

Chapter 2

Fundamental Concepts



Chapter Objectives

- At the end of this chapter you will be able to define and use the following concepts:
 - Object
 - Class
 - Attribute
 - Method
 - Message
 - Association
 - Aggregation
 - Inheritance
 - Polymorphism

Object



Define

- An **object** is a specific entity or concept that has meaning in an application domain.
- Examples are:

Object
Rayanne
Software Architects
Empire State Building
Visa Credit Card 4763 1172 4138 4263
This overhead projector
My 1985 Volkswagen Bus

Object

- In an object oriented system, each object is identified, referenced, and manipulated individually.
- Each object *has*:
 - A unique identifier
 - One or more attributes
 - One or more behaviors
 - Relationships with other objects
- Another view of an object is a chunk of memory that holds the values of attributes and the code that implements behaviors.

Class



Define

- A **class** is the definition of a set of potential objects that have the same attributes, behaviors, and relationships.
- Examples are:

Class	Object
Person	Rayanne
Company	Software Architects
Building	Empire State Building
Visa Credit Card	Visa Credit Card 4763 1172 4138 4263
Overhead Projector	This overhead projector
Vehicle	My 1985 Volkswagen Bus

Class

- Each class *defines*:
 - One or more attributes
 - One or more behaviors
 - Relationships with other classes

- A class is a factory for making objects.



- Each object is an **instance** of (or an **instantiation** of a class).

Exercise 2.1



Exercise

- Identify these as either a class or an object.

Class

Object

State capital

Los Angeles

City

North Carolina

City Hall

Earth

Moon

Milky Way

Continent

Attribute



Define

- An **attribute** is a data value contained within an object that has meaning in an application domain.
- Attributes have values called **attribute values**.
(Clever, eh?)
- Attributes and their values are typically:
 - Basic data types (Integers, Floats, Booleans, Character Strings)
 - Other objects

Behavior



Define

- A **behavior** is a service provided by an object.
- A **method** is the implementation of a behavior in a particular object oriented programming language (C++, Java, Smalltalk, ...)
- Examples:

Class	Method
Circle	draw()
Television	disableSoundDuringCommercials()
Automobile	start()

Method

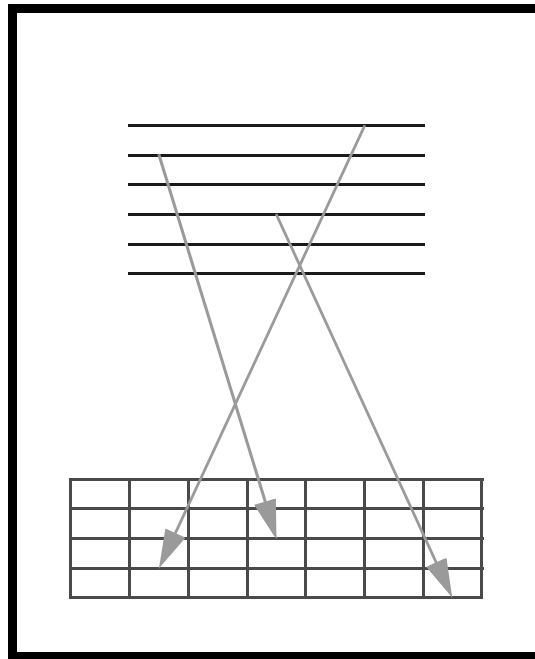


Define

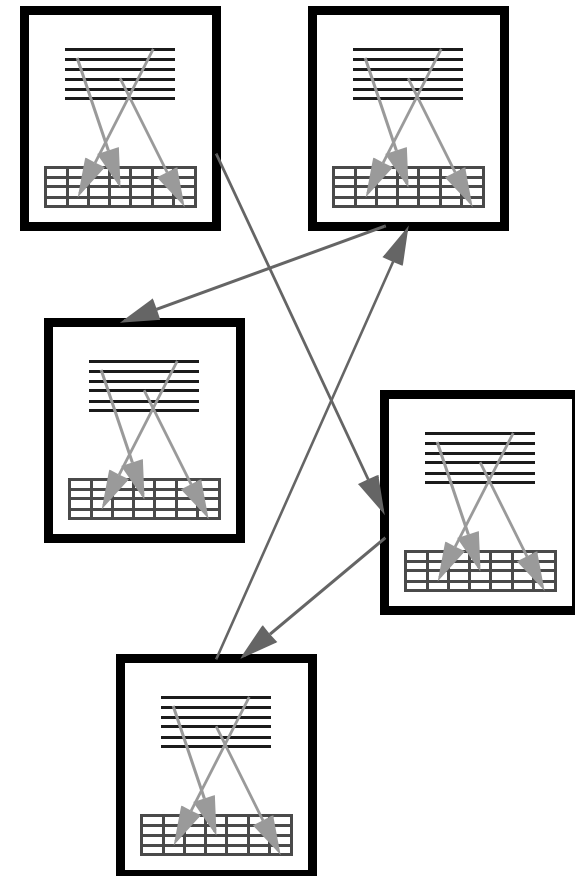
- Parameters may be passed to a method. The **signature** of a method is defined as its name and the number and types of the input parameters (if any).
- Methods are often divided into two types:
 - Accessor methods that return the value of an attribute.
 - Modifier methods that command an object to do something interesting (perform a computation, change its state, send messages to other objects, ...).

