Chapter 10

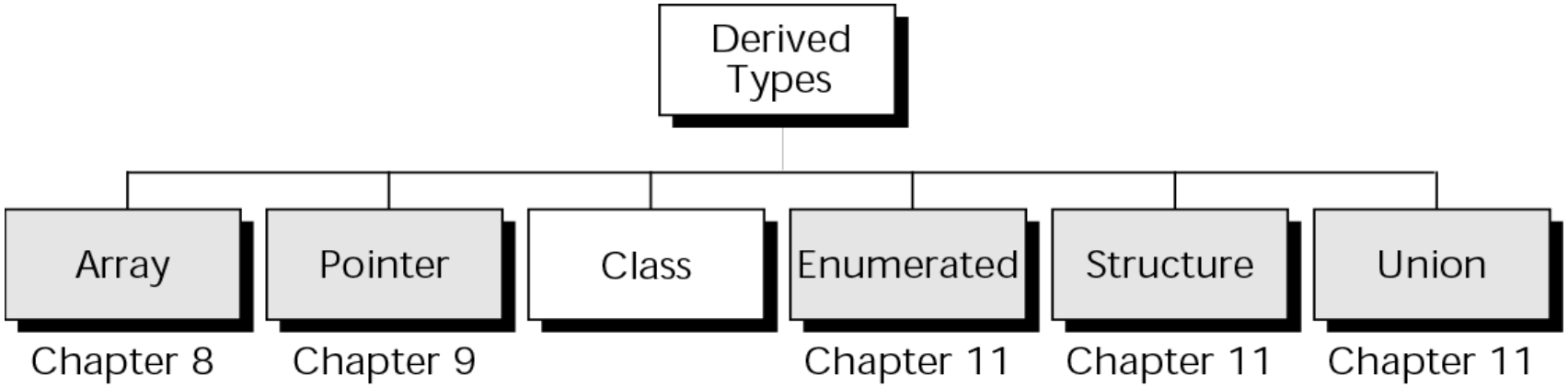
Classes

OBJECTIVES

After studying this chapter you will be able to:

- **Write programs that use simple class structures.**
- **Declare class types.**
- **Understand and use the *private*, protected, and *public* access.**
- **Define a function as a member of a class using the scope resolution operator.**
- **Access class members using the member operator.**
- **Write class constructors and destructors.**
- **Understand and explain manager, mutator, and accessor functions.**
- **Use the Unified Modeling Language (UML) to design and document classes.**

Figure 10-1 Derived types



BASIC CLASS CONCEPTS

Figure 10-2 Class access specifiers

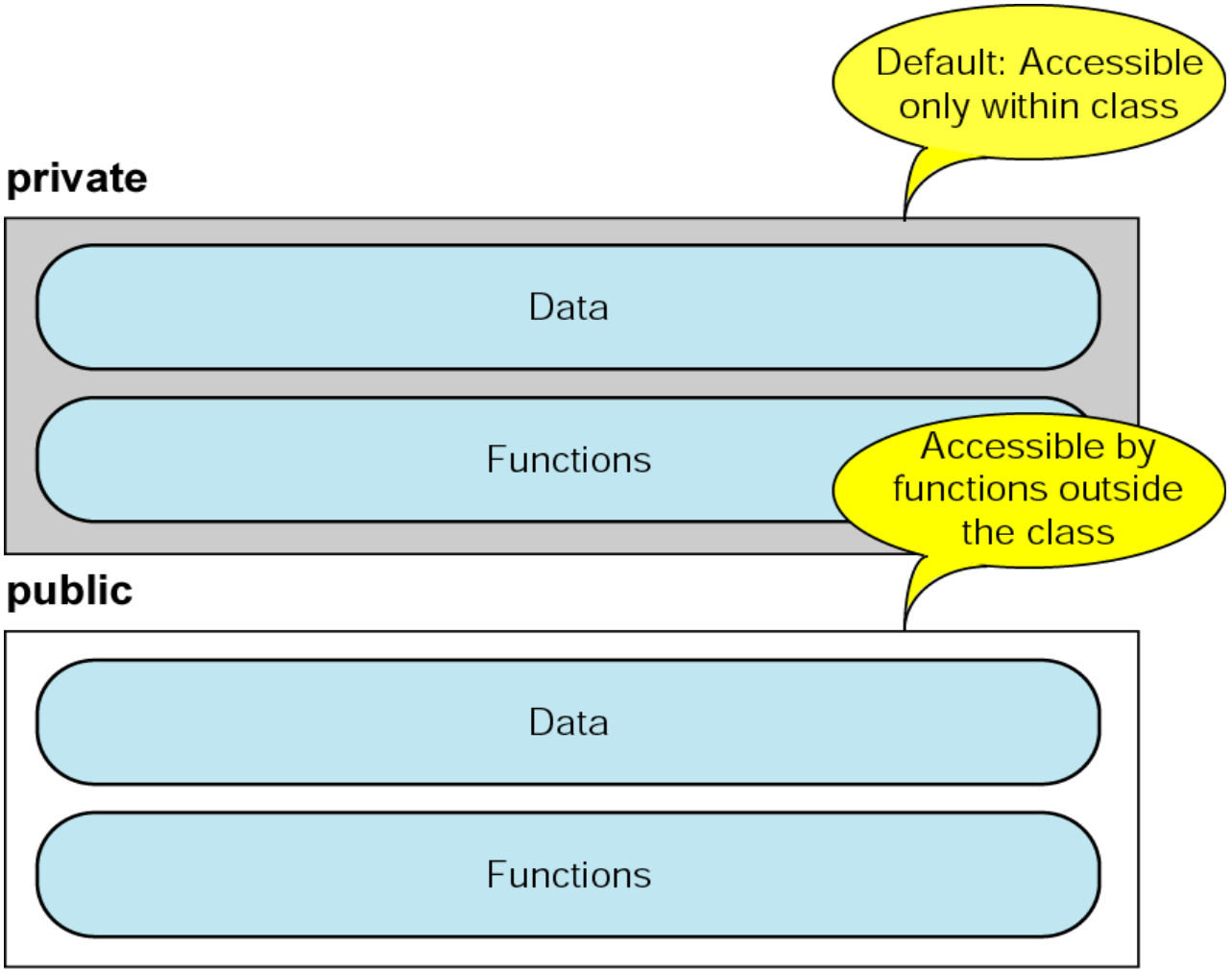
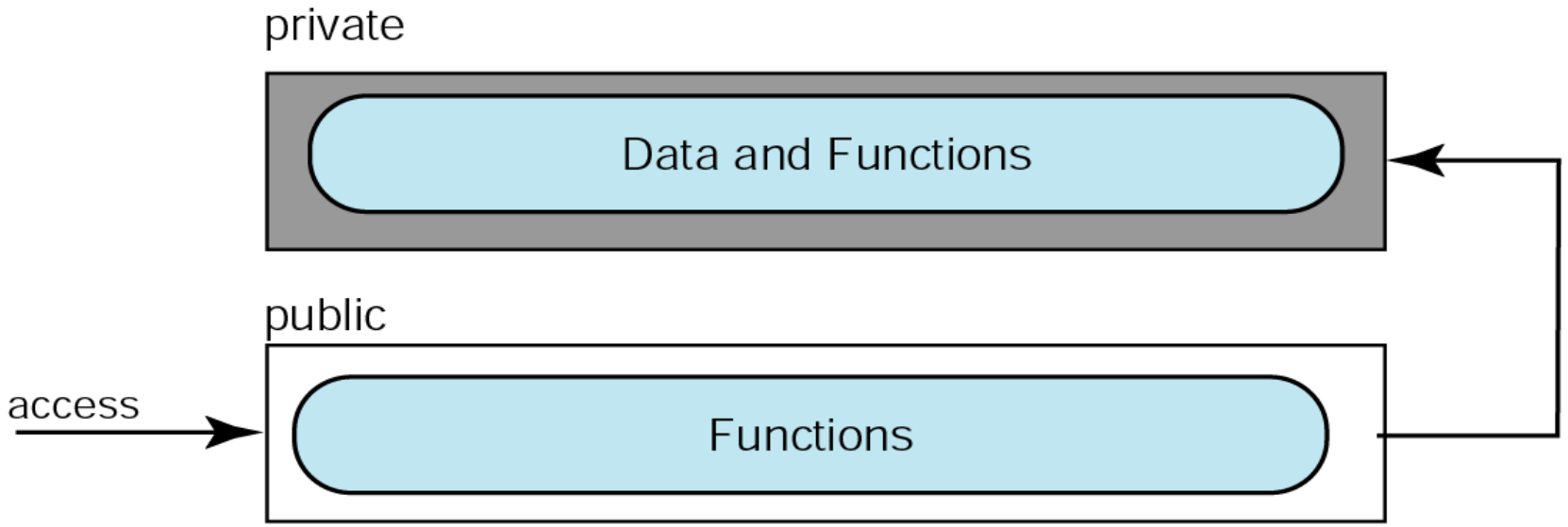


