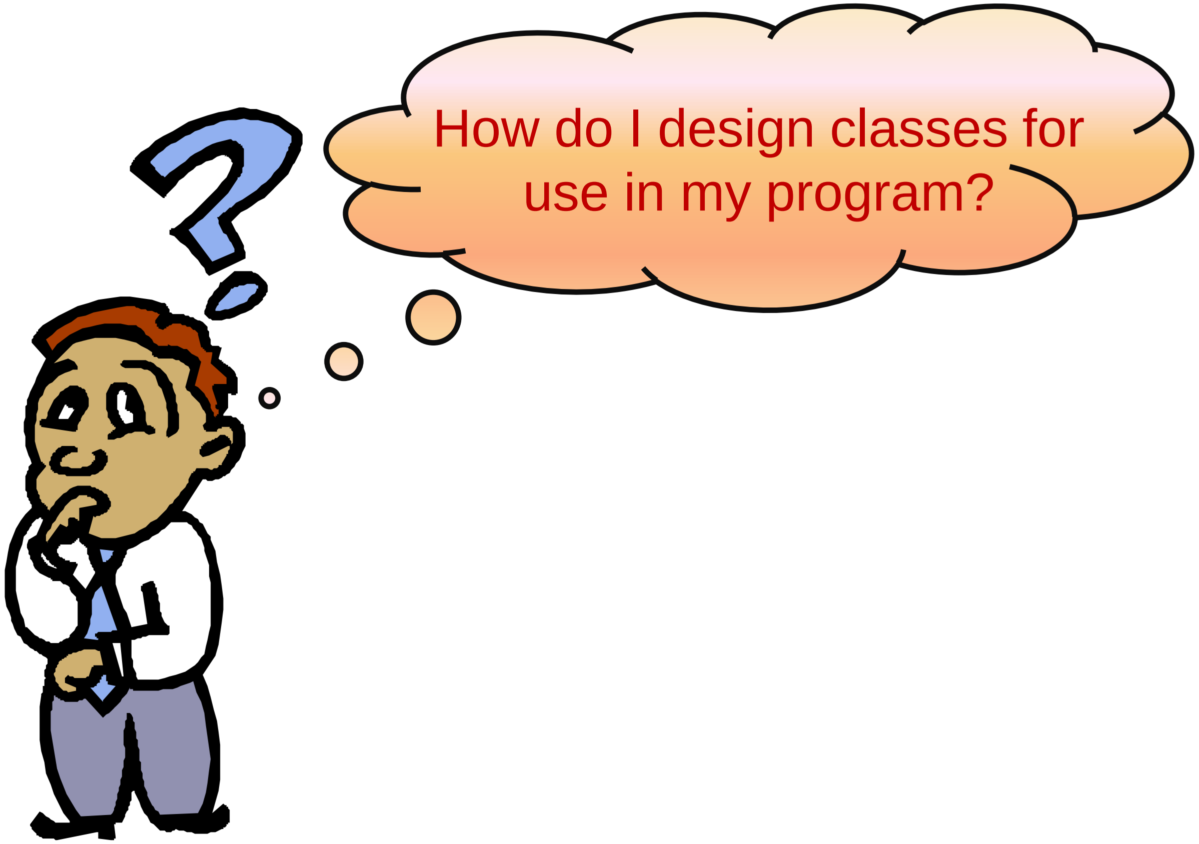
Programming in Java

Objectives

- In this session, you will learn to:
  - Design classes
  - Identify relationships between classes
  - Define information hiding
  - Define encapsulation
  - Explore the concept of abstraction
  - Explore the concept of inheritance

