Ch 13: Introduction to Classes

CS 2308 Fall 2011

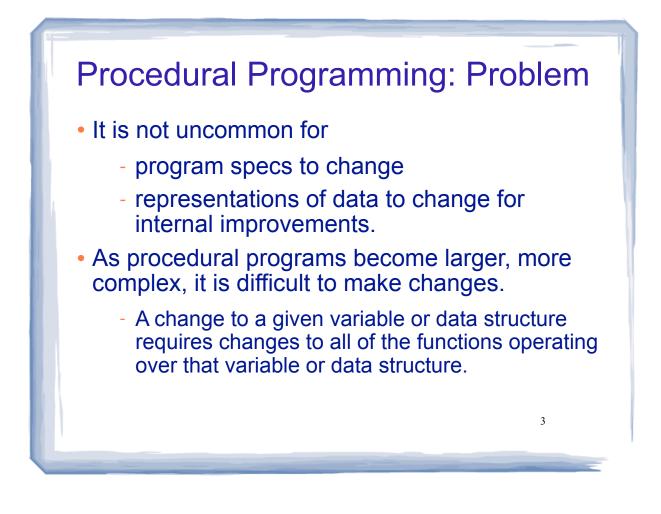
Jill Seaman

Lecture 11

Procedural Programming

- Data is stored in variables
 - Perhaps using arrays and structs.
- Program is a collection of functions that perform operations over the variables
 - Good example: book inventory program
- Usually variables are passed to the functions as arguments
- Focus is on organizing and implementing the functions.

1



Object Oriented Programming: Solution

An object contains

- data

- functions that operate over the data
- Code outside the object can access the data only via the member functions.
- If the representation of the data in the object needs to change:
 - The member functions must be redefined to handle the changes.
 - The code outside the object does not need to change, it accesses the object in the same way.

Object Oriented Programming: Concepts

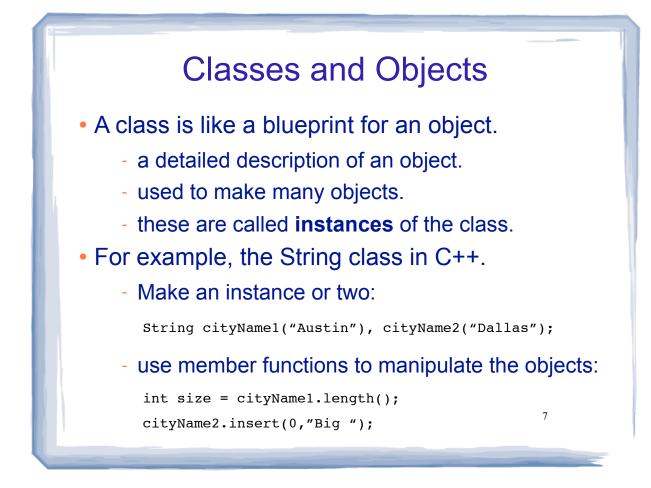
- Encapsulation: combining data and code into a single object.
- **Data hiding** (or **Information hiding**) is the ability to hide the details of data representation from the code outside of the object.
- Interface: the mechanism that code outside the object uses to interact with the object.
 - The member functions
 - Specifically outside code needs to know only the function prototypes.

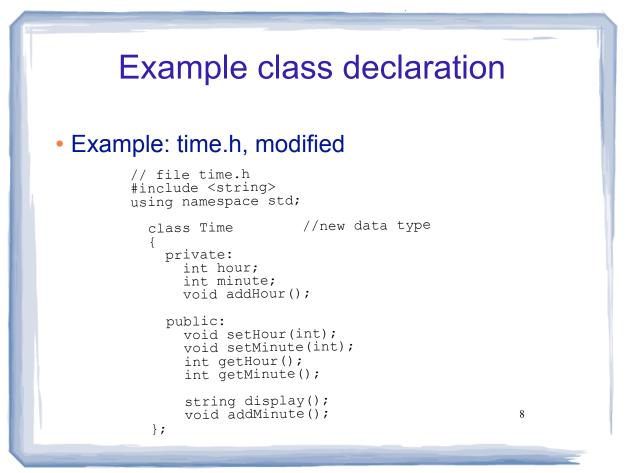
5

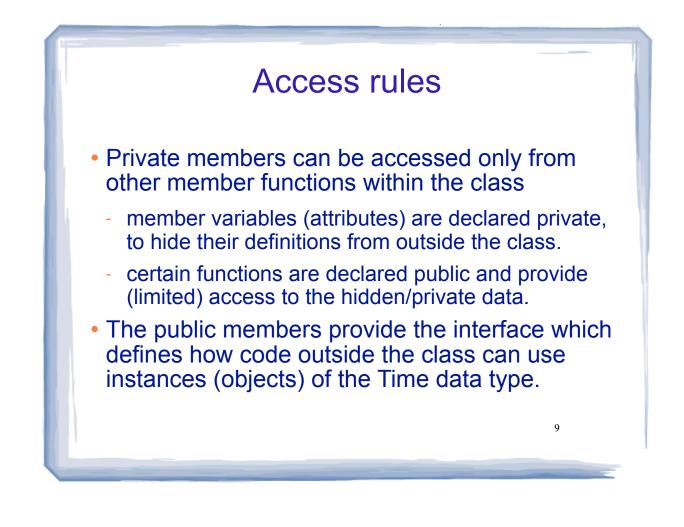
6

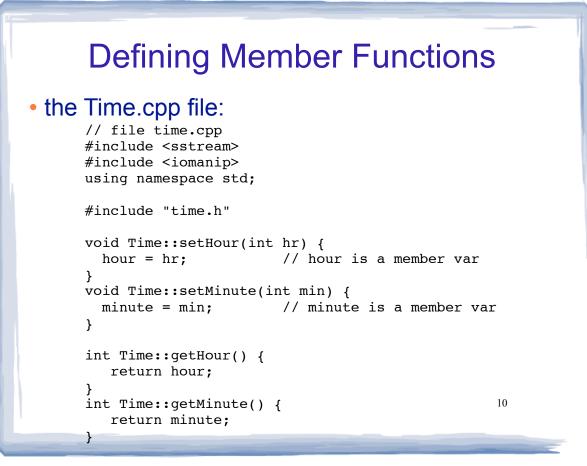
Object Oriented Programming: Real World Example

- In order to drive a car, you need to understand only its interface:
 - ignition switch
 - gas pedal, brake pedal
 - steering wheel
 - gear shifter
- You don't need to understand how the steering works internally.
- You can operate any car with the same interface.









Defining member functions

- Member function definitions occur OUTSIDE of the class definition, usually in a separate file.
- The name of each function is preceded by the class name and :: operator
 - Time::setHour(int hr)
- Accessor functions (a.k.a. "getters")
 - returns a value from the object (without changing it)

11

12

- Mutator functions (a.k.a. "setters")
 - Changes values of member variables.

Defining an instance of the class

ClassName objectname:

Time t1;

- This defines t1 to contain an object of type Time (the values of hour and minute are not set).
- Access public members of class with dot notation:

```
t1.setHour(3);
t1.setMinute(41);
t1.addMinute();
```

• Use dot notation OUTSIDE class only.

Setters and getters: what's the point?

- Why have setters and getters that just do assignment and return values?
- Why not just make the member variables public?
- Setter functions can validate the incoming data.
 - setMinute can make sure minutes are between 0 and 59 (if not, it can throw an *exception*).
- Getter functions could act as a gatekeeper to the data (validate "user") or provide type conversion.

13

