

Inline Functions

When a member function definition is short, you can give the function definition within the definition of the class. You simply replace the member function declaration with the member function definition; however, since the definition is within the class definition, you do not include the class name and scope resolution operator. For example, the class `Pair` defined below has inline function definitions for its two constructors and for the member function `get_first`:

```
class Pair
{
public:
    Pair( ) {}
    Pair(char first_value, char second_value)
        : first(first_value), second(second_value) {}
    char get_first()
    {
        return first;
    }
    ...
private:
    char first;
    char second;
};
```

Note that there is no semicolon needed after the closing brace in an inline function definition, though it is not incorrect to have a semicolon there.

Inline function definitions are treated differently by the compiler and so they usually run more efficiently, although they consume more storage. With an inline function, each function call in your program is replaced by a compiled version of the function definition, so calls to inline functions do not have the overhead of a normal function call.