Chapter 12
Inheritance

CISC 5004 - Programming C++
Sam Kamens
Fall, 2022

Derived Classes

- Concept
- Inheritance
  - A derived class (subclass) inherits the properties of its base class (superclass)
  - An object of a derived class type has access to all public members of the derived class and its base class
  - Example code: Class definition, Usage

Inheritance Scenarios

- Variations
  - Derived class as a base class for another class
  - Base class from which multiple classes are derived
  - Class derived from multiple base classes
- Example: Business and Restaurant

Access by members of derived classes

- Members of a derived class can only access public members of the base class
  - Results in a compile error
- Protected Member Access
  - Allows access to members for derived classes (but no-one else)
  - Example

public/private/protected in Class definitions

- `class Derived : `**`public`**` Base { .. }`
  - `public` members of `Base` are `public` in `Derived`
  - `protected` members of `Base` are `protected` in `Derived`
- `class Derived : `**`protected`**` Base {..}`
  - `public` and `protected` members of `Base` are `protected` in `Derived`
- `class Derived : `**`private`**` Base { .. }`
  - `public` and `protected` members of `Base` are `private` in `Derived`
  - This is the default
- Most classes we use will use `public`

Overriding

- Derived class defined a class with the same signature as a base class’s function
    - Example
- Note: overriding != overloading
- Calling base class function from derived class

Polymorphism

- Refers to determining program behavior based on data types
    - Compile-time
    - Runtime
        - Compiler can not determine which function to call at compile time\
    - Example
        - Note the use of derived/base class pointer conversion

Virtual Functions

- Base class function name must be declared `virtual` to enable runtime polymorphism
- Derived class uses `override` to indicate that it is overriding a virtual function
- Example
- Implementation Note
  - Compiler creates a virtual table to look up which function to call at runtime
  - This makes runtime polymorphism slower than compile-time polymorphism

Pure virtual functions

- Base class does not provide a definition for a virtual function, but every derived class is required to provide one
- A class that has at least one pure virtual function is called an abstract base class
- Example

Abstract Classes

- Features of Object Oriented Programming
  - Classes
  - Inheritance
  - Abstract classes
    - As opposed to Concrete class
- Standard example: shape classes

Is-a vs. Has-a

- Is-a: inheritance
- Has-a: composition

UML

- Unified Modeling Language
  - Diagrams to visualize structure and behavior of programs
  - Class diagram
  - Inheritance

C++ Examples

- Employees and Overrides