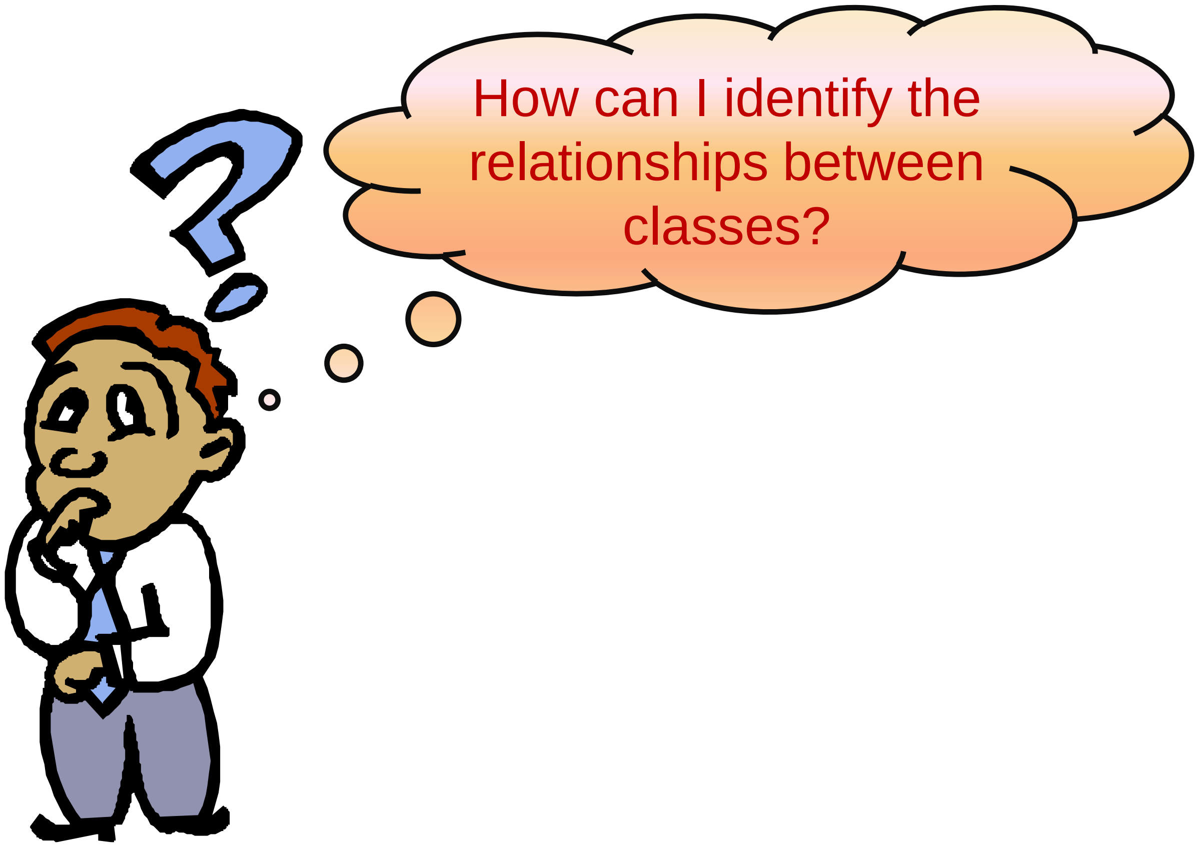
Designing Classes



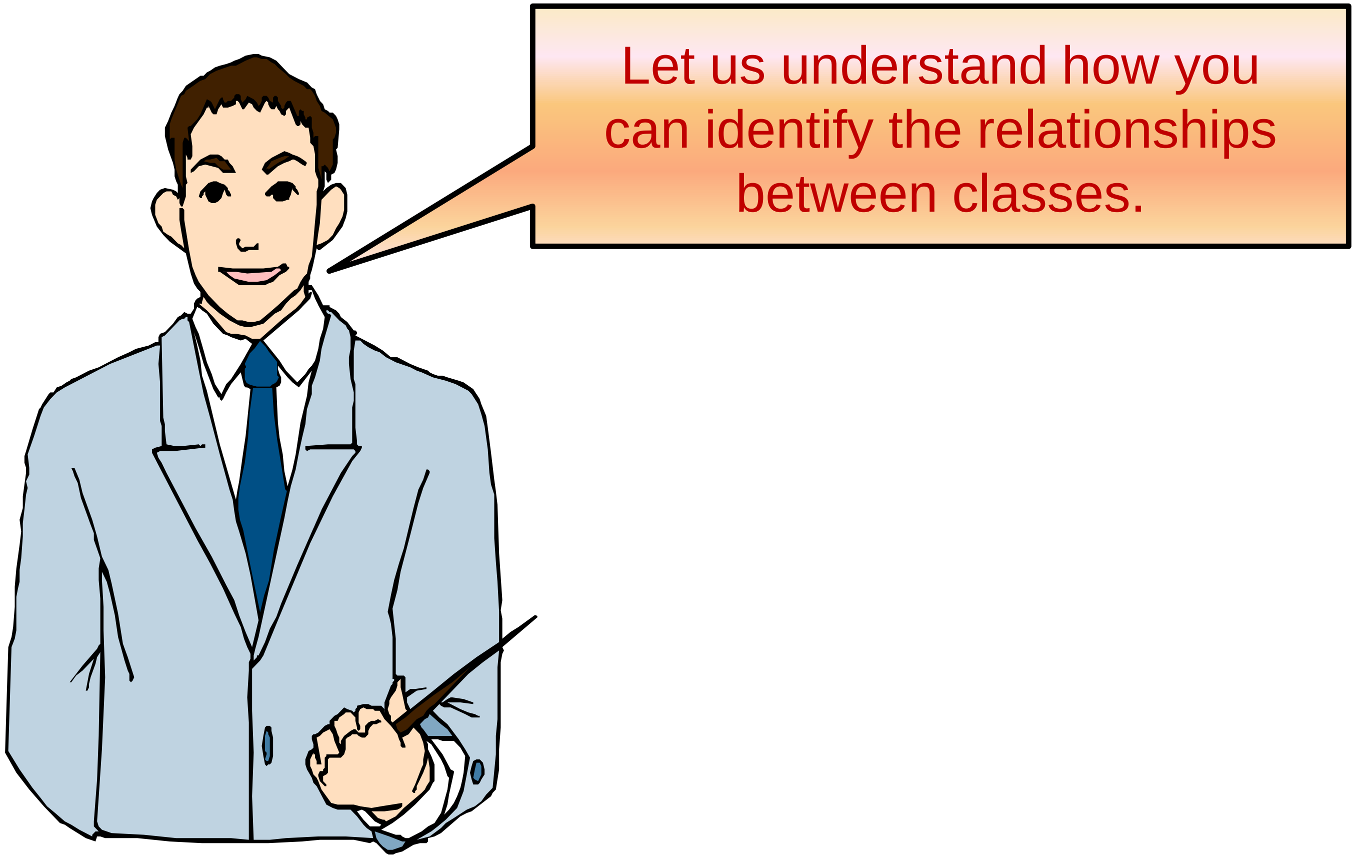
Designing Classes (Contd.)

- Perform the following steps to design classes:
  1. Identify the set of classes that will constitute the software system.
  2. Determine the responsibilities of each class.
  3. Determine the interactions between classes.
  4. Identify the attributes and behavior of each class.
  5. Identify the scope of the class members.

