CSE-2102
Object Oriented Programming

Dr. Muhammad Ibrahim
Assistant Professor
Dept. of Computer Science and Engineering
University of Dhaka

Example of Inheritance

- Vehicle, car, ship, airplane, sedan car, bicycle, …
- Person, student, teacher, officer, …
- Person, customer, seller, auditor, …

Draw class diagrams for the above cases.

Code examples: from textbook’s Inheritance chapter

Two Questions

Question 1: What happens if both a superclass and a subclass define a function having the same name but different number/type of parameters?

Question 2: What happens if both a superclass and a subclass define a function having exactly the same prototype?

Two Questions

Question 1: What happens if both a superclass and a subclass define a function having the same name but different number/type of parameters?

Question 2: What happens if both a superclass and a subclass define a function having exactly the same prototype?

Answer 1: Overloading. Both methods are available in subclass.

Answer 2: Overriding. The redefinition hides the definition of superclass. However, using “super”, the superclass version is still available in subclass.

Referencing Subclass Object with Superclass Variable

- A superclass variable can refer to a subclass object (sometime called "upcasting").
    - May be a direct subclass or indirect.
- However, only the members of subclass that are inherited from superclass can be accessed.
    - Natural because superclass variable has no knowledge of the members that are added to subclasses.
    - Note: overridden method call will invoke subclass implementation, not superclass.
- This mechanism has some practical applications regarding Dynamic Method Dispatch (later in this lecture).

Typecast with Objects

```
class Animal {
void eat () {...}
}
class Dog extends animal {
void bark() { …}
}

Animal anml;
Dog dg = new Dog();
anml = dg; //we could also write
           //anml = (Animal)dg;
```

But now we cannot write: `anml.bark()` because bark() is not known in Animal class.

To call bark(), we can write:

`((Dog)anml).bark()`

Order of Execution of Constructors

- Constructor of a superclass is executed before the constructor of a subclass
  - Natural because constructor is used for initializing variables, so initialization of common variables should precede initialization of specific variables
- If “super” is not used (as the first statement in) a subclass’s constructor, a default or parameterless constructor for superclass is automatically executed.

Two points:

1. Although private members are not directly accessible outside a class, they can be accessed through public members, if any.
   a. In this sense, they are “inherited” in the subclass but can’t be directly accessed.
2. In C++, there is no notion of “super”.

Using Keyword “super”

Two uses:

1. Using “super” to call superclass constructors.
    a. To minimize coding.
2. Somewhat like “this” reference: always refers to superclass variable.

In case of multi-level inheritance, “super” refers to the members of an immediate superclass.

End of Lecture 9.

Reading material: Chapter 8 of the textbook.