

24. Inheritance :: The *Object* Class

The Java Class Library contains a class named *Object* (which is in the *java.lang* package) that is the superclass for all classes. For example, when you create a new class:

```
public class MyNewClass {  
    ...  
}
```

The compiler modifies your code so *MyNewClass* inherits from *Object*:

```
public class MyNewClass extends java.lang.Object {  
    ...  
}
```

unless your class inherits from some other class, in which case *MyNewClass* will still be a subclass of *Object* via transitivity:

```
public class Super { // Super is a subclass of Object  
    ...  
}  
  
public class MyNewClass extends Super { // MyNewClass is a subclass of Super.  
    ...                               // MyNewClass also a subclass of Object.  
}
```

24. Inheritance :: Super/Sub-Classes and Direct Super/Sub-Classes

We can make a distinction—when necessary—between a **superclass** and a **direct superclass**. *Super* would be the **direct superclass** of *MyNewClass* because *MyNewClass* extends *Super*. *Object* is also a superclass of *MyNewClass*, but it is not the direct superclass.

Likewise, we can make a distinction between a **subclass** and a **direct subclass**:

```
public class Super {  
    ...  
}  
  
public class Sub extends Super {    // Sub is a direct subclass of Super.  
    ...  
}  
  
public class SubSub extends Sub {    // SubSub is a direct subclass of Sub.  
    ...                               // SubSub is a subclass of Super.  
}
```

Since *Object* is either the direct superclass of all classes or a superclass of all classes, then all classes are either direct subclasses of *Object* (because the class does not extend another class) or subclasses of *Object* (because the class extends another class).