Inheritance and Polymorphism

Ho Dac Hung

Object-Oriented Programming

- Object-oriented programming is based on a paradigm in which objects are used to model a specification. Objects are created from classes, which provide encapsulation. Inheritance extends a class and provides a mean of polymorphism.

Extending a Class

- Making one class an extension of another involves inheritance. Inheritance allows a class to define a specialized type of an already existing class.
- Classed that are derived from existing classes demonstrate an is-a relationship. A class “is a” type of another class.

Implementing a Subclass

- A class that inherits another class includes the keyword extends in the class declaration and takes the form:

```
public class <name> extends <classname>{
    <class definition>
}
```

Implementing a Subclass

- Designing a subclass requires selecting the superclass, or base class and then defining any additional variable and method members for the subclass.
- In many cases, existing methods in the base class will also be overridden by new definition in the subclass, also called the derived class.

Implementing a Subclass

- In a subclass, the keyword super is used to access methods of the base class.
- Members that are declared private are not accessible to derived classes. Therefore, accessor methods are used to get inherited member variable values.

Polymorphism

- Polymorphism is an OOP property in which objects have the ability to assume different types. In OOP, polymorphism is based on inheritance.
- Because a subclass is derived from a superclass, a superclass object can reference an object of the subclass.

Polymorphism

- Polymorphism is further demonstrated when the referenced object determines which method to execute. This is possible when a subclass overrides a supperclass method.

Abstract Classes

- An abstract class model an abstract concept. Anstract classes cannot be instantiated because they should not represent objects. They instead describe the more general details and actions of a type of object.

Abstract Classes

- Abstract classes are declared with the keyword abstract in the class declaration. They are intended to be inherited. An abstract class can also contain an abstract method. An abstract method is declared with the keyword abstract and contains a method declaration, but no body. The abstract methods must be implemented in its subclasses.

Interfaces

- An interface is a class with method declarations that have no implementations. An interface cannot be inherited. It may only be implemented in a class.
- An interface can add behavior to a class, but it does not provide a hierarchy for the class.

```
<access level> interface <name> {
    <return type> <method name> (<params>);
    …additional methods
}
```