**Chapter 9. Inheritance and Polymorphism**

* Object-oriented programming allows you to derive new classes from existing classes. This is called *inheritance*
* A subclass inherits accessible data fields and methods from its superclass, and may also add new data fields and methods

**9.2. Superclasses and Subclasses**

* In Java terminology, a class C1 extended from another class C2 is called a *subclass*, and C2 is called a *superclass*.
* A superclass is also referred to as a *supertype*, a parent class, or a base class, and a subclass as a *subtype*, a child class, an extended class, or a derived class.
* A subclass inherits accessible data fields and methods from its superclass, and may also add new data fields and methods.
* Consider geometric objects. Suppose you want to design the classes to model geometric objects like circles and rectangles.
* Geometric objects have many common properties and behaviors. They can be drawn in
	+ a certain color,
	+ filled or unfilled.
* Thus a general class GeometricObject can be used to model all geometric objects.
* This class contains the properties
	+ color and
	+ filled and their appropriate
	+ get and
	+ set methods. Assume that this class also contains the
	+ dateCreated property and the
	+ getDateCreated() and
	+ toString() methods. The toString() method returns a string representation for the object.
* Since a circle is a special type of geometric object, it shares common properties and methods with other geometric objects.
* Thus it makes sense to define the Circle class that extends the GeometricObject class. Likewise, Rectangle can also be declared as a subclass of GeometricObject.
* Figure 9.1 shows the relationship among these classes. An arrow pointing to the superclass is used to denote the inheritance relationship between the two classes involved
* 