Abstract classes and interfaces

* When we extended a class into a child class, we could override the parent’s methods to provide custom methods for the child
	+ BankAccount:
		- deposit()
		- withdraw()
	+ under polymorphism, if I treat a CheckingAccount object as a BankAccount, it still uses the overridden versions of those methods
* What if we require the implementer to extend our class and override certain methods, which we don’t want to provide a default implementation for?
* Call the parent an abstract class – any method that you want the child to override can be called an abstract method
* You can’t create an object of abstract class type

Abstract Declarations

* put “abstract” in the class declaration

public abstract class MyClass …

* abstract methods are declared with “abstract” and terminated with semicolon – no implementation

public abstract void myMethod();

* When you extend an abstract class, you must:
	+ Override ALL the abstract methods; ot
	+ Make the child itself abstract (in which case the child inherits the abstract method)

Suppose you have an abstract class with:

* no fields
* no constructors
* only abstract methods

These are called interfaces in Java.