

A First Book of C++

From Here to There

Additional Class Capabilities

Assignment

- **Memberwise assignment:** Allows assignment of data member values of an object to their counterparts in another object of the class
 - Compiler builds this type of default assignment if there are no instructions to the contrary
- **Assignment operators:**
 - Declared in class declaration section
 - Defined in class implementation section
 - **Example:** `a = b;` in `main()` of Program 13.1

Assignment (continued)

Program 13.1: Assignment Example: Declaration

```
#include <iostream>
#include <iomanip>
using namespace std;
// class declaration
class Date
{
private:
    int month;
    int day;
    int year;
public:
    Date(int = 7, int = 4, int = 2006); //
    constructor prototype
    void showDate(); // member function to
    display a Date
};
```

Assignment (continued)

Program 13.1: Assignment Example: Implementation

```
// Implementation section
Date::Date(int mm, int dd, int yyyy)
{
    month = mm;
    day = dd;
    year = yyyy;
}
void Date::showDate()
{
    cout << setfill('0')
          << setw(2) << month << "/"
          << setw(2) << day << "/"
          << setw(2) << year % 100;
    return;
}
```

A First Book of C++: From Hello To There, Third Edition

4

Assignment (continued)

Program 13.1: Assignment Example: main()

```
int main()
{
    Date a(4,1,1999), b(12,18,2006); // declare two
                                   // objects
    cout << "\nThe date stored in a is originally ";
    a.showDate(); // display the original date
    a = b; // assign b's value to a
    cout << "\nAfter assignment the date stored in a is ";
    a.showDate(); // display a's value
    cout << endl;
    return 0;
}
```

A First Book of C++: From Hello To There, Third Edition

5

Assignment (continued)

- **Assignment operator declaration:**
 - **Format:** `void operator=(Date&);`
 - Declares simple assignment operator for Date class of Program 13.1
 - Add to public section of class declarations
 - **Keyword void:** Assignment returns no value
 - **operator=** indicates overloading of assignment operator with new version
 - **(className&):** Argument to operator is class reference

A First Book of C++: From Hello To There, Third Edition

6

Assignment (continued)

- **Assignment operator implementation format:**

```
void Date::operator=(Date& newdate)
{
    day = newdate.day; // assign the day
    month = newdate.month; // assign the month
    year = newdate.year; // assign the year
}
```

- Add to implementation section of Program 13.1
- **newdate**: a reference to the `Date` class
 - Reference parameters facilitate overloaded operators
- **day**, **month**, and **year** members of **newdate**:
Assigned to corresponding members of current object

Assignment (continued)

- **Program 13.2:** Program 13.1 + declaration and implementation of overloaded assignment operator (`operator=`)
- Allows for assignments such as:
`a.operator=(b);`
 - Calls overloaded assignment operator to assign `b`'s members to `a`
- `a.operator=(b)` can be replaced with `a = b;`

Assignment (continued)

- **Other issues affecting assignment operators**
 - Use constant reference parameter
 - **Format:** `void Date::operator=(const Date& newdate);`
 - Precludes inadvertent change to `newdate`
 - Assignment returns no value
 - Cannot be used in multiple assignments such as:
`a = b = c`
 - Reason: `a = b = c` equivalent to `a = (b + c)`
 - But `(b + c)` returns no value making assignment to `a` an error
