Classes and Objects

- Java programs are written as collections of classes, which serve as templates for individual objects. Each object is an instance of a particular class, which can serve as a pattern for many different objects.

- Classes in Java form hierarchies. Except for the class named Object that stands at the top of the hierarchy, every class in Java is a subclass of some other class, which is called its superclass. A class can have many subclasses, but each class has only one superclass.

- A class represents a specialization of its superclass. If you create an object that is an instance of a class, that object is also an instance of all other classes in the hierarchy above it in the superclass chain.

- When you define a new class in Java, that class automatically inherits the behavior of its superclass.

Biological Models of Class Structure

The structure of Java’s class hierarchy resembles the biological classification scheme introduced by Scandinavian botanist Carl Linnaeus in the 18th century. Linnaeus’s contribution was to recognize that organisms fit into a hierarchical classification scheme in which the placement of individual species reflects anatomical similarities.

Carl Linnaeus (1707–1778)
The **Fruit** Hierarchy

The classes that represent fruit objects form a hierarchy, part of which looks like this:

```
Fruit
  Apple
  Banana
  Orange
  PineApple
```

The **Fruit** class represents the collection of all fruit objects. The four subclasses shown in this diagram correspond to particular types of objects: apples, bananas, oranges, pine apples. The class diagram makes it clear that any **Apple**, **Banana**, **Orange**, or **PineApple** is also a **Fruit**.
The **Animal** Hierarchy

The classes that represent animal objects form a hierarchy, part of which looks like this:

```
Animal
  /   \
Cat   Dog
     /   \n    Crow Horse
```

The **Animal** class represents all animal objects. The four subclasses shown in this diagram correspond to particular types of objects: cats, dogs, crows, horses. The class diagram makes it clear that any **Cat**, **Dog**, **Crow**, or **Horse** is also an **Animal**. But the inverse is NOT true. That is, any **Animal** is not a **Cat**; any **Animal** is NOT a **Dog**, etc.

Inheritance

Inheritance relation in Java is specified using the `extends` keyword.

```
public class Cat extends Animal { ... 

public class Dog extends Animal { ... 
```
Inheritance

If no superclass is defined, by default, the class will inherit from the `Object` class.

```java
public class Animal { ... }
```

is the same as

```java
public class Animal extends Object { ... }
```

That is, the hierarchy really is:

```
Object
   \_______________ Animal
        \       /   \   /
         \     /     \  |
          \   /       \|
           \ /        /|
            \|      / |
             \|    /  |
              \| /   |
               \|
                Cat Dog Crow Horse
```

Inheritance

A `subclass` may redefine a method that is defined by a `superclass`. In this case, it is said that the subclass `overrides` the method.