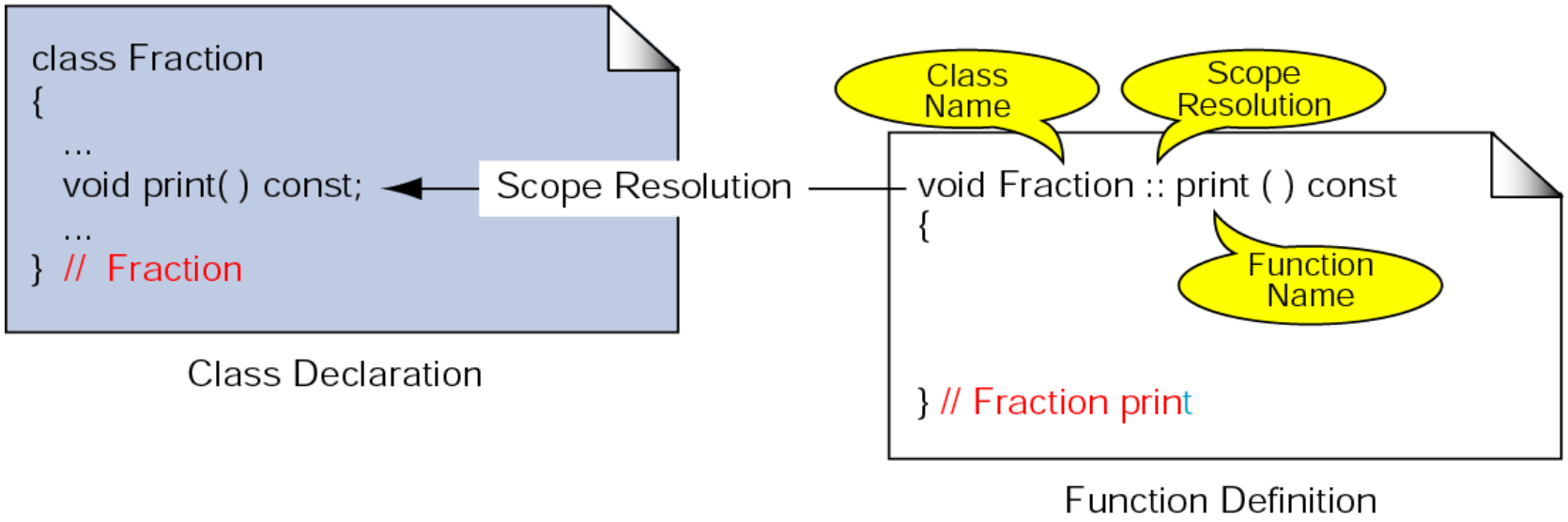
Figure 10-3 Accessing data in a class



Note:

We recommend that class identifiers start with an uppercase letter to ensure uniqueness and to help identify the type as a class.

Figure 10-4 Scope resolution operator



CLASS OBJECTS

Note:

Use the scope resolution operator to refer to a specific class type and use the member operator to refer to data or functions within an instantiated class object.

