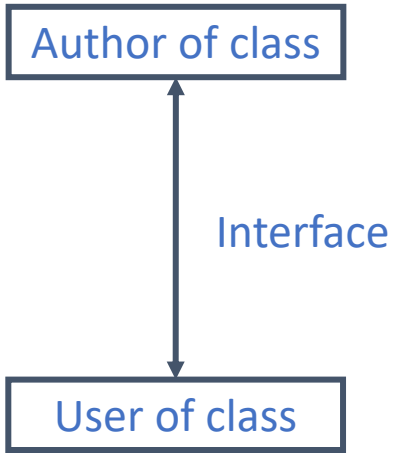
Programming Languages and Tools: Programming with C++ CS:3210:0003

Lecture/Lab #13

Encapsulation via Classes

- Interface: public member functions to interact with objects
  - Must be stable (minimal changes)
- Implementation: member variables + bodies of member functions
  - Hidden, easier to change without breaking code
- Data hiding separates interface and implementation
- Confusingly, encapsulation refers both to:
  1. Data hiding
  2. Bundling of data and functions together



Encapsulation in C++

- Data members are private
- Member functions are public, allowing access to data members
- Benefits:
  1. User of class can avoid implementation details
     Ex: std::string
  2. Allows developer to maintain class invariants
     Ex: in a class modelling a tennis game score, a desirable invariant is that both players can never win the same point or game
  3. Change implementation details without breaking user code

Constructors

- A class type with private member variables
  - Is not an aggregate type
  - Cannot be aggregate initialized
    ```
    ClassName Obj {x, y}; //not allowed
    ```
- Constructor: special member function automatically called after a non-aggregate type object is created
- When object is defined:
  - Compiler looks for a matching constructor
  - If found, memory for object is allocated and constructor called
  - Otherwise, compilation error

Constructors

- Same name as class
- No return type
- Usually public
- Initializes member variables, among other things
  - Using member initializer/initialization list
  - Syntax:

```
constrName(args)
  : privMem1 { initValue1 }, …, privMemn { initValuen }
  { constrBody }
```

- Members initialized in the order in which they are defined in the class

Member Initialization

- Priority order for class member initialization:
  1. If a member is listed in the member initializer list, that initialization value is used
  2. Otherwise, if the member has a default initializer, that initialization value is used
  3. Otherwise, the member is default initialized