

Object-Oriented Programming Concepts

Session: 5

Multiple Inheritance and Interfaces

Objectives

- ◆ Describe Multiple Inheritance
- ◆ List the problems associated with Multiple Inheritance
- ◆ Describe Interface
- ◆ Explain Multiple Inheritance using Interfaces
- ◆ Explain constructor execution in Multiple Inheritance

Introduction

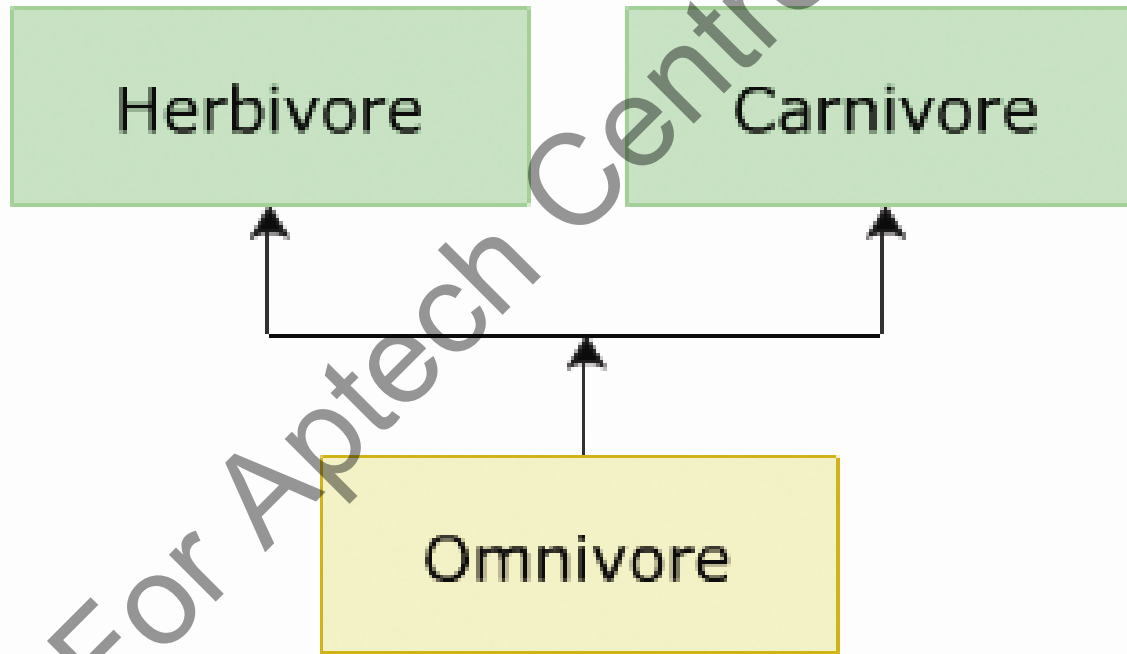
Emphasizes the correct way of expressing the relationships between classes

Explains creation of a class by combination of other classes

Describes the use of interfaces to achieve multiple inheritance in programming languages

