

Classes, Objects, Scope, and Constructors

2/14/05

CS250 Introduction to Computer Science II

1

The Person Class

```
class Person
{
public:
    int age;

    int returnAge();
    int returnBirthYear();
};

int main()
{
    Person person;
    person.age = 28;
    cout << "person is: " << person.returnAge();
    cout << "person was born in: "
        << person.returnBirthYear();
    return 0;
}
```

Annotations: "Data Member" points to `int age;`; "Member Function prototypes" points to `int returnAge();` and `int returnBirthYear();`

2/14/05

CS250 Introduction to Computer Science II

2

The Person Class

```
int Person::returnAge()
{
    return age;
}

int Person::returnBirthYear()
{
    return 2003 - age;
}
```

2/14/05

CS250 Introduction to Computer Science II

3

Private & Public

- Called member access specifiers
- Class data members and member functions can be either private or public
- Private data members and member functions can only be accessed within that class
- Public data members and member functions can be accessed from outside of that class

2/14/05

CS250 Introduction to Computer Science II

4

Example Using Private & Public

```
class Person
{
private:
    int age;
public:
    void setAge(int);
    int returnAge();
    int returnBirthYear();
};

int main()
{
    Person person;
    person.setAge(28);
    cout << "person is: " << person.returnAge() << endl;
    cout << "person was born in: "
        << person.returnBirthYear();
    return 0;
}
```

Annotations: "Because age is a private data member, we can't use `person.age = 28` here." and "Instead, we need to create a new function in the class to set the age." both point to `person.setAge(28);`

2/14/05

CS250 Introduction to Computer Science II

5

Continued

```
void Person::setAge(int newAge)
{
    age = newAge;
}
```

2/14/05

CS250 Introduction to Computer Science II

6

Fig. 6.3

```
class Time
{
public:
    Time();
    void setTime( int, int, int );
    void printUniversal();
    void printStandard();

private:
    int hour;    // 0 - 23 (24-hour clock format)
    int minute; // 0 - 59
    int second; // 0 - 59

}; // end class Time
```

2/14/05

CS250 Introduction to Computer Science II

7

Objects

- How do we create objects of the class time?
 - A regular object
 - An array of objects
 - A pointer to a Time object
 - A reference to a Time object

2/14/05

CS250 Introduction to Computer Science II

8

Constructor

- `Time()`;
- Special member function to initialize data members
- Does not have a return value
- The constructor is called whenever an object of that class is created

2/14/05

CS250 Introduction to Computer Science II

9

Examples

- Which of the following statements is invalid and why?
 - `Time &timeRef;`
 - `timeRef.printUniversal;`
 - `Time *pTime;`
`pTime->printStandard();`
 - `Time timeArray[5];`
`for(int i=0; i<5; i++)`
`cout << timeArray[i].printStandard();`
 - `Time t;`
`t.hour = 14;`

2/14/05

CS250 Introduction to Computer Science II

10

Object-Oriented Features

- Information hiding
 - Separate the implementation from the interface
 - Objects are concerned with the interface, for example what functions are available to manipulate the data
 - Objects are not concerned with the implementation. They do not care how the functions do what they do, as long as they do it correctly

2/14/05

CS250 Introduction to Computer Science II

11

Summary

- Today I introduced
 - Classes and objects
 - Constructors
 - Scope
 - Information hiding
- We have covered:
 - P. 411 - 420 in Chapter 6

2/14/05

CS250 Introduction to Computer Science II

12