used, but an exception to the rule exists within a class definition.

[✓ False]

5) If the programmer defines SetWidth() inline as above, then the programmer should probably define SetLength() as inline too.

[✓ True]

False

Correct

The program above is written to illustrate the different ways of defining member functions, but good style is to be consistent. Since both functions are very short, a consistent style would be to define both inline.

Feedback?

Inline member functions on one line

Normally, good style dictates putting a function's statements below the function's name and indenting. But, many programmers make an exception by putting very-short inline member function statements on the same line, for improved readability. This material may use that style at times. Example:

```
...
     void SetName(string restaurantName) { name = restaurantName; }
     void SetRating(int userRating) { rating = userRating; }
...
```

CHALLENGE ACTIVITY 7.4.1: Inline member functions.

Jump to level 1

Type the program's output.

```
Blackco
```

```
#include <iostream>
#include <string>
using namespace std;

class Book {
   public:
      void SetTitle(string bookTitle) { title = bookTitle; }
      void SetAuthor(string bookAuthor) { author = bookAuthor; }
      void Print() const {
         cout << title << ": " << author << endl;
      }

   private:
      string title;
      string author;
};

int main() {
   Book myBook;

   myBook.SetTitle("Blackcollar");
   myBook.SetAuthor("T. Zahn");

   myBook.Print();

   return 0;
}
```

1 2

Check Next **Done**. Click any level to practice more. Completion is preserv

✓ All member functions are defined inline. Normally, good style dictates putting a function's s name and indenting. But, many programmers make an exception by putting very-short inline m same line, for improved readability.

Yours
```
Blackcollar: T. Zahn
```

Expected
```
Blackcollar: T. Zahn
```

Feedback?