Identifying Relationships Between Classes



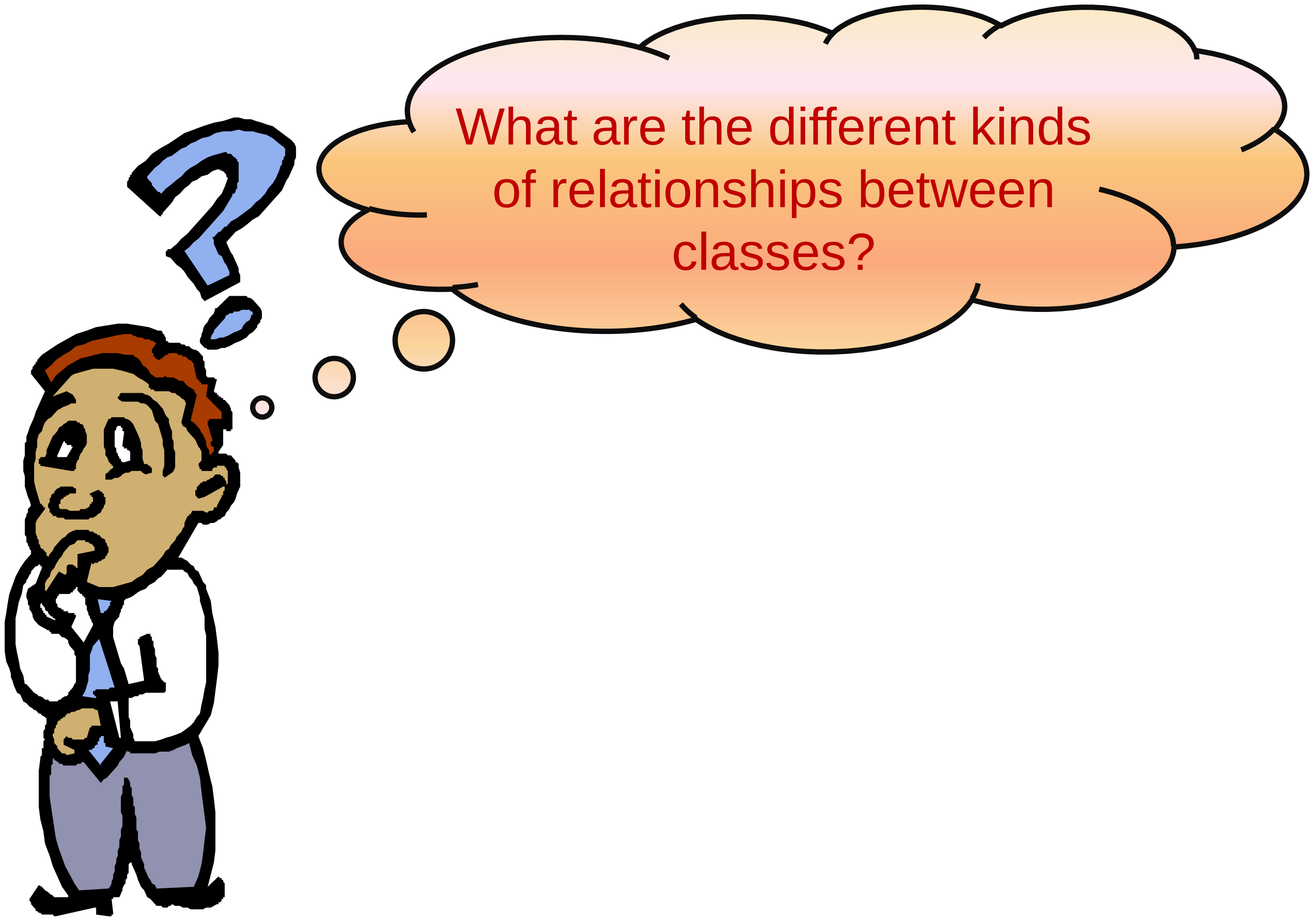
Identifying Relationships Between Classes (Contd.)


Let us understand how you can identify the relationships between classes.

Identifying Relationships Between Classes (Contd.)

- Object-Oriented Programming (OOP):
  - Classes and objects are related to each other
  - Object’s behavior is shown by the action it performs:
    - Behavior finds out the relationship between the objects and classes

Kinds of Relationships

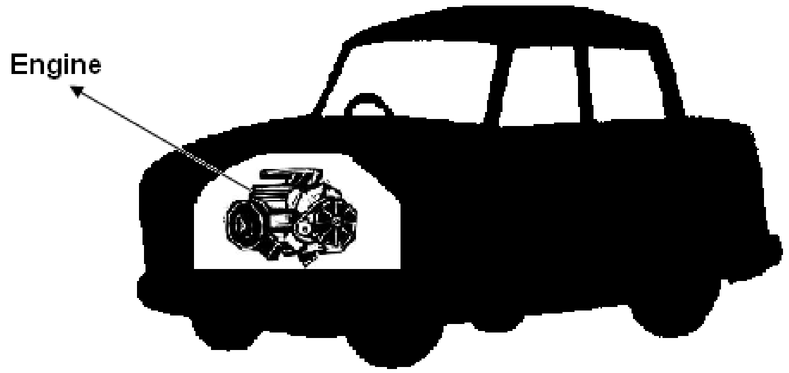

What are the different kinds of relationships between classes?

Kinds of Relationships (Contd.)

- Relationships between the objects of different classes:
  - Composition
  - Generalization
  - Utilization
  - Instantiation

Kinds of Relationships (Contd.)

- Composition relationship:
  - Object includes another object as its part.
  - Contains objects that have a “has-a” relationship.
  - The following figure shows the composition relationship between a car and an engine.

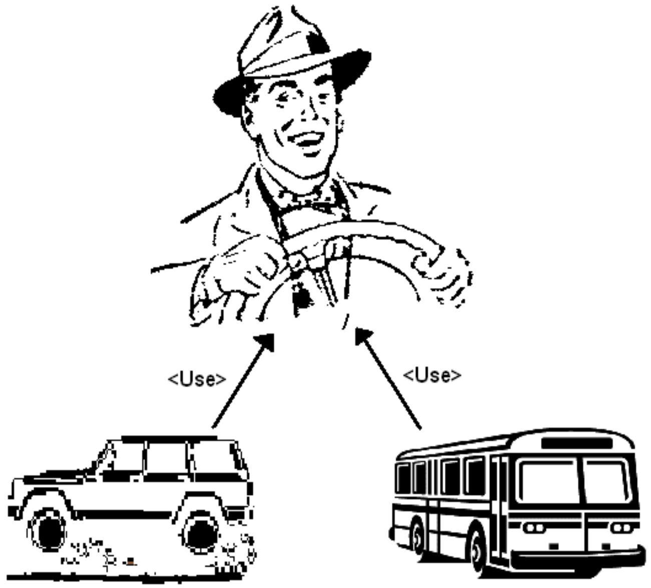


Kinds of Relationships (Contd.)

