CST209

Object-oriented Programming C++

(Week 7)

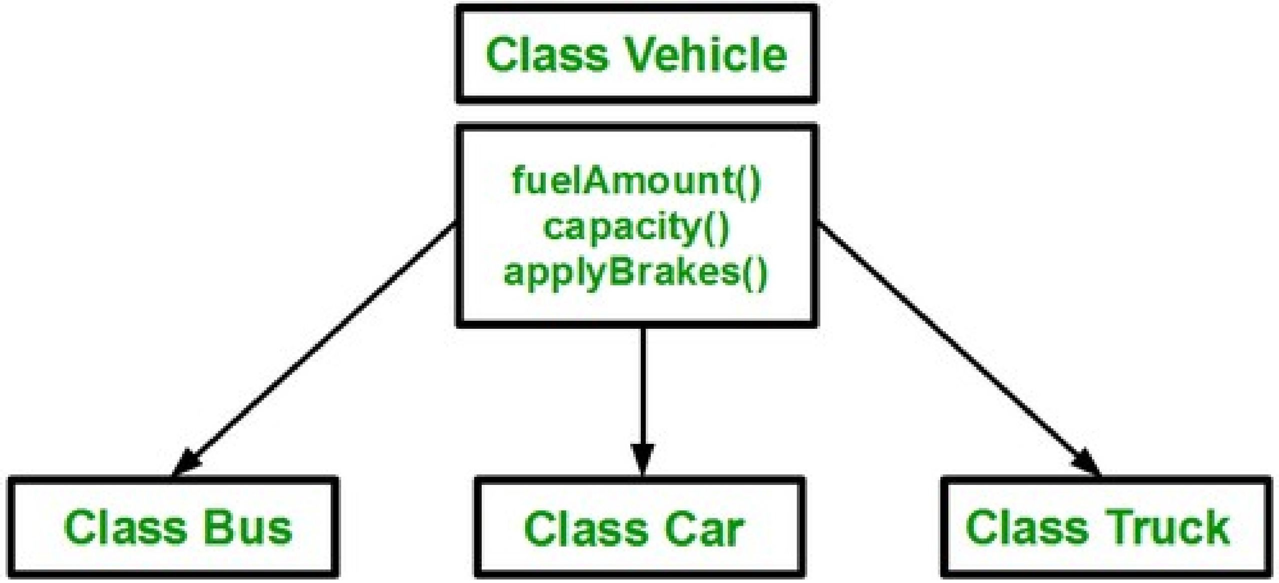
Dr. Teo Bee Guan

Content

- Inheritance

Inheritance

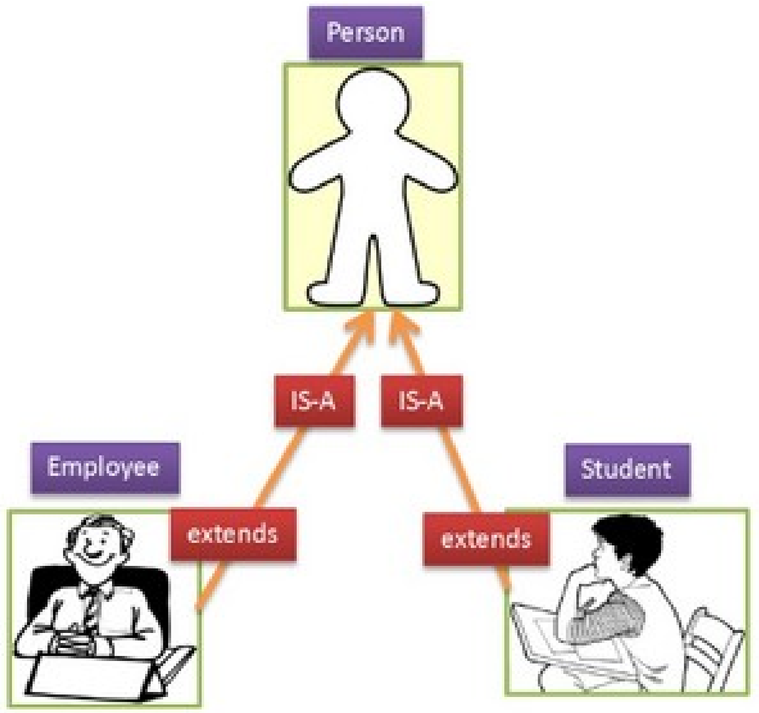
- Inheritance allows a new class to be based on an existing class.
- The new class inherits all the member variables and functions (except the constructors) of the class it is based on.



Inheritance

- When one object is a specialized version of another object, there is an "is a" relationship between them.

- When an "is a" relationship exists between classes, it means that the specialized class has all of the characteristics of the general class, plus additional characteristics that make it special.



Inheritance

- Inheritance involves a **base** class and a **derived** class.
- The base class is the general class and the derived class is the specialized class.
- The derived class is based on, or derived from, the base class.
- You can think of the base class as the parent and the derived class as the child.

Inheritance

- The derived class inherits the member variables and member functions of the base class without any of them being rewritten.

- Furthermore, new member variables and functions may be added to the derived class to make it more specialized than the base class.

Practice: Example 1, 2

In-class Exercise 1

Create a class named `Shape` with only a private member variable, name. You should provide a *getter* and *setter* member function in the class. Create another class named `Rectangle` that inherits/extends from the class `Shape`. The `Rectangle` class should have private member variables, `width` and `height`. Create getter and setter for each of the member variable and another member function to calculate the area of rectangle. Test your class in the `Main` function.

Protected Members

- Access modifier that permit members can be accessed in inherited classes.
- Protected members are not as private as private members, which are accessible only to members of the class in which they are declared, but they are not as public as public members, which are accessible in any function.

Practice: Example 3

In-class Exercise 2

Modify your `Shape` class by changing the access modifier for the member variable, `name`, from private to protected.

Create a new member function, `printRectangleDetails()` in the `Rectangle` class to print the details of an rectangle object (name, width, height and area).

You have to directly access the protected member variable, `name`, in your `printRectangleDetails()` function.

Redefining base class functions

- Inheritance is commonly used to extend a class or give it additional capabilities. Sometimes it may be helpful to overload a base class function with a function of the same name in the derived class.

- A subclass may have a method with the same signature as a superclass method.

- In such a case, the subclass method overrides the superclass method.

Practice: Example 4

In-class Exercise 3

Add a `printShapeDetails()` member function in your `Shape` class. This function shall only print the name of the shape.

In your `Rectangle` class, modify the `printShapeDetails()` function to print out the name of the shape along with the width, height and area.

See you next class