

CHAPTER 11

Inheritance

starting out with >>>

PYTHON®

FOURTH EDITION



TONY GADDIS



Pearson

Copyright © 2018 Pearson Education, Inc.

Topics

- **Inheritance**
- **Polymorphism**

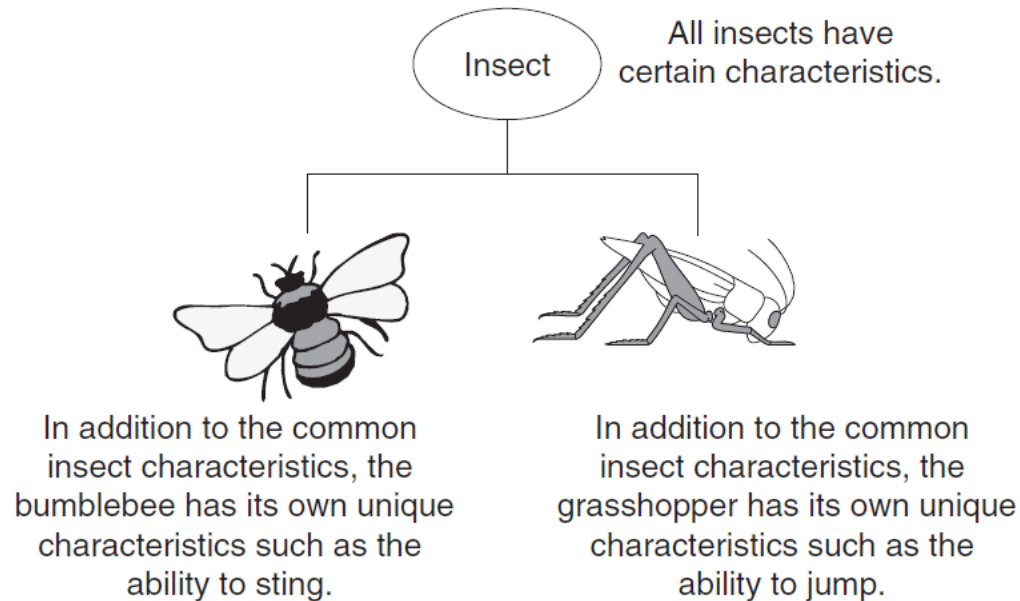
Introduction to Inheritance

- **In the real world, many objects are a specialized version of more general objects**
 - Example: grasshoppers and bees are specialized types of insect
 - In addition to the general insect characteristics, they have unique characteristics:
 - Grasshoppers can jump
 - Bees can sting, make honey, and build hives



Introduction to Inheritance (cont'd.)

Figure 11-1 Bumblebees and grasshoppers are specialized versions of an insect



Inheritance and the “Is a” Relationship

- **“Is a” relationship: exists when one object is a specialized version of another object**
 - Specialized object has all the characteristics of the general object plus unique characteristics
 - Extends the capabilities of a class by creating another class that is specialized version of it

Inheritance and the “Is a” Relationship (cont’d.)

- **Inheritance**: used to create an “is a” relationship between classes
- **Superclass (base class)**: a general class
- **Subclass (derived class)**: a specialized class
 - An extended version of the superclass
 - Inherits attributes and methods of the superclass without having to be rewritten
 - New attributes and methods can be added



Inheritance and the “Is a” Relationship (cont’d.)

- **For example, need to create classes for cars, pickup trucks, and SUVs**
- **All are automobiles**
 - Have a make, year model, mileage, and price
 - This can be the attributes for the base class
- **In addition:**
 - Car has a number of doors
 - Pickup truck has a drive type
 - SUV has a passenger capacity



Inheritance and the “Is a” Relationship (cont’d.)

- **In a class definition for a subclass:**
 - To indicate inheritance, the superclass name is placed in parentheses after subclass name
 - Example: `class Car(Automobile):`
 - The initializer method of a subclass calls the initializer method of the superclass and then initializes the unique data attributes
 - Add method definitions for unique methods