- Generalization relationship:
  - Classes inherit commonly used state and behavior from other classes
  - Each class allowed to inherit from one or multiple classes
  - Structure and behavior that is common to different classes are combined to form a super class
  - Subclass is a specialized form of the super class
  - Super class is a generalized (common) class
  - Generalization helps to create programs that can be customized in accordance with new requirements

Kinds of Relationships (Contd.)

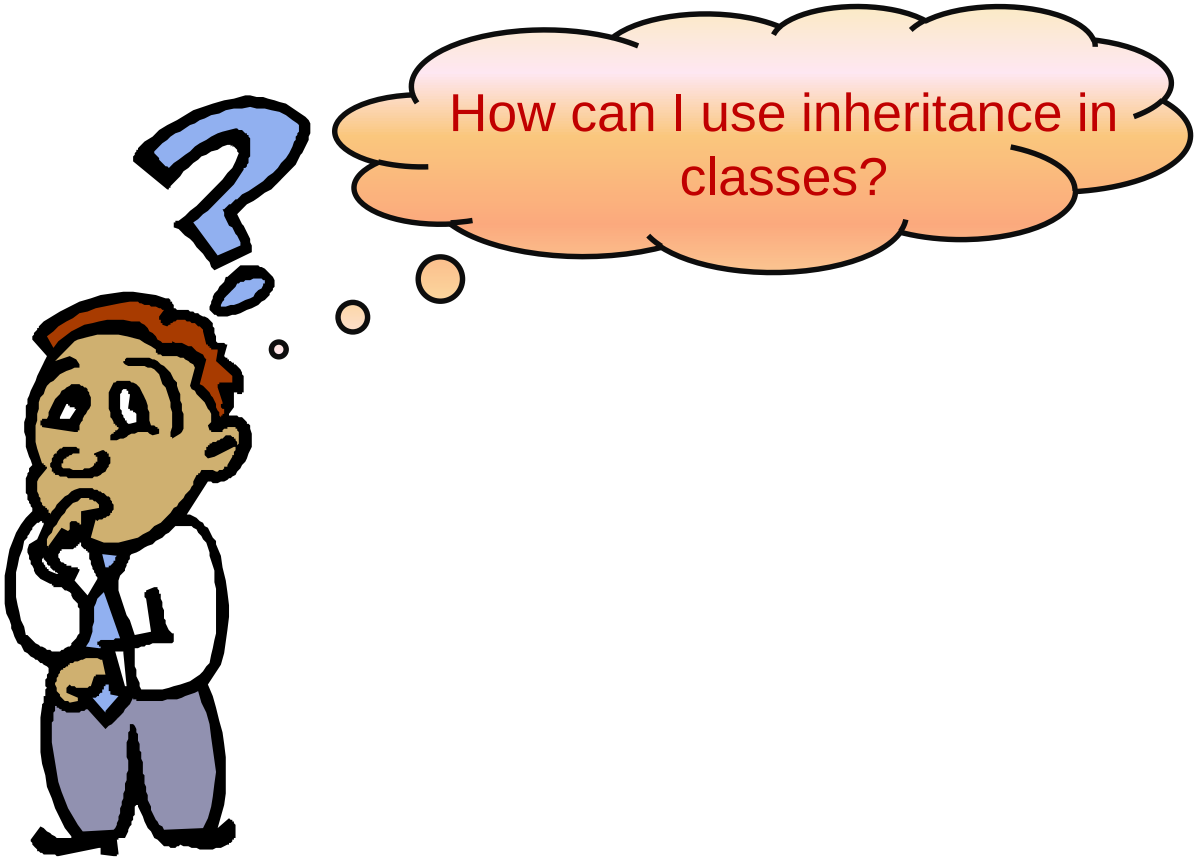
- Utilization relationship:
  - Class uses another class.
  - The following figure shows the utilization relationship between a car and a driver; and a bus and a driver.