Figure 10-5 Class objects

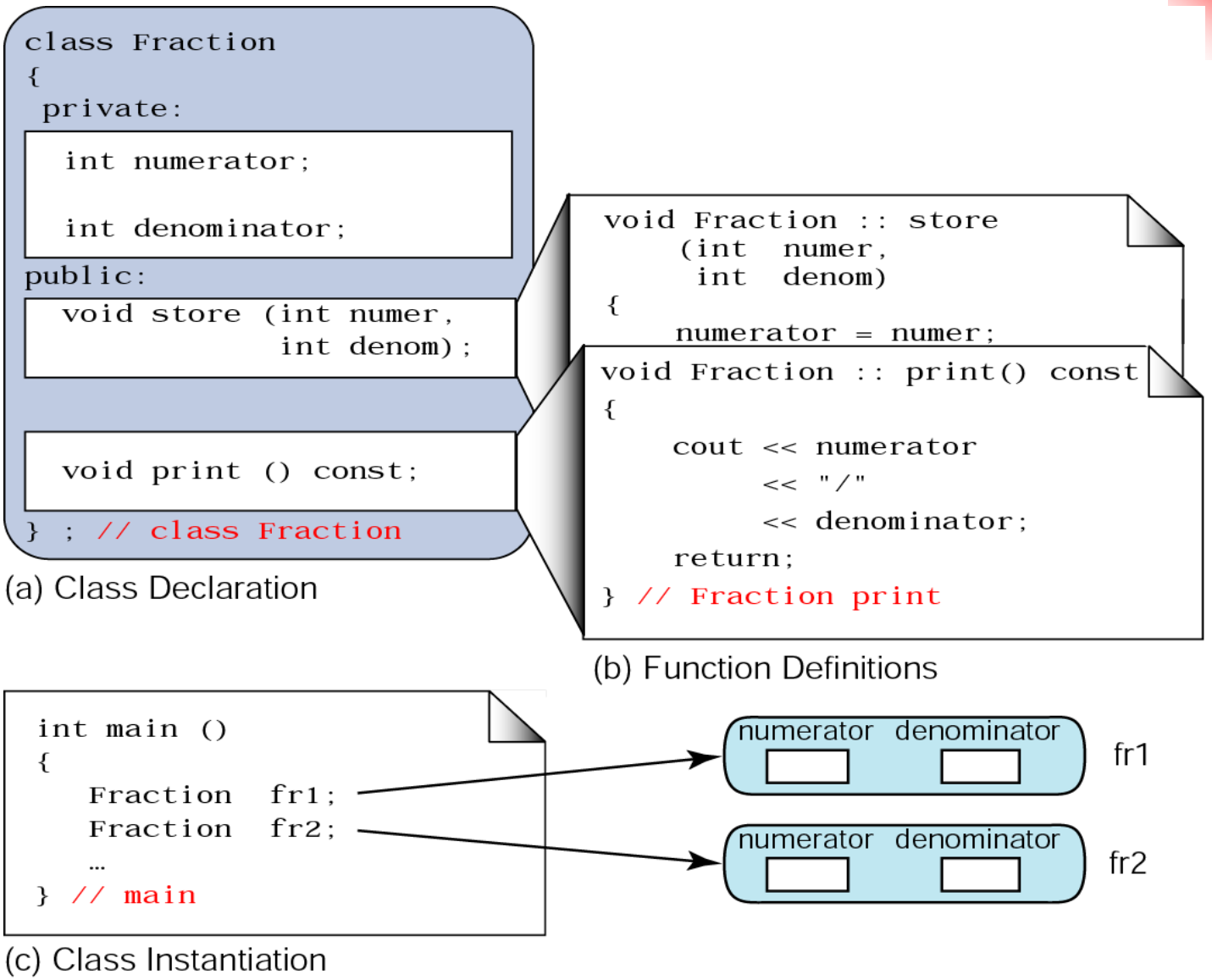
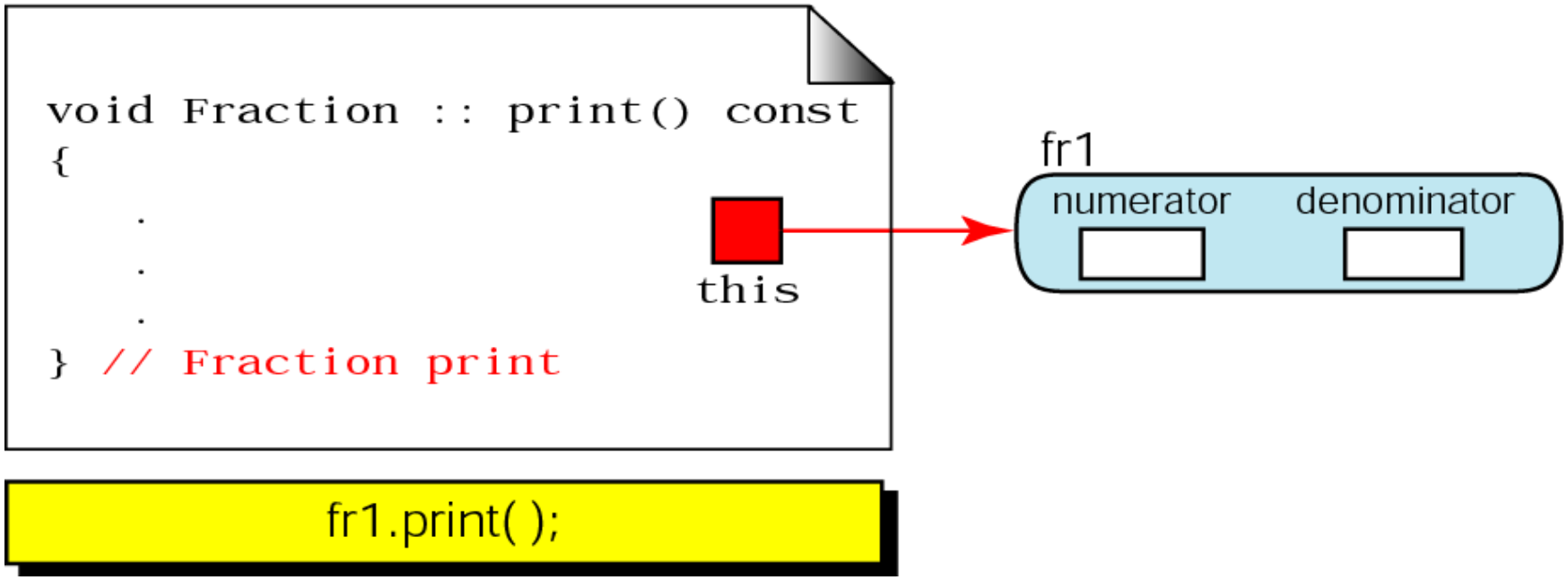
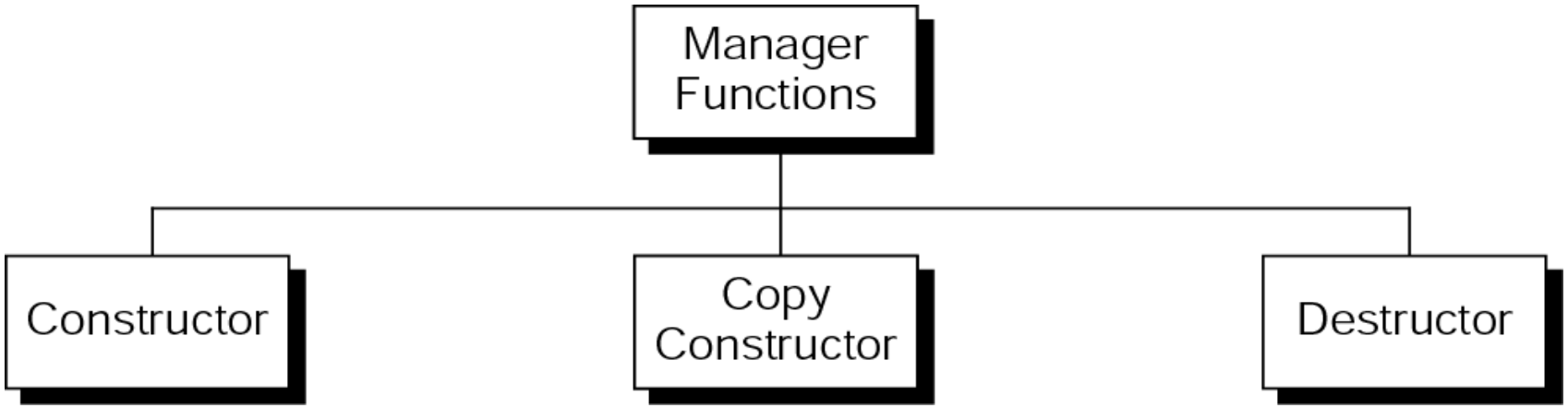


Figure 10-6 The *this* pointer



MANAGER FUNCTIONS

Figure 10-7 Manager Functions



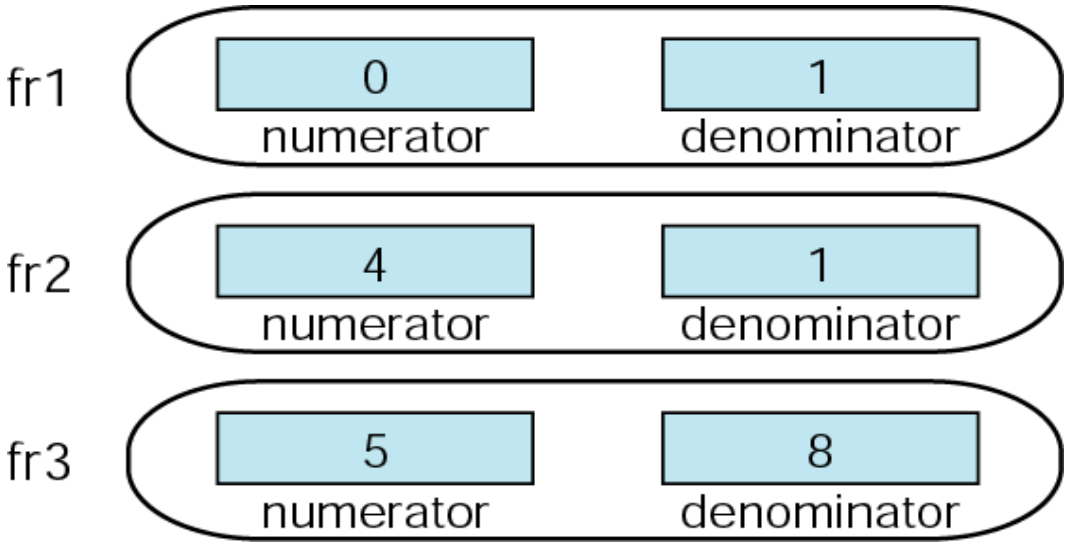
Note:

A constructor is called every time an object is instantiated.