Under The Covers

Procedural



Object Oriented



Classes



Document

ClassName

-privateAttribute
-privateAttribute:type
-privateAttribute:type=defaultValue

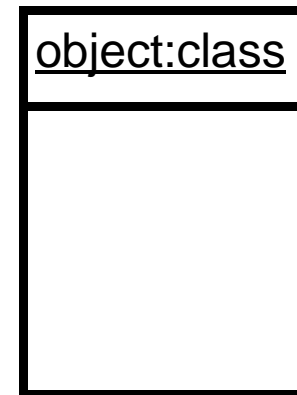
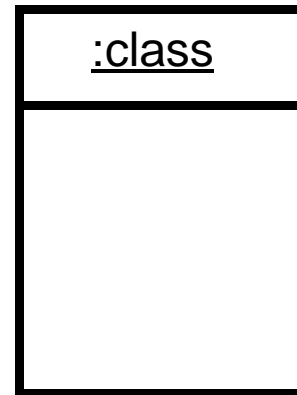
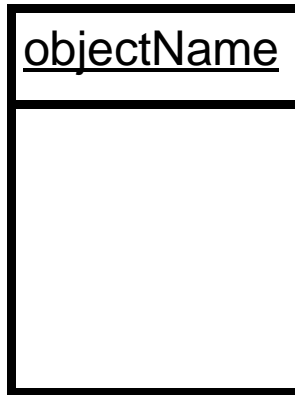
+publicMethod()
+publicMethod(...)
+publicMethod(...):returnType

-privateMethod(...):returnType

Object Notation



Document



Class Implementation



Develop

C++

```
class Widget {  
    private:  
        string description;  
        string number;  
        int reorderPoint;  
  
    public:  
        reorder();  
        string getDescription();  
  
        int getQuantityOnHand();  
  
};
```

Java

```
public class Widget {  
  
    private String description;  
    private String number;  
    private int reorderPoint;  
  
    public reorder();  
    public String  
        getDescription();  
    public int  
        getQuantityOnHand();  
  
}
```

Abstract Class



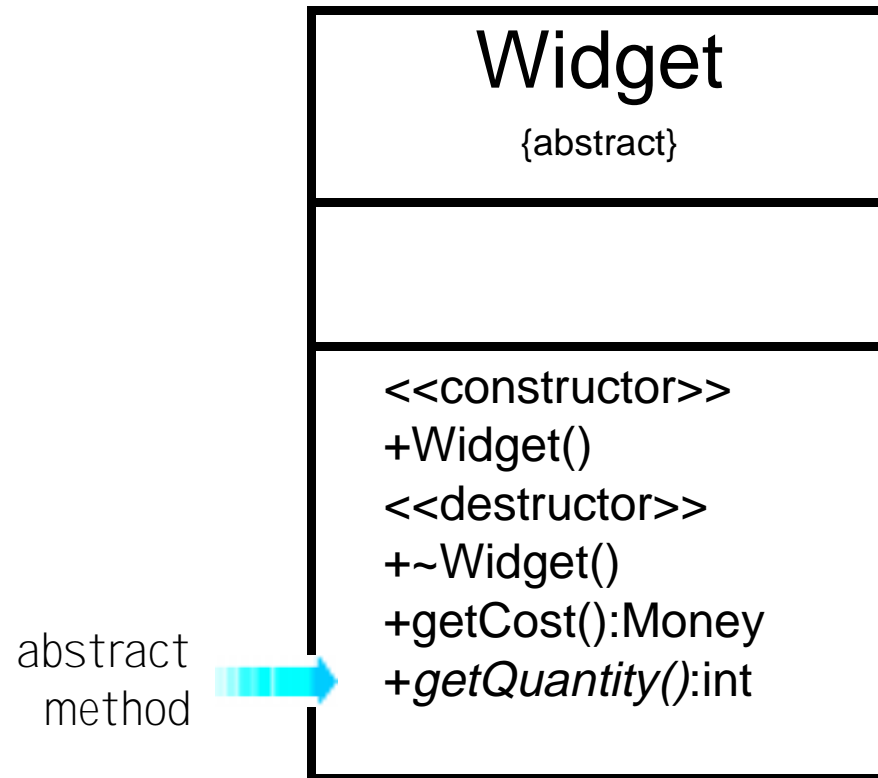
Define

- An **abstract class** represents an important concept in the application domain at an abstract (high) level. As such, it specifies an interface (protocol) that other classes will adhere to.
- An abstract class is a class from which no objects can be created.
- An abstract class is a class that contains at least one **abstract method**, i.e., a method for which no implementation can be written at this level of abstraction.

Abstract Class



Document



Concrete Class



Define

- A **concrete class** is a class in which every method has an implementation, either defined in this class or obtained from another.

Exercise 2.2



Exercise

- Define the protocol for an alarm clock.

