

Objects in C++

See also: Chapters 1, 8

Class Declarations Definitions

In C++, the prototype for a class is called a *class definition*, even though it more closely resembles a function *declaration*. The class definition follows this form:

```
class SpreadsheetCell
{
    public:
        void setValue(double inValue); // method
        double getValue();

    protected:
        double mValue; // member
};
```

Access Levels

Remember that access levels define how *clients* and *subclasses* of your class can *use* methods and members. They do not apply to other objects of the same class. Also, remember access levels have nothing to do with subclass overriding privileges. In C++, a subclass can override a private method even though it can't call the original method.

Access Specification	Meaning	When to Use
public	Any code can call a public method or access a public member of an object.	Methods that you want clients to use. Accessor methods for data members.
protected	Methods of the class can call protected methods & use protected members. Methods of a <i>subclass</i> can as well.	"Helper" methods that you do not want clients to use. Most data members.
private	Only methods of the class can call a private method and access a private member. Methods in subclasses <i>cannot</i> access private methods or members.	Only if you want to restrict access from subclasses.

Class Implementation

```
// SpreadsheetCell.cpp
#include "SpreadsheetCell.h"

void SpreadsheetCell::setValue(double inValue)
{
    mValue = inValue;
}

double SpreadsheetCell::getValue()
{
    return (mValue);
}
```

Stack Based Object Creation and Usage

```
SpreadsheetCell myCell, anotherCell;
myCell.setValue(6);
anotherCell.setValue(myCell.getValue());

cout << "cell 1: " << myCell.getValue() << endl;
cout << "cell 2: " << anotherCell.getValue() << endl;
```

Heap Based Objects

```
SpreadsheetCell* myCellp = new SpreadsheetCell();

myCellp->setValue(3.7);
cout << "cell 1: " << myCellp->getValue() <<
    " " << myCellp->getString() << endl;
delete myCellp;
```