Note:

The name of a constructor is the same as the name of the class and it may not have a return type.

Figure 10-8 Fractions created by Program 10-7



Note:

A class must have at least one constructor, either defined by the program or by the compiler.

Note:

The default constructor is called whenever an object is crated without passing any arguments.

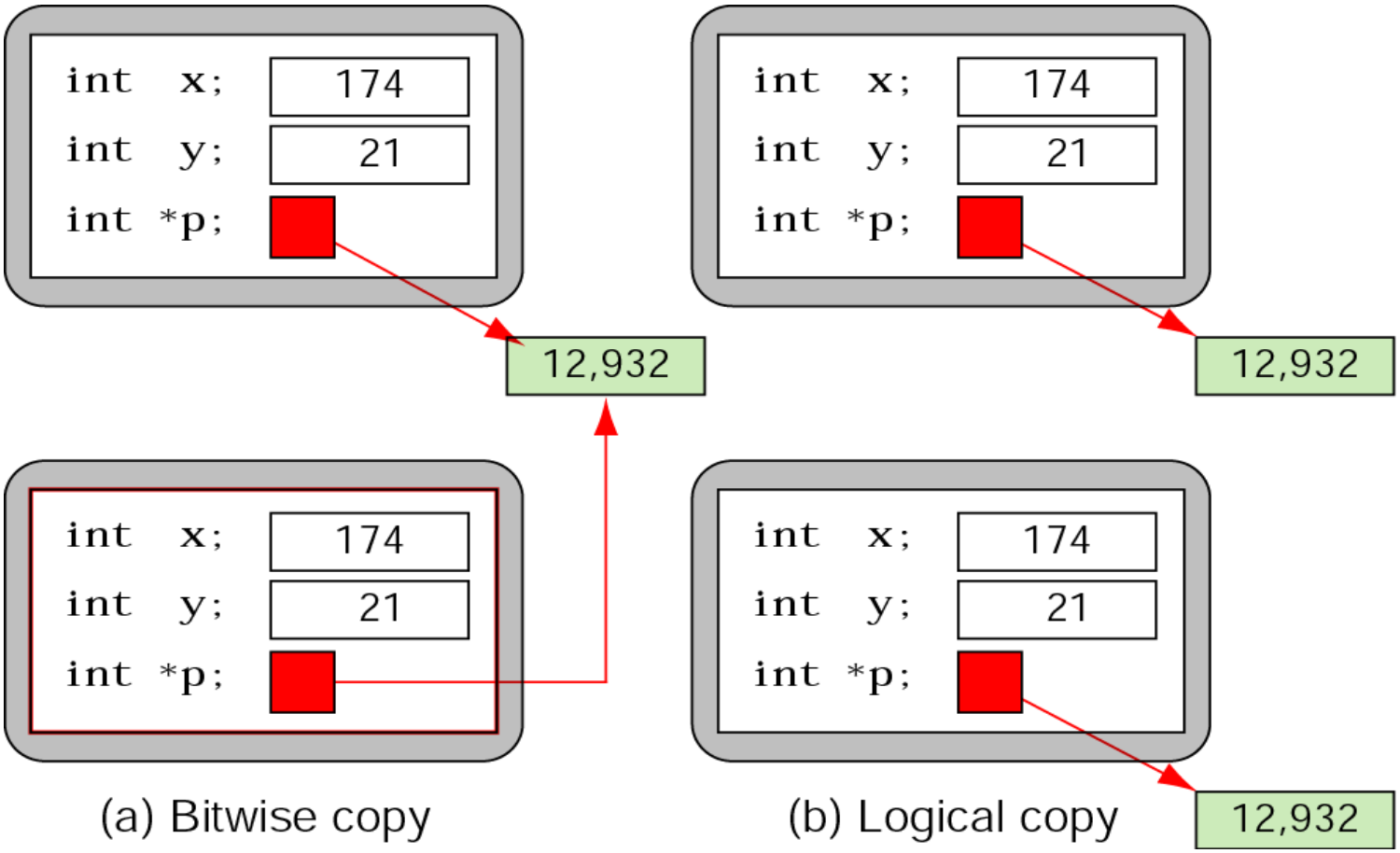
Note:

If we define any type of constructor, we must also define a default constructor if it is needed.

Note:

There is only one default constructor; it cannot be overloaded.

Figure 10-9 Bitwise and logical copies



Note:

There is always one copy constructor available in a class.

Note:

Always pass objects to a function by reference; when the object cannot be changed, pass it as a constant.

Note:

Always return objects by value.

Note:

A class can have one, and only one, destructor.