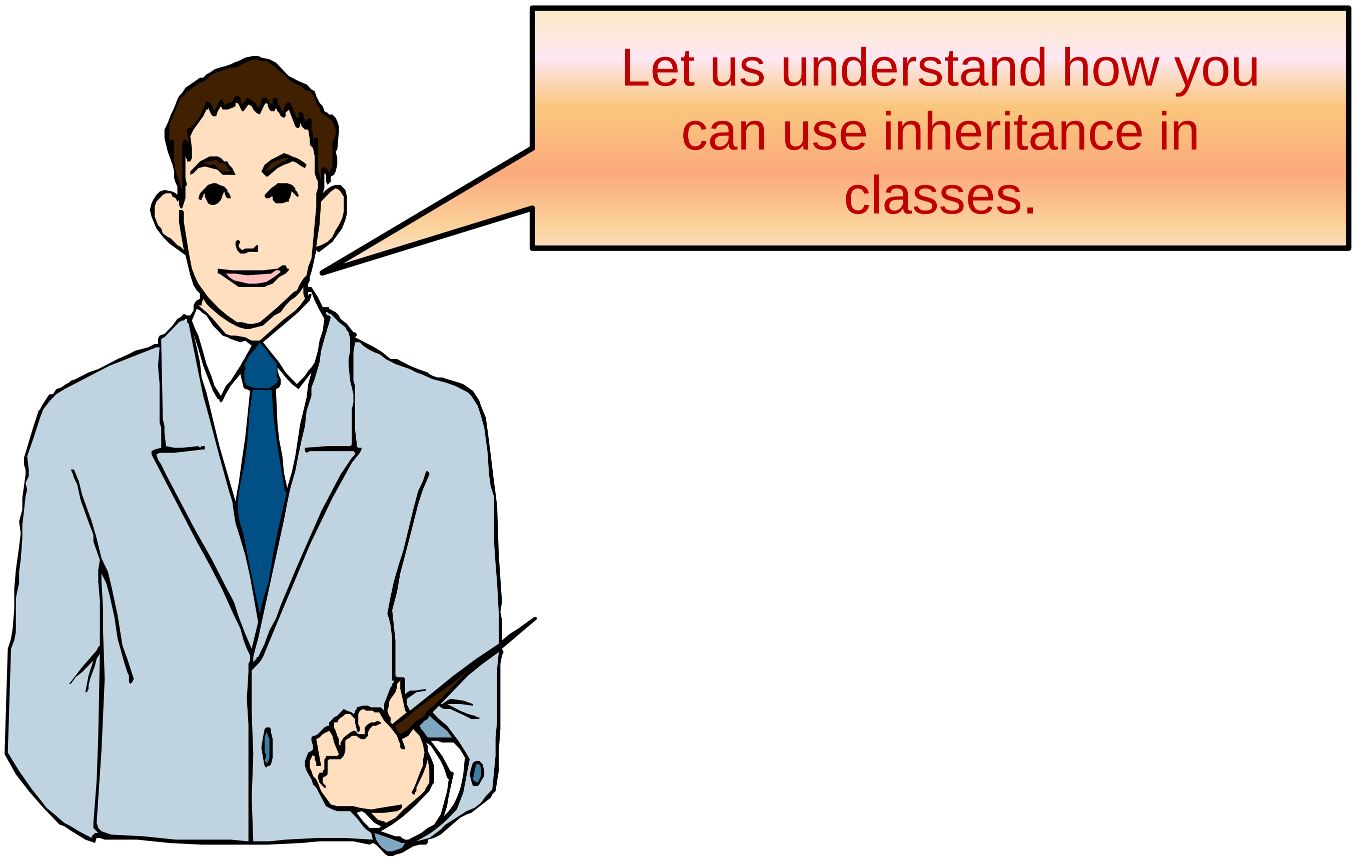
Kinds of Relationships (Contd.)

- Instantiation relationship:
  - Relationship between a class and an instance of that class

Exploring the Concept of Inheritance



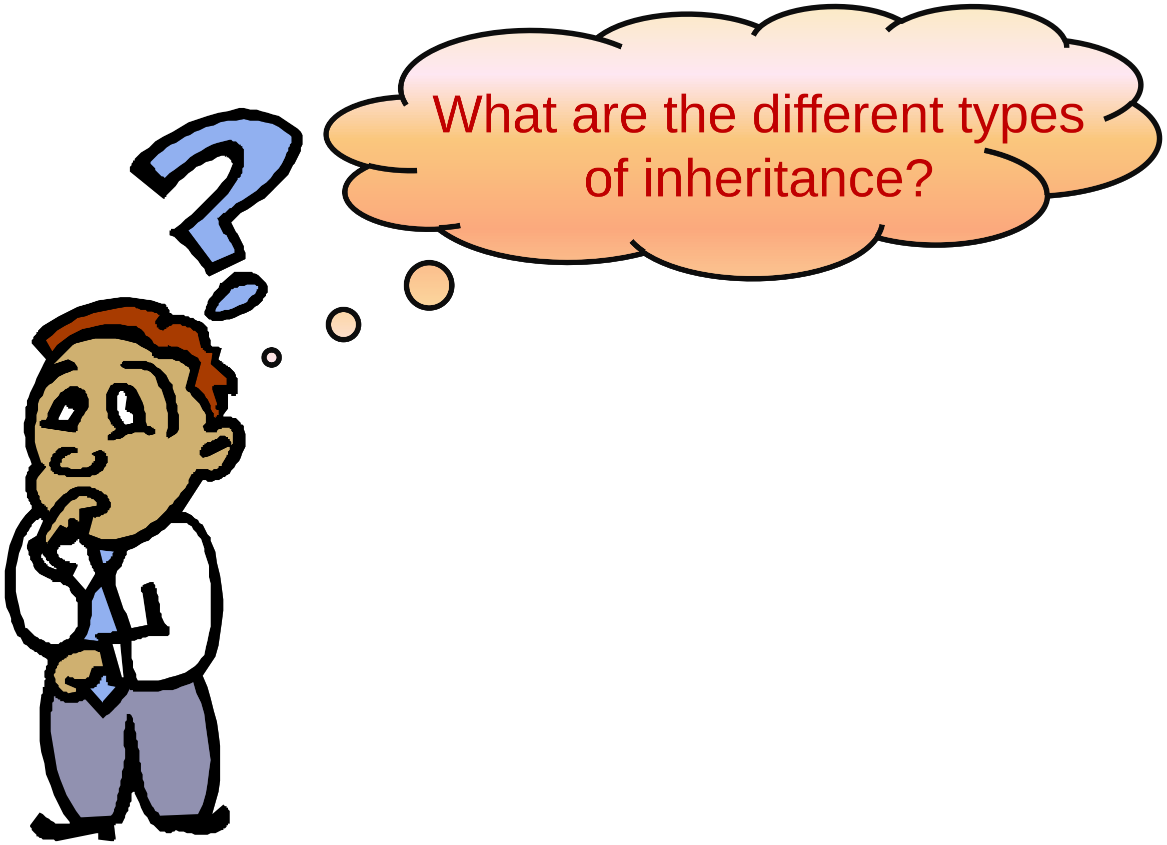
Exploring the Concept of Inheritance (Contd.)



Exploring the Concept of Inheritance (Contd.)

