Classes, Objects, Scope, and Constructors

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
```

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;`

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

Problem

- Create two classes as follows:
- Rectangle is a class that has two private members length and width. There are three member functions associated with the class Rectangle and they are printArea, printPerimeter, and setValues.
- Circle is a class that has one private member radius. There are three member functions associated with the class Circle and they are printArea, printCircumference, and setRadius.

Summary

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