Mutators and Accessors

Class Invariants

Figure 10-10 Class invariants

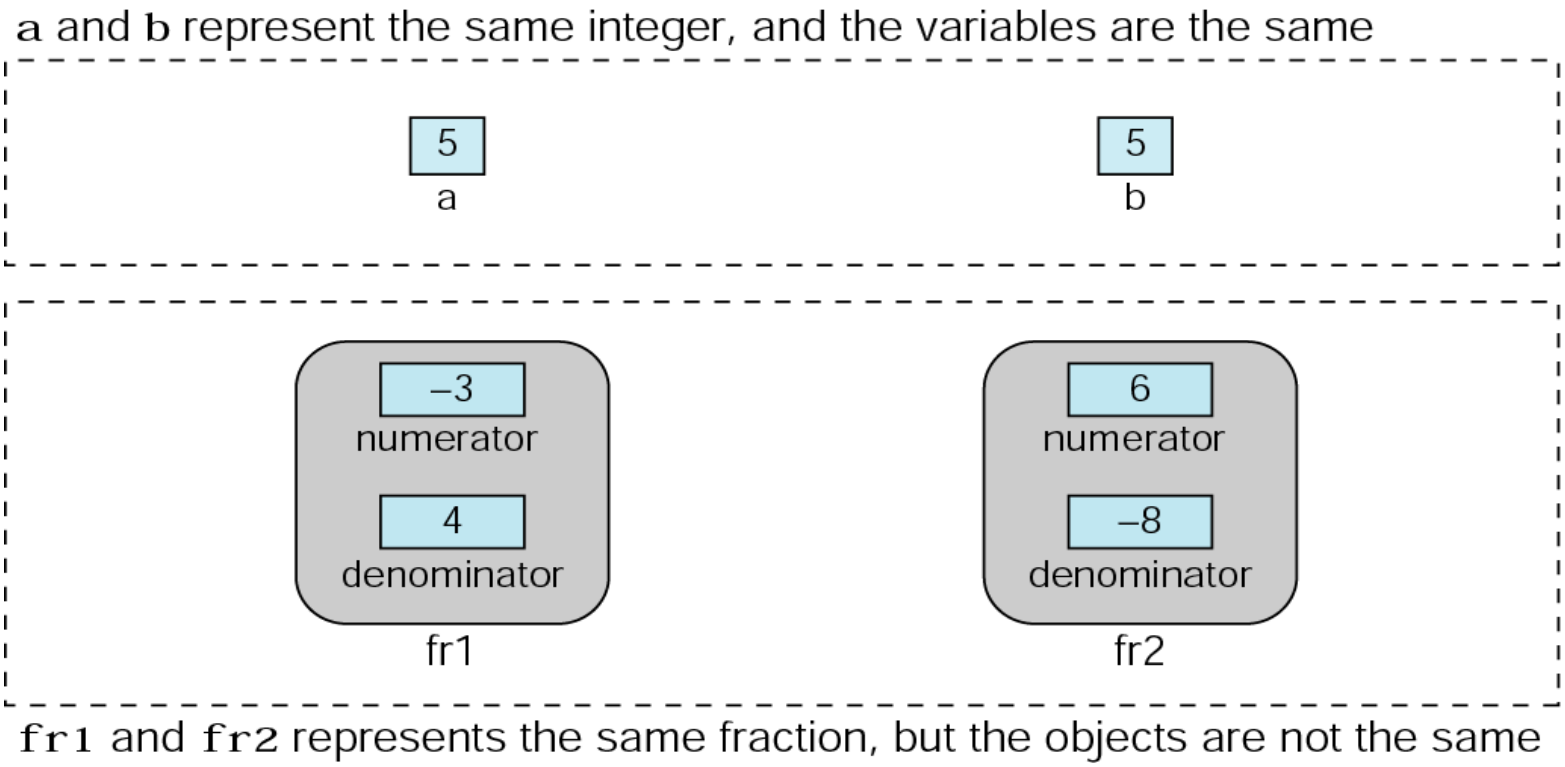
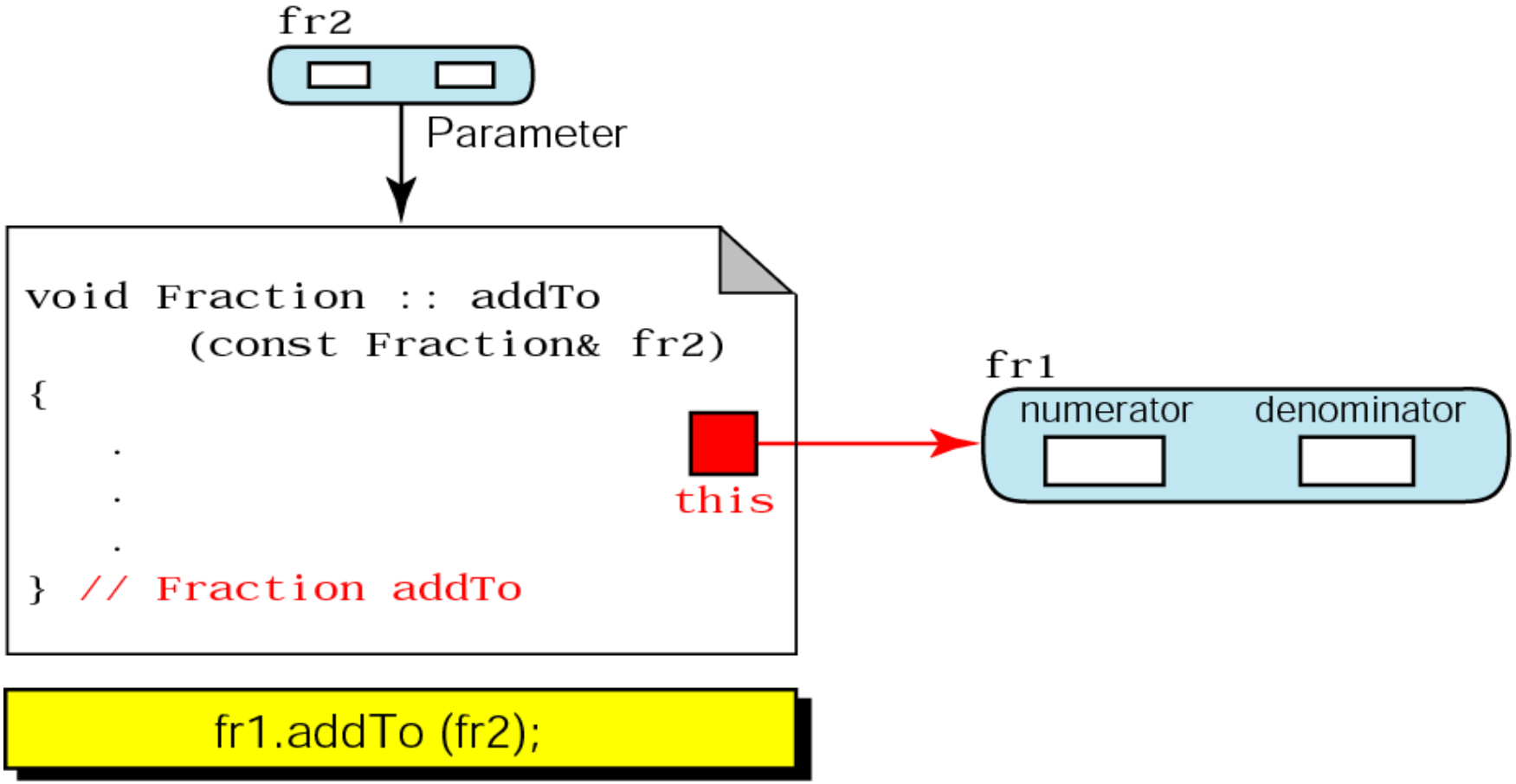


Figure 10-11 Greatest common divisor

$$\gcd(x, y) = \begin{cases} x & \text{if } y = 0 \\ \gcd(y, x \bmod y) & \text{otherwise} \end{cases}$$

COMPLEX CLASS FUNCTIONS

Figure 10-12 Add two fractions



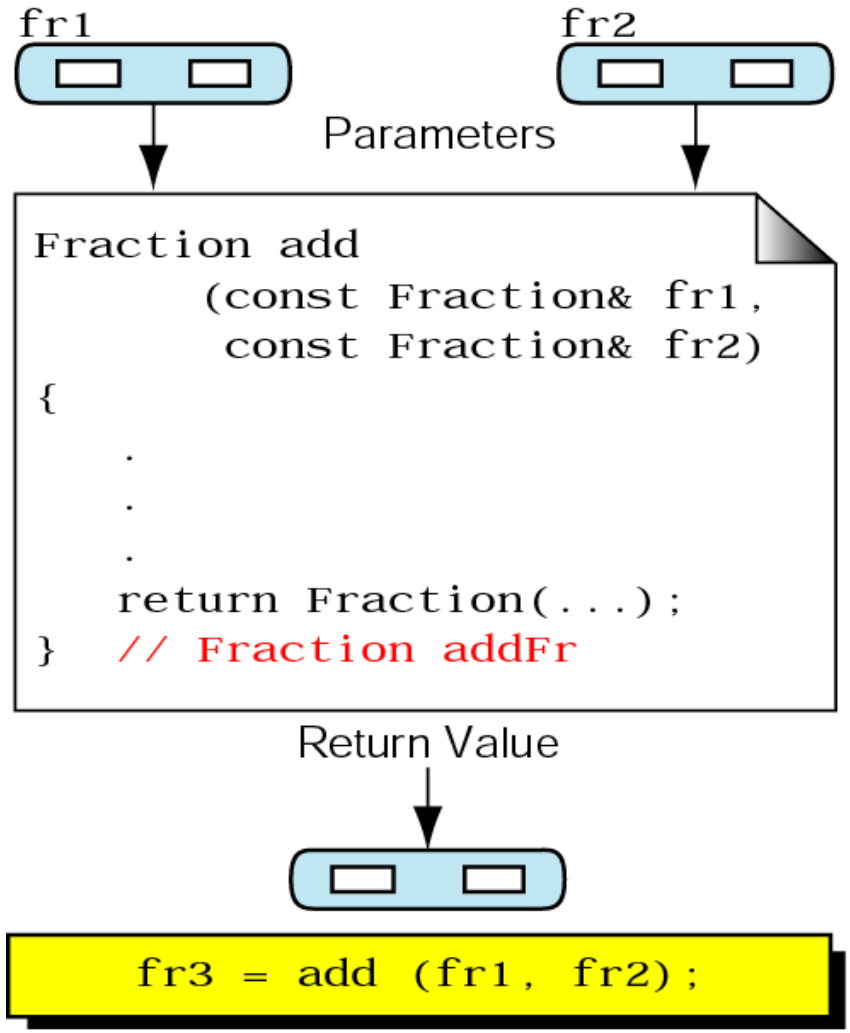
Note:

Functions that are members of a class can only be called through objects of that class. They cannot be called independently.

Note:

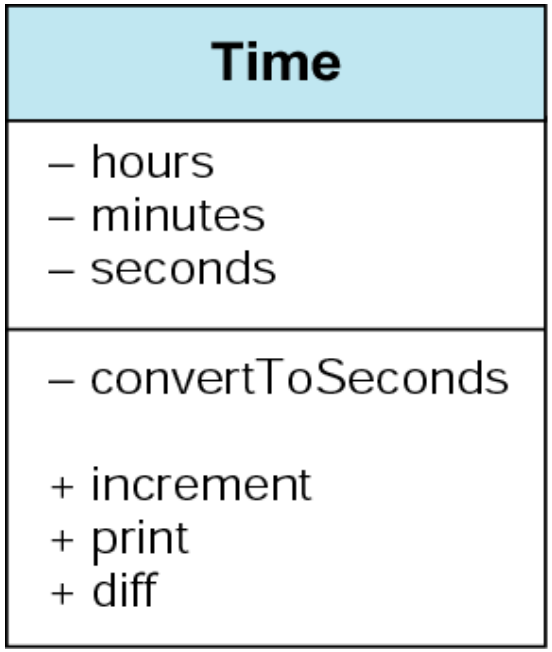
Friend functions are not members of a class, but are associated with it.

Figure 10-13 Add and return fraction value



PROGRAMMING APPLICATIONS

Figure 10-14 Time class design



Access:
– private
+ public

SOFTWARE ENGINEERING AND PROGRAMMING STYLE

Figure 10-15 UML views

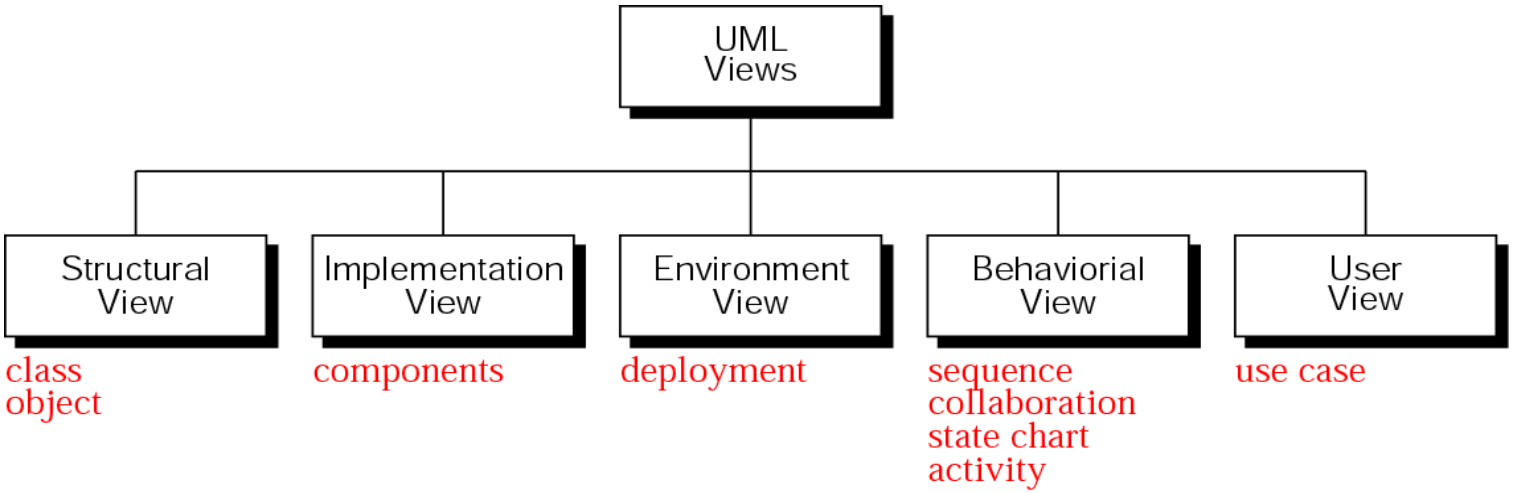
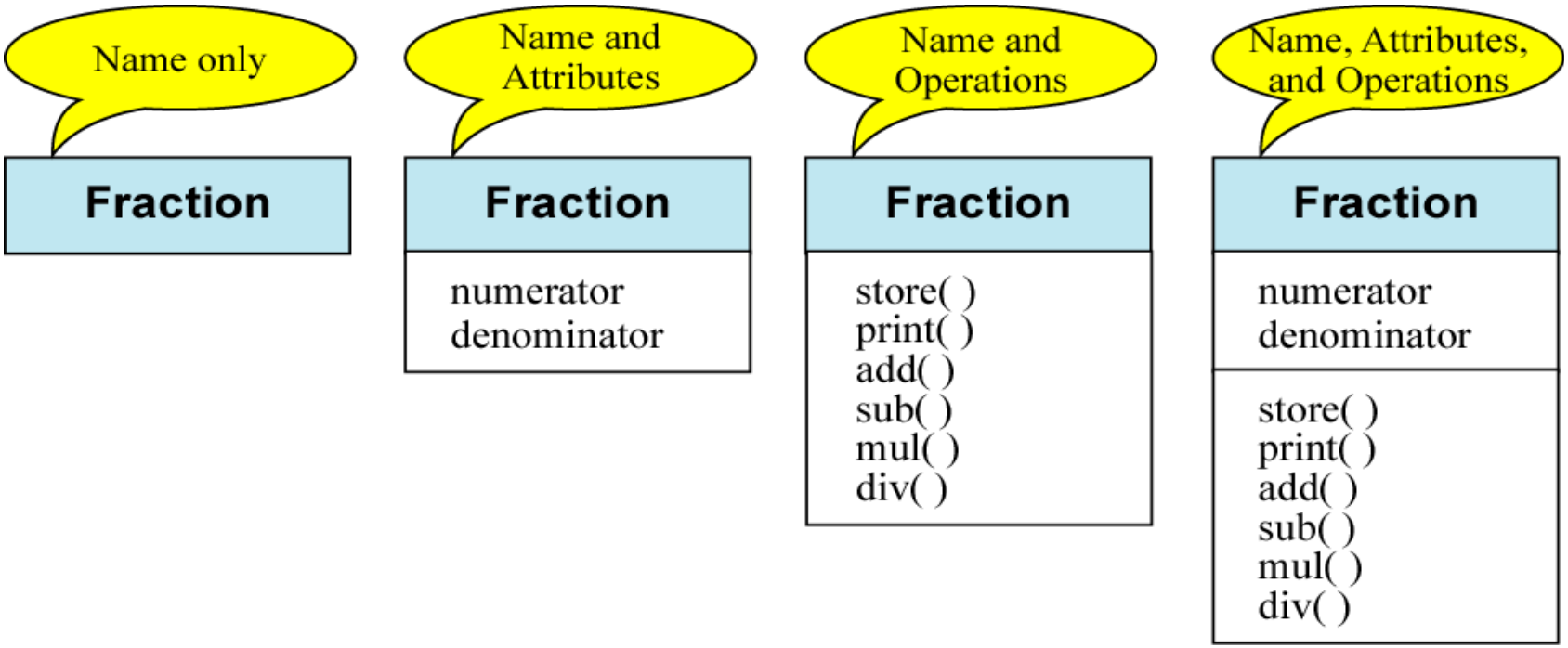