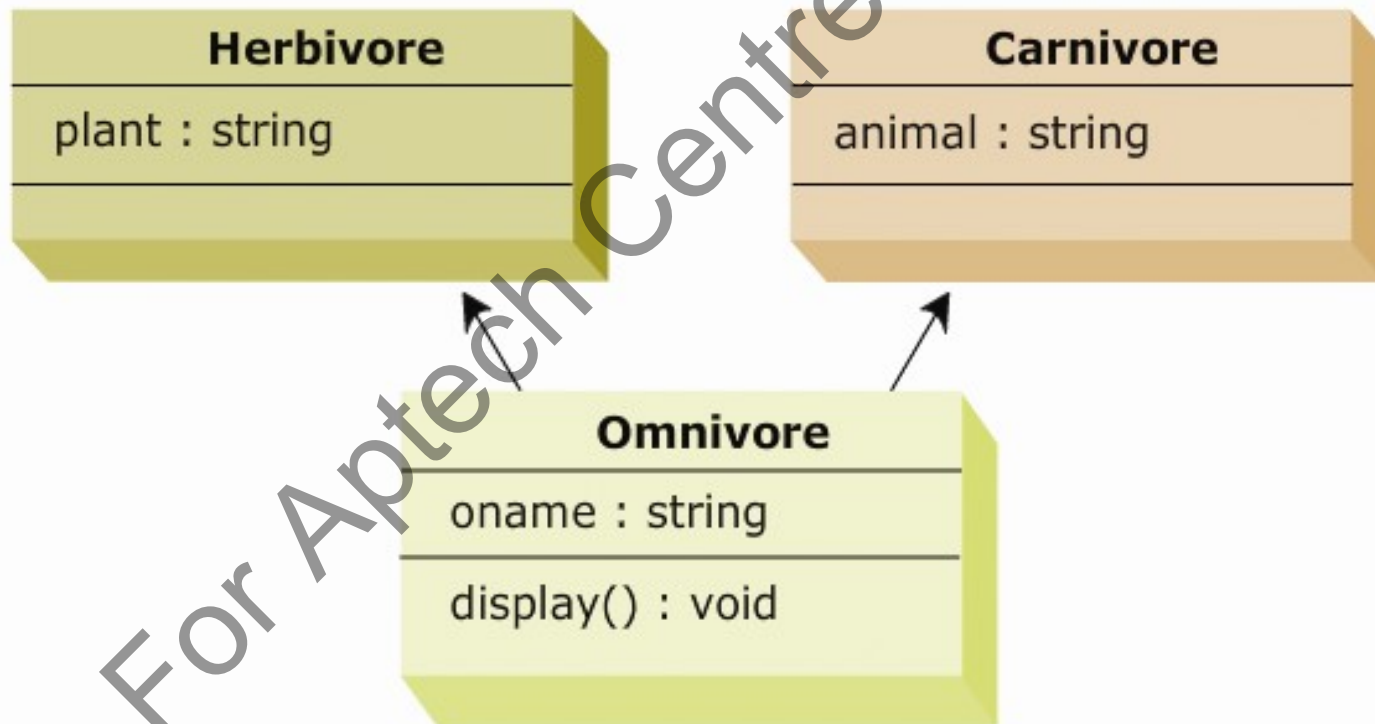
Multiple Inheritance 1-2

- ◆ It is a technique in which one class inherits its properties from more than one classes.
- ◆ The figure shows an example of multiple inheritance.



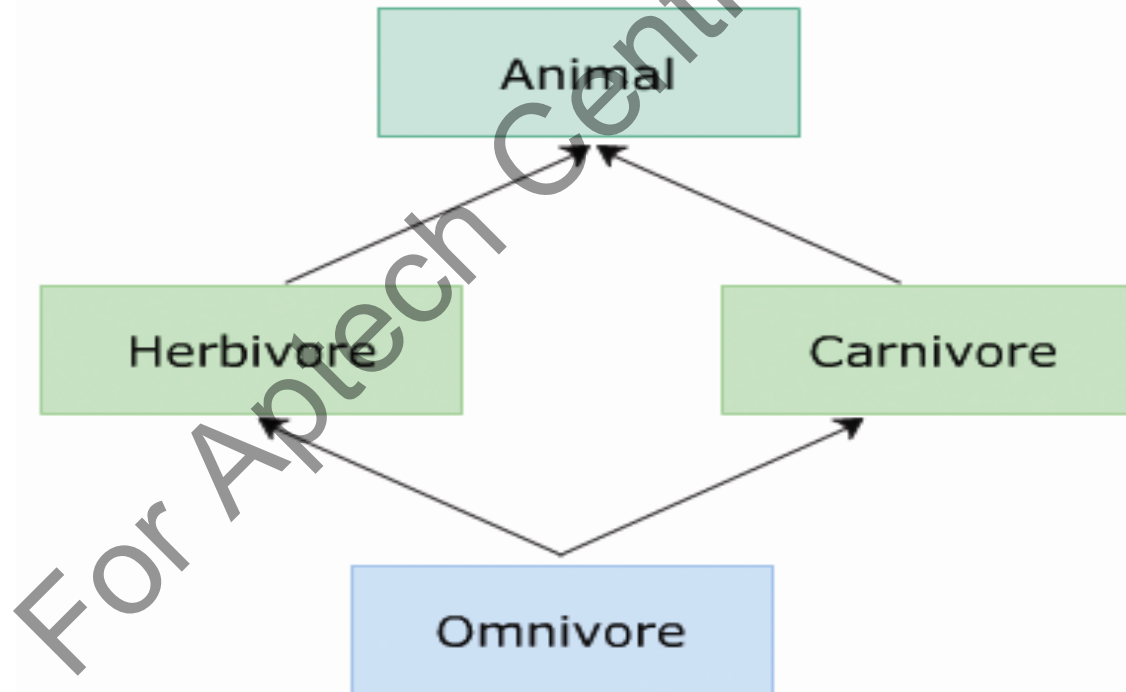
Multiple Inheritance 2-2

- ◆ The figure shows the class diagram of the inheritance hierarchy created by the three classes namely **Herbivore**, **Carnivore**, and **Omnivore**.