`car_demo.py`

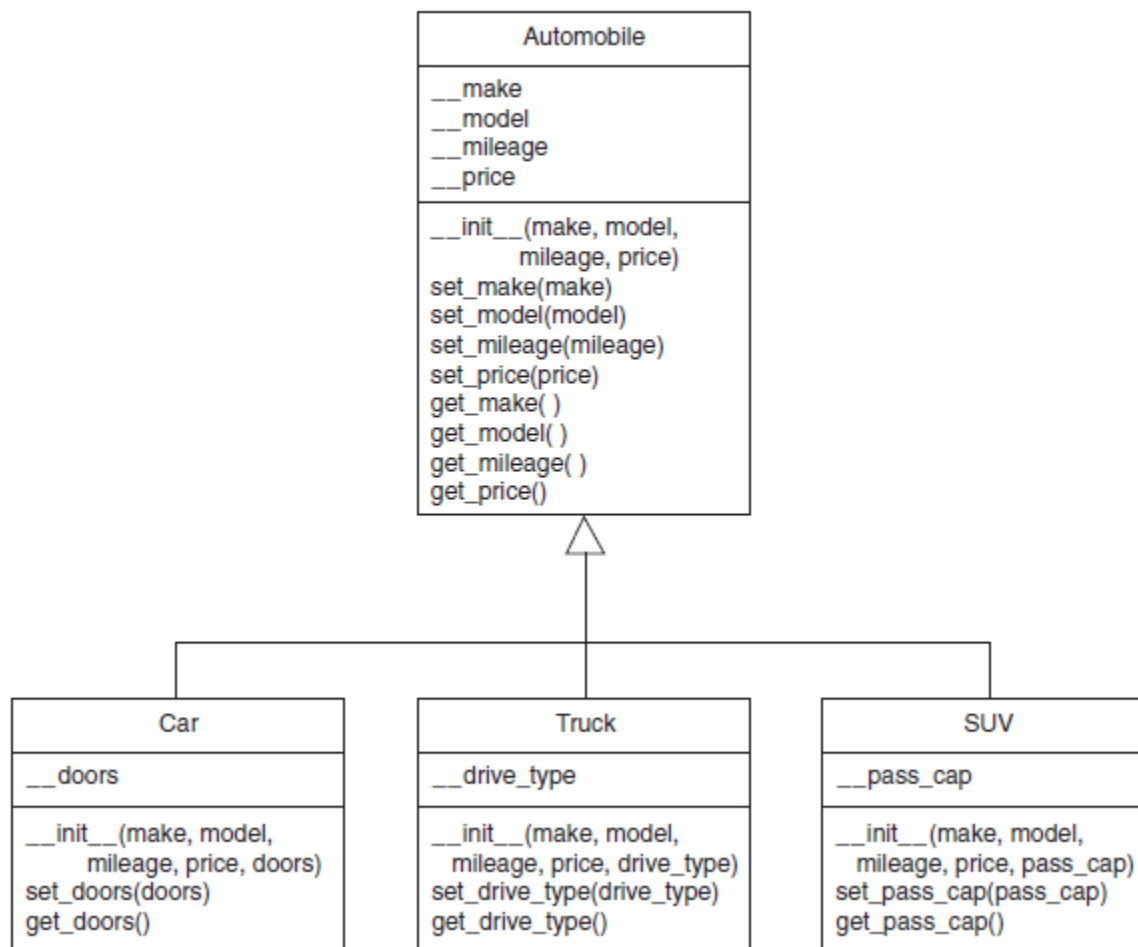
`car_truck_suv_demo.py`



Inheritance in UML Diagrams

- In UML diagram, show inheritance by drawing a line with an open arrowhead from subclass to superclass

Figure 11-2 UML diagram showing inheritance



Polymorphism

- **Polymorphism**: an object's ability to take different forms
- **Ability to define a method in a superclass and override it in a subclass**
 - Subclass defines method with the same name
- **Ability to call the correct version of overridden method depending on the type of object that called for it**

Polymorphism (cont'd.)

- In previous inheritance examples showed how to override the `__init__` method
 - Called superclass `__init__` method and then added onto that
- The same can be done for any other method
 - The method can call the superclass equivalent and add to it, or do something completely different

[animals.py](#)
[polymorphism_demo.py](#)



Pearson

Copyright © 2018 Pearson Education, Inc.

The `isinstance` Function

- Polymorphism provides great flexibility when designing programs
- `AttributeError` exception: raised when a method receives an object which is not an instance of the right class
wrong_type.py
- `isinstance` function: determines whether object is an instance of a class
 - Format: `isinstance(object, class)`

polymorphism_demo2.py



Summary

- **This chapter covered:**
 - Inheritance, including:
 - “Is a” relationships
 - Subclasses and superclasses
 - Defining subclasses and initializer methods
 - Depicting inheritance in UML diagrams
 - Polymorphism
 - The `isinstance` function