- Inheritance:
  - Subclass objects get the copy of the data members and member functions of the super class
  - Subclass:
    - Class inherits attributes from another class
  - Super class:
    - Class from which attributes are derived

Types of Inheritance

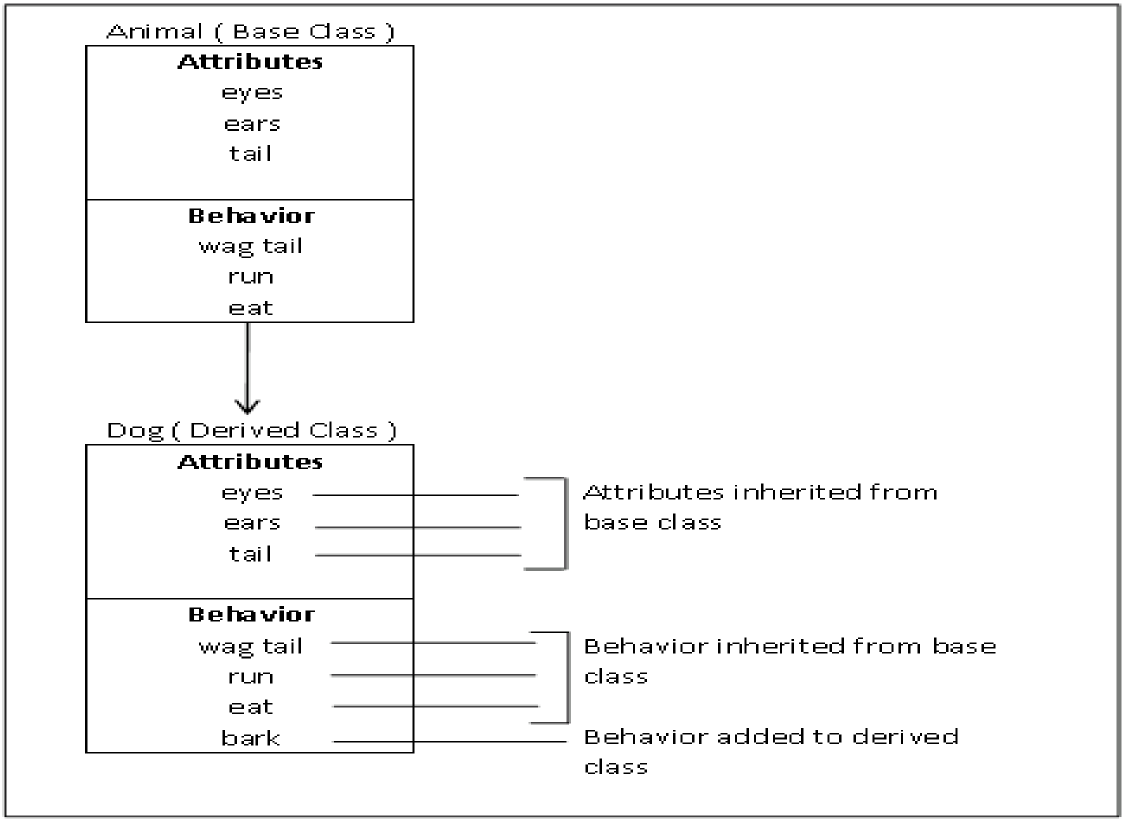


Types of Inheritance (Contd.)

- Types of inheritance:
  - Single inheritance
  - Multiple inheritance
  - Multilevel inheritance

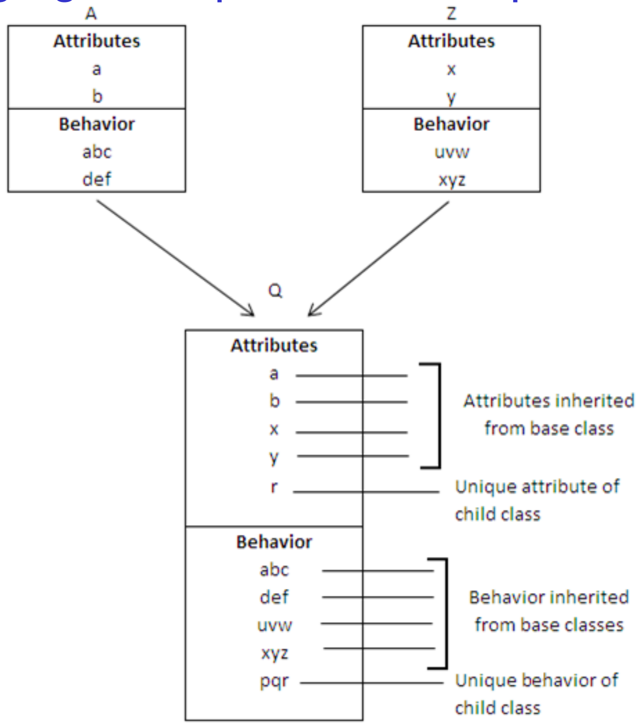
Types of Inheritance (Contd.)

- Single inheritance:
  - One child class has one parent class.
  - The following figure depicts an example of single inheritance.