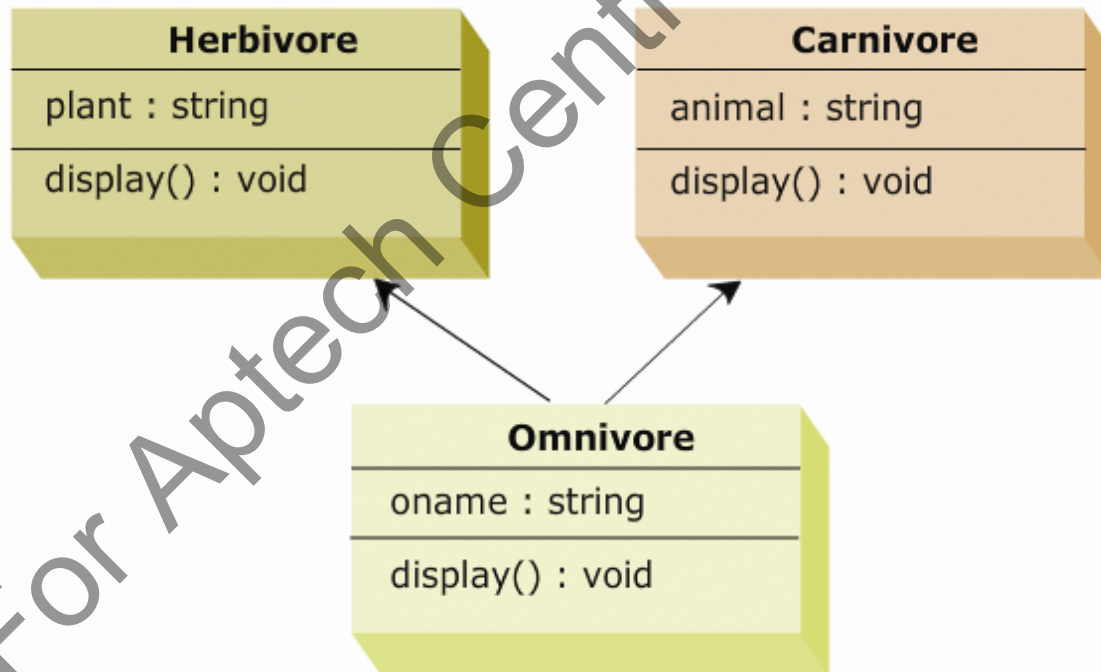
Problems with Multiple Inheritance 1-2

- ◆ **The Diamond Problem** - a class inherits from two parent classes each of which in turn inherit from a common super class.
- ◆ The figure depicts the diamond problem.



Problems with Multiple Inheritance 2-2

- ◆ **Ambiguity of Name** - a base classes have a method with the same name and signature.
- ◆ The figure depicts the name ambiguity problem.



Multiple Inheritance with Interfaces 1-3

- ◆ An interface only declares the signature of the methods it wishes to expose.

Syntax

```
<visibility-modifier> interface  
<interface-name>  
{  
// abstract method declarations  
}
```


Multiple Inheritance with Interfaces 2-3

- ◆ The following table shows features that differentiate a class and an interface.

Class	Interface
A class can be instantiated	An interface cannot be instantiated
A class can have concreted methods, that is, methods with a body	An interface cannot have concrete methods
A child class may not override all methods of a parent class	A class implementing an Interface must implement all methods of an interface
Multiple classes cannot be inherited by a child class to simulate multiple inheritance	Multiple interfaces can be implemented by a class to simulate multiple inheritance
An 'is-a' relationship is required to implement inheritance between classes	An 'is-a' relationship is not mandatory for implementing an interface
Methods of an interface are implicitly public	Methods of a class can have any access modifier

Multiple Inheritance with Interfaces 3-3

Some advantages of interface are as follows:

- ◆ Allows simulating multiple inheritance of classes
- ◆ Helps to avoid ambiguity between the methods of the different classes as seen in C++
- ◆ Allows combining features of two or more interfaces such that a class needs to only implement the combined result
- ◆ Helps to hide an inherited member name from any code outside the derived class

Multiple Inheritance and Constructors

Parent Class



No-argument constructor: Invoked in the sequence in which the child class has inherited the parent classes

Parameterized constructor: User can handle by invoking the parent class constructor from the constructor of the child class and pass arguments to the constructor

Summary 1-2

- ◆ Multiple Inheritance is a technique in which one class inherits its properties from more than one classes.
- ◆ Diamond Problem is a situation that arises when a class inherits from two parent classes each of which in turn inherit from a common super class.
- ◆ An interface is a type similar to a class that consists of only method declarations without any implementation.
- ◆ A class can implement more than one interface to simulate multiple inheritance.

Summary 2-2

- ◆ Invocation of constructors in multiple inheritance depends on the sequence in which the child class invokes them in its own constructor.
- ◆ Ambiguity of name is a condition in multiple inheritance wherein the base classes have a method with the same name and signature.