Figure 10-16 Class diagrams



Note:

The name compartment is mandatory in a class diagram.

Figure 10-17 Expanded class diagram

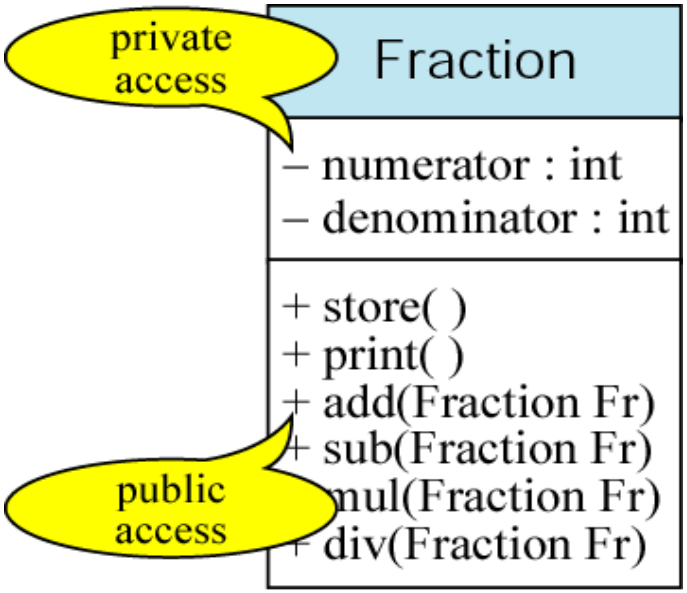


Figure 10-18 An abstract class

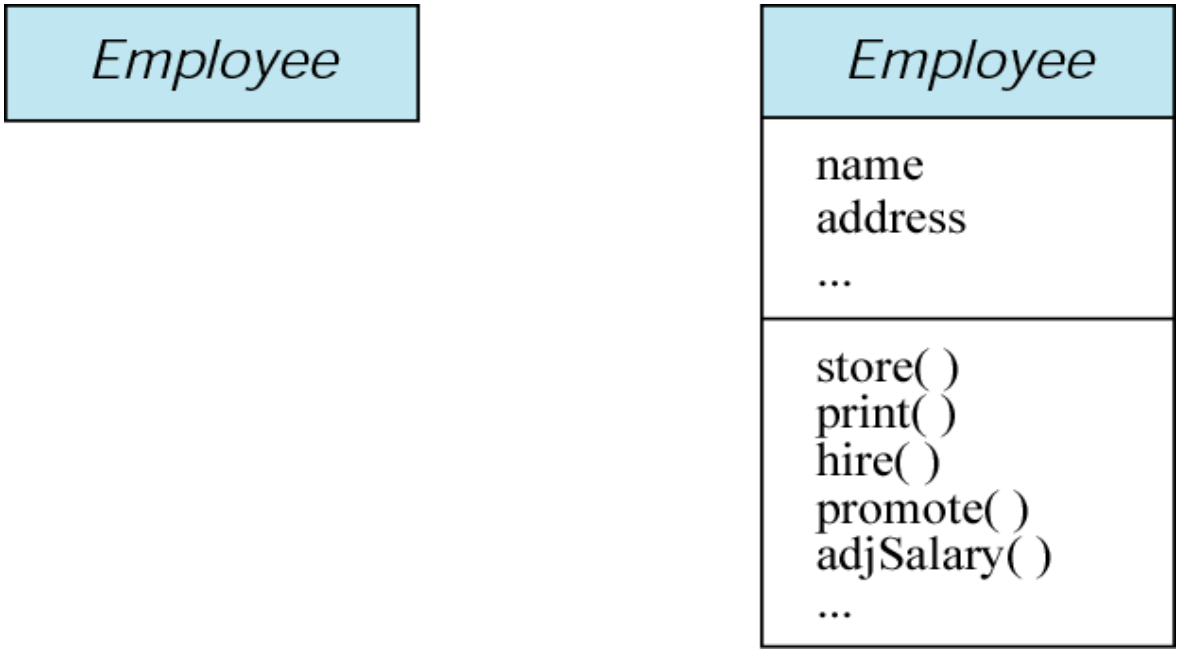


Figure 10-19 Class template diagrams

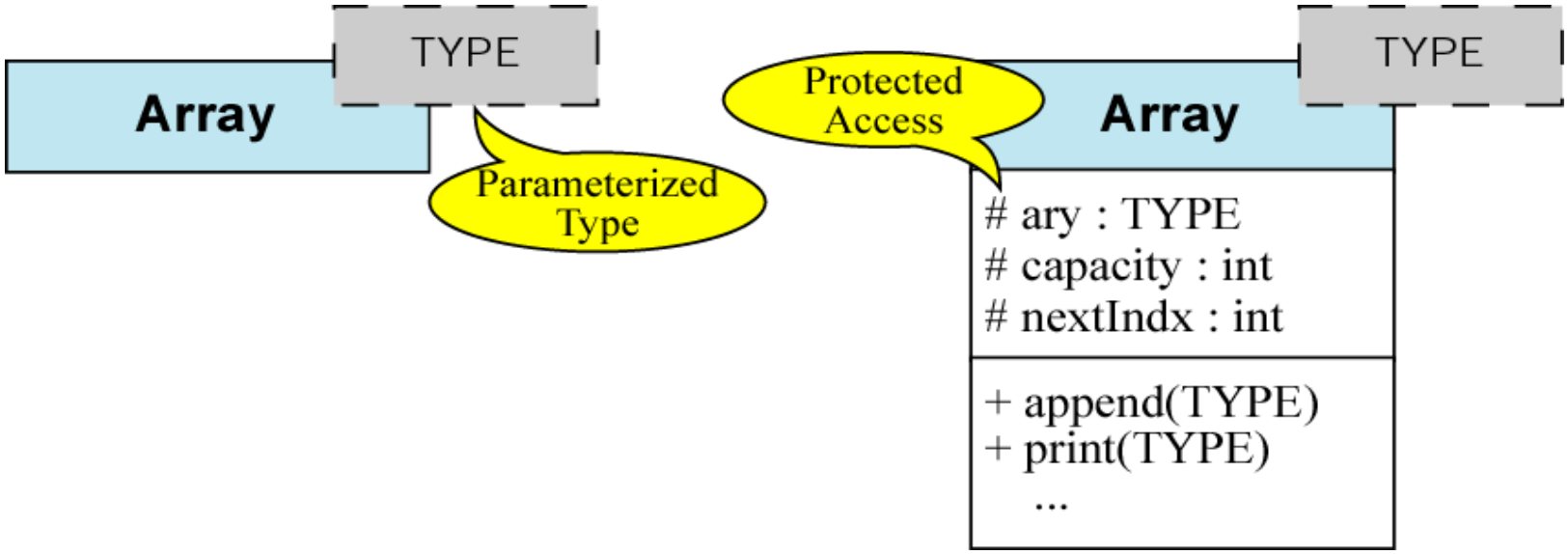


