



Visibility

Bachelor of Science - École polytechnique

gael.thomas@inria.fr

Key concepts

- Two keywords for the visibility
 - `private`: the members are hidden from outside the class
 - `public`: the members are visible from outside the class
- Two keywords to declare a class
 - `struct`: default visibility is `public`
 - `class`: default visibility is `private`

External interface

- The user of an object is interested by what an object does
 - Not by how the object is implemented
- A class can thus define an external interface for an object
 - The methods that a user can call from the outside
 - The fields that a user can directly access from the outside
- And an object can have an internal interface
 - Methods hidden from the outside
 - Fields hidden from the outside

public versus private

- Two new keywords to define the interfaces
 - **private**: the private interface (internal interface)
 - **public**: the public interface (external interface)

```
struct monster_t {  
private:  
    std::string name;  
    int health;  
  
    void internal_method();  
  
public:  
    monster_t(std::string name, int health);  
  
    void print();  
};
```

} private interface:
 hidden except from
 the methods of
 monster_t

} public interface:
 outside the
 scope, we can only
 use these methods

struct versus class

- C++ uses two keywords to define a class
 - `struct`: default visibility is `public`
 - `class`: default visibility is `private`

```
class monster_t {  
    std::string name;  
    int health;  
  
    void internal_method();  
  
public:  
    monster_t(std::string name, int health);  
  
    void print();  
};
```

With the `class` keyword, these members have the `private` visibility

struct versus **class**

- Which one is best?
 - Mostly a question of habit
 - **class** is probably more common than **struct**

Notes: in the slides, we mostly use **struct** in order to avoid adding superfluous public for the examples, not because **struct** is better than **class**

The friend keyword

- A friend class or method can access private members

```
class all_private_t {
    int x;
    friend void f(all_private_t* p);
};

void f(all_private_t* p) {
    p->x = 42;
}
```

Key concepts

- Two keywords for the visibility
 - `private`: the members are hidden from outside the class
 - `public`: the members are visible from outside the class

- Two keywords to declare a class
 - `struct`: default visibility is `public`
 - `class`: default visibility is `private`