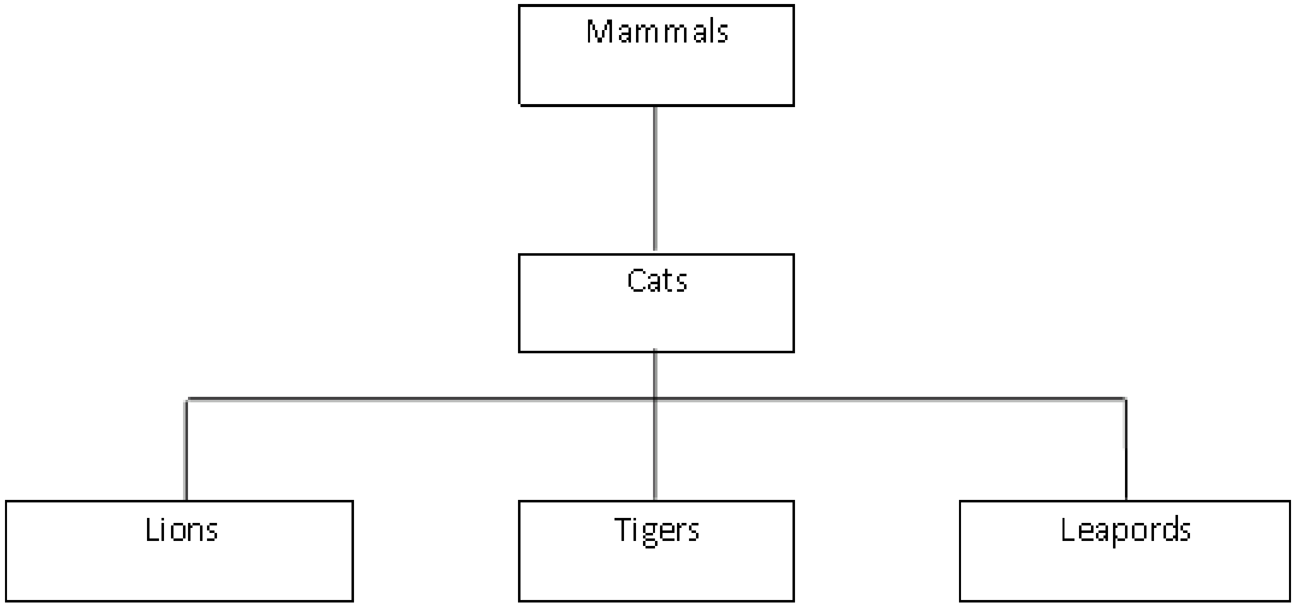
Types of Inheritance (Contd.)

- Multiple inheritance:
  - One child class has more than one parent class.
  - The following figure depicts an example of multiple inheritance.

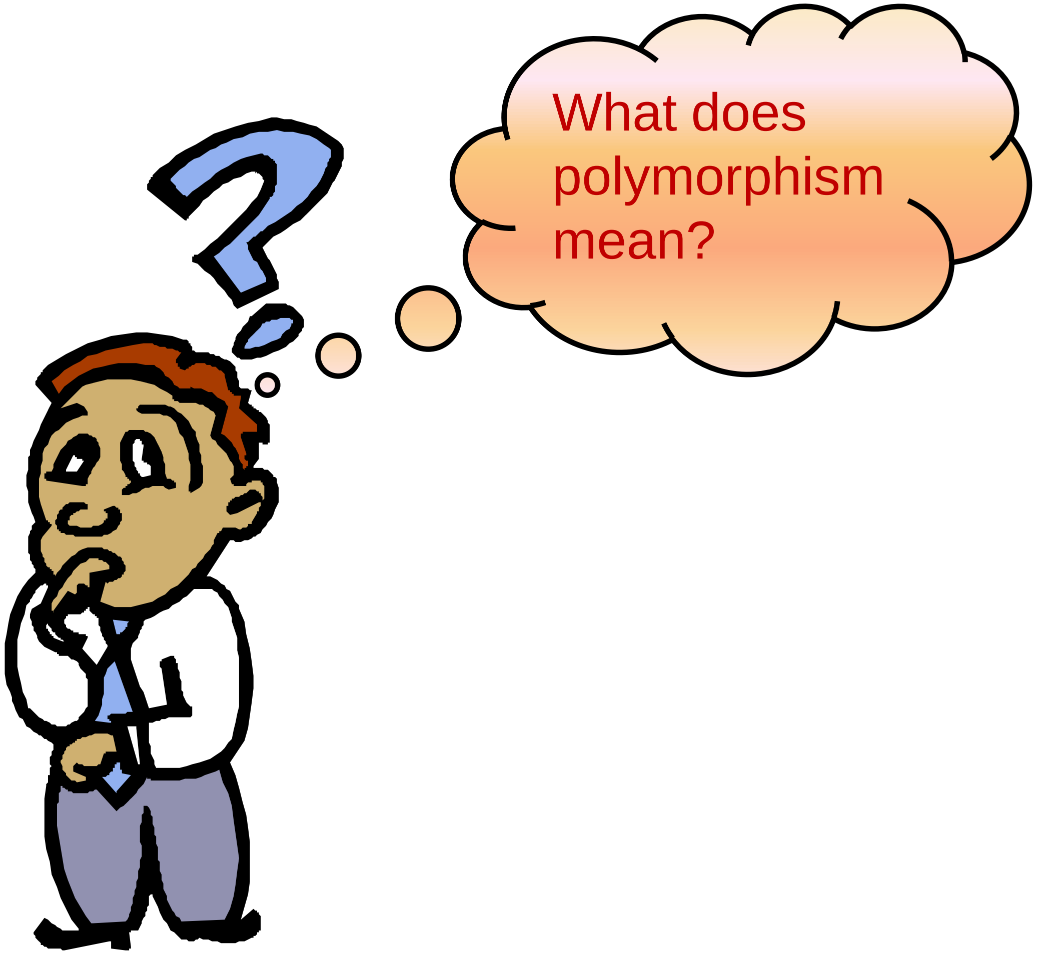

A
Attributes
a
b
Behavior
abc
def
Z
Attributes
x
y
Behavior
uvw
xyz
Q
Attributes
a
b
x
y
r
Behavior
abc
def
uvw
xyz
pqr
Attributes inherited from base class
Unique attribute of child class
Behavior inherited from base classes
Unique behavior of child class

Types of Inheritance (Contd.)

