Lecture 8 – Classes

CST238 – Intro to Data Structures
YoungJoon Byun
ITCD

1

Lecture Objectives

- After completion of this lecture, you will be able to

2

Chapter 4: More about OOP and ADTs Classes

- 4.1 Procedural vs. Object-Oriented Programming
- **4.2 Classes**
- 4.3 Example: A First Version of a User-Defined Time Class
- 4.4 Class Constructors
- 4.5 Other Class Operators

3

C++ Classes – Introduction

- C++ classes model objects which have
  - attributes represented as data members
  - operations represented as functions (or methods)
- A class is a heart of object oriented programming.

4

Example of User-Defined Class (1 of 3)

```
1.  class DayOfYear {
2.  public:
3.      void readDate( );
4.      void printDate( );
5.  private:
6.      string month;
7.      int day;
8.      int year;
9.  };
```

5

Example of User-Defined Class (2 of 3)

```
10. void DayOfYear::readDate( )
11. {
12.         cout << “Enter the month: ”;
13.         cin >> month;
14.         cout << “Enter the day: ”;
15.         cin >> day;
16.         cout << “Enter the year: ”;
17.         cin >> year;
18. }
19. void DayOfYear::printDate( )
20. {
21.     cout << month << “/”  << day << “/” << year << endl;
22. }
```

6

Example of User-Defined Class (3 of 3)

```
23.  int main( )
24.  {
25.      DayOfYear birthday;
26.      cout << "When is your birthday? " << endl;
27.      birthday.readDate( );
28.      cout << "Your birthday is ";
29.      birthday.printDate( );
30.      return 0;
31.  }
```

7

Class vs. Object

- A class is a special kind of programmer-defined type.
- An object is an instance of the class.

8

Three Items for Class Definition and Usage

(1) Class declaration (interface)

(2) Class implementation

(3) Class driver

9

Class Declaration Syntax

- class Class_Name
  {
    public:
      Declarations of public members;
    private:
      Declarations of private members;
  }; // Don't forget the ";"

10

Designing a Class

- Data members are normally placed in the private section of a class
- Function members are usually in the public section
- Typically the public section is followed by the private section
  - However, it is not required by a compiler.

11

Definition of a Function Member

- Returned_Type Class_Name::Function_Name(Parameter_List)
  {
    Function Body Statements
  }
- Example

```
void DayOfYear::printDate( )
{
   cout << month << "/"  << day << "/"
        << year << endl;
}
```

12

Invoking a Function Member

- A method can be invoked using the name of an object variable and the name of function method.
  - Invoking a method is equivalent to executing the method function body.

```
objectVariable.method(parameters);
e.g., birthday.readDate();
```

13

Exercise – Is this valid?

```
1. int main( )
2. {
3.     DayOfYear today;
4.     today.readDate();
5.     today.day = 28;
6.     today.printDate();
7.     return 0;
8. }
```

14

`private` Modifier

- The modifier **`private`** means that a private member cannot be accessed from the outside of the class
- Example

```
DayOfYear birthday;
birthday.month = 12;    // Invalid.
```

- Typically, **all** data member are **`private`**

15

`public` Modifier

- The modifier **`public`** means that there are no restrictions to access the member from outside of a class.

```
DayOfYear birthday;
birthday.readDate();
```

  - Most function methods are **`public`**

16

Summary

- Review classes in C++ (chap. 4.2)
- Next Lecture
  - Example class: **`Time`** (chap. 4.3)
  - Class constructors (chap. 4.4)
  - Other class operations (chap 4.5)

17

References

- Larry Nyhoff, *ADTs, Data Structures, and Problem Solving with C++*, 2nd Edition, Prentice-Hall, 2005
- Walter Savitch, *Problem Solving with C++*, 6th Edition, Addison-Wesley, 2006
- Dr. Meng Su's Lecture Notes
  http://cs.bd.psu.edu/~mus11/122Fa06/cse122Fa06.htm

18