Lecture 6

Christopher Godley

CSCI 2270 Data Structures

June 12th, 2018

Classes

- Just like structs, a class is a template for a complex data type.
- Instances of a class are called an object
- Wait, what’s a struct?...

```
struct Date{
        int month;
        int day;
        int year;
};
```

Classes

- Struct:
  - How do we maintain data integrity?
  - If we’re setting the value for month, what prevents us from entering a 13?
- A Class is like a Struct but:
  - Structs only contain data
  - Classes maintain member variables along with functions

Classes

- Features of Classes:
  - Complex type
  - Functions included in class definition, called **methods**, control access to member variables
  - Variables and methods can be public or private
  - Constructor called to create instance of class
  - Destructor to destroy class (free memory)
  - Many more exist, but these are the foundational features

Classes: Methods

- Methods:
    - Public
        - Can be accessed outside of the class
    - Private
        - Can only be accessed by class methods.
        - Ex) A private variable can be viewed in the main function directly

Classes: Example

- Create a class called date with three private members: month, day, year. All integers, the constructor takes 3 arguments for month, day, year. Public method to print month, day, year.

Classes: Example

```
class Date {
    private:
        int month;
        int day;
        int year;
    public:
        Date(int m, int d, int y) {
            month = m;
            day = d;
            year = y;
        }
        void printDate() -> next page
```

Classes: Example

```
class Date {
        private:
                …
        public:
                …
                void printDate() {
                        cout << month << “_” << day << “_” << year;
                }
};
```

Classes: Example

```
int main() {
        Date d(1, 1, 2017);

        // What's the problem here?
        cout << d.month;
        d.month  = 12;
```

They're private!

Classes: Updating Private Variables

- How can we update a private variable?
  - Create a public method that controls access
- Add the following method to the Date class:

```
void setMonth(int m) {
    if m > 0 and m < 13 {
        month = m;
    } else {
        cout << “out of range” << endl;
    }
}
```

Classes: Example

- Back in main:

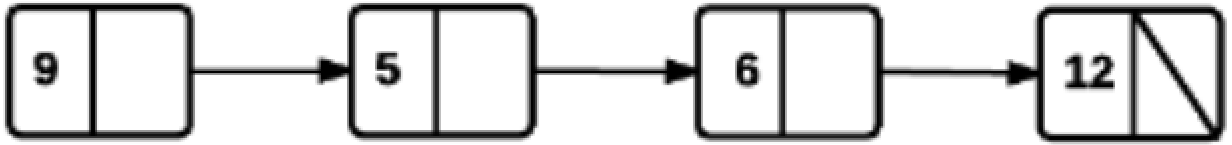
```
int main() {
    …
    d.setMonth(13); // should print the failure
    d.setMonth(5); // month is updated
```

- Note:
  - setMonth() is a member of Date. We need the instance of Date, and then call setMonth on that instance.
  - Similar to notation of accessing a struct member using instance of struct

Linked Lists

Linked Lists

- Two types:
  - Singly Linked List
    - Each node stores the pointer to the next



Linked Lists

- Doubly Linked
  - Each node stores the pointer to the previous as well as the next node