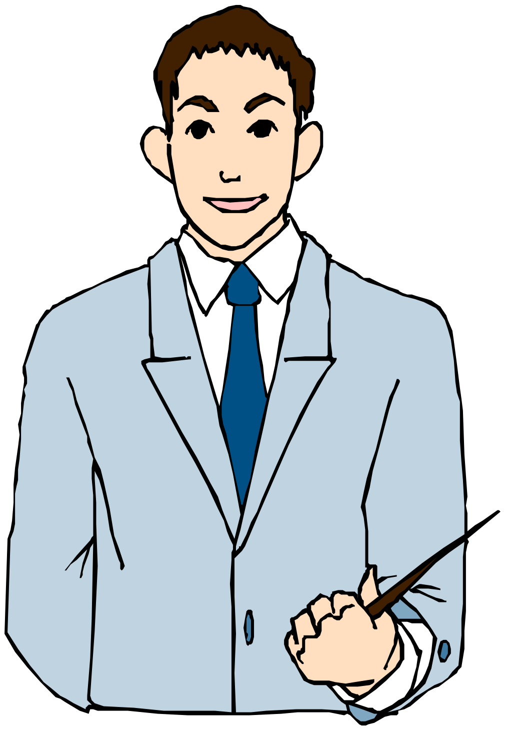
- Multilevel inheritance:
  - Subclass is derived from a derived class.
  - The following figure depicts an example of multilevel inheritance.



Exploring the Concept of Polymorphism



Exploring the Concept of Polymorphism (Contd.)



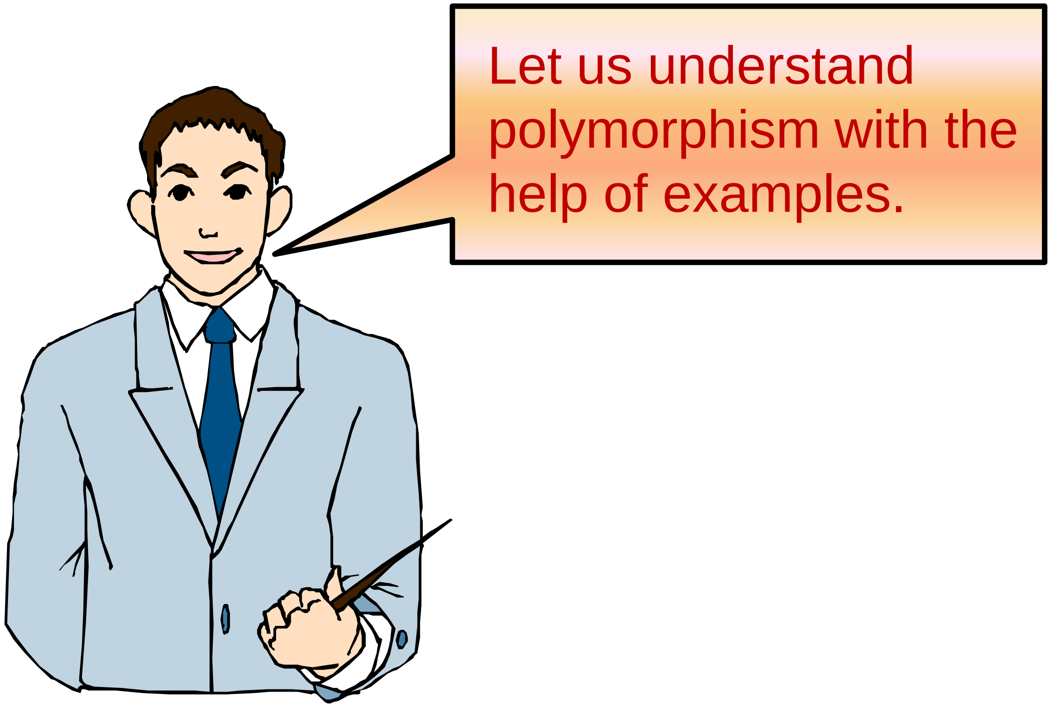
Greek Words and Their Meanings

Poly = Many
Morphos= Forms

Polymorphism = Many Forms

Exploring the Concept of Polymorphism (Contd.)

- Polymorphism:
  - Ability to allow a function to exist in different forms

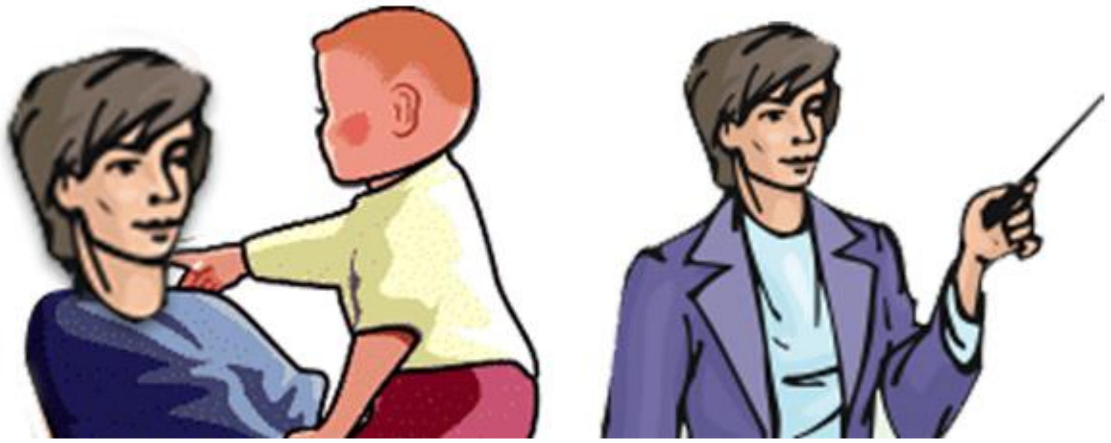


Exploring the Concept of Polymorphism (Contd.)

Different moods of a person depending upon the situation he is in depict polymorphism

Exploring the Concept of Polymorphism (Contd.)

Role of a lady being a mother and an instructor at the same time depicts polymorphism



Summary

- In this session, you learned that:
  - In Object-oriented programming, classes and objects are related to each other.
  - Relationships that exist between the objects of different classes are:
    - Composition
    - Generalization
    - Utilization
    - Instantiation
  - Types of inheritance are:
    - Single inheritance
    - Multiple inheritance
    - Multilevel inheritance