Topic 9: Object Oriented Programming

VE482 Introduction to Operating System

Kexuan Huang
Weili Shi

October 21, 2021

What are virtual functions and abstract classes?

Virtual Functions

Definition

- A virtual function is a member function that you expect to be redefined in derived classes.
- When you refer to a derived class object using a pointer or a reference to the base class, you can call a virtual function for that object and execute the derived class's version of the function.

Code

```
virtual void foo();
```

Virtual Functions

Pure Virtual Functions

- A pure virtual function is a virtual function that is required to be implemented by a derived class if the derived class is not abstract.
- A virtual function can be declared by using the pure specifier (= 0) syntax.

Code

```
virtual void foo() = 0;
```

Abstract Classes

Definition

- Abstract classes act as expressions of general concepts from which more specific classes can be derived. You can't create an object of an abstract class type. However, you can use pointers and references to abstract class types.
- You can create an abstract class by declaring at least one pure virtual member function.
- Classes derived from the abstract class must implement the pure virtual function or they, too, are abstract classes.

Abstract Classes

Code

```
class Foo {
public:
    virtual void bar() = 0;
    ...
};
```

What is the friend keyword and why should it be used as little as possible?

The friend Keyword

Definition

- The friend declaration appears in a class body and grants a function or another class access to private and protected members of the class where the friend declaration appears.

Code

```
class Foo {
  friend class Bar;
  friend void func();
  private:
    double _var;
};
```

The friend Keyword

Merits of friend Keyword

- Provide data access feasibility and flexibility for functions and classes

Demerits of friend Keyword

- Friend function have access to private members of a class from outside the class which violates the law of data hiding
- Lead to spaghetti-code situation if numerous friends muddy the boundary between classes
- Maximum size of memory will be occupied by objects according to size of friend members

Why are the override and final keywords added in C++11?

The override Keywords

Definition

- In a member function declaration or definition, override specifier ensures that the function is virtual and is overriding a virtual function from a base class.

Code

```
class A {
  virtual void foo();
  virtual ~A();
};
class B : A {
  void foo() override;
  ~B() override;
};
```

The final Keywords

Definition

- When used in a virtual function declaration or definition, final specifier ensures that the function is virtual and specifies that it may not be overridden by derived classes.

Code

```
class A {
  virtual void foo();
};
class B : A {
  void foo() final;
};
class C : B {
  void foo() override; // Error
};
```