Figure 10-20 Class association

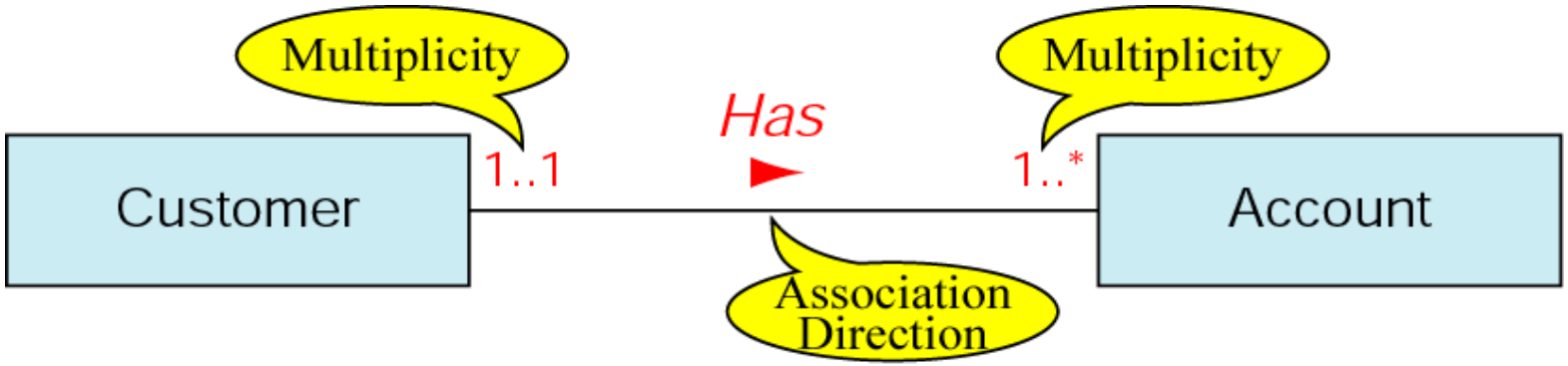


Figure 10-21 Aggregation

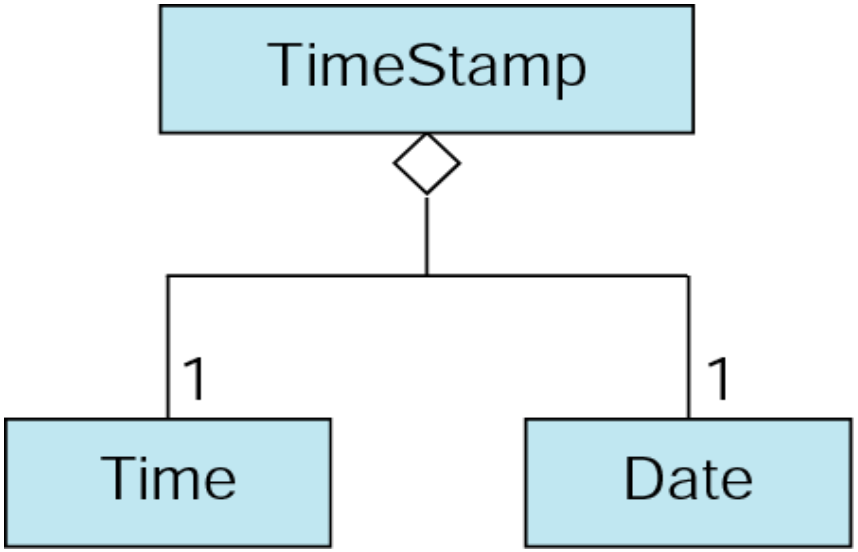


Figure 10-22 Composition

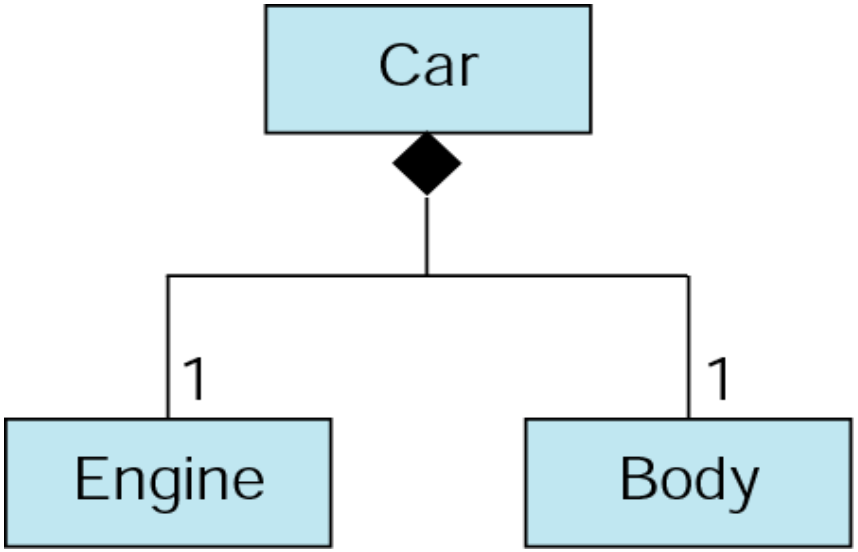


Figure 10-23 Generalization diagram

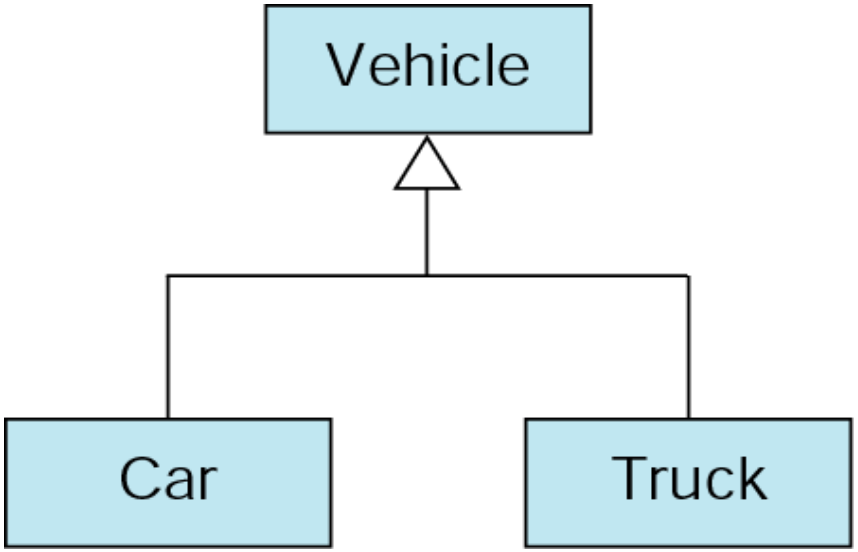


Figure 10-24 Object diagram