# TEACHER’s guide

# Inheritance, Polymorphism, and Abstract Classes

**OBJECTIVES:** The student will understand inheritance in Java.

The student will understand abstract methods and classes. The student will learn when a class is concrete.

The student will learn the meaning of polymorphism, early binding, and late binding.

The student will understand Java interfaces and how they provide some of the features of multiple inheritance.

**ACTIVITIES/TIME:** One and a half Weeks

**MATERIALS:** Student Lesson A20: *Inheritance, Polymorphism, and Abstract Classes*

Lab Assignment A20.1, *OldMacDonald*

Teacher’s Guide A20: *Inheritance, Polymorphism, and Abstract Classes*

Lab Assignment A20.1, Answers, *Animal.java, Chick.java, Cow.java, Farm.java, NamedCow.java, Pig.java*

**REFERENCES:** **Java 101: Inheritance**  
 <http://www.med.harvard.edu/JPNM/Java/Java101/Inheritance.html>

This site provides a simple example of inheritance.

**INSTRUCTOR**

**NOTES:** Several terms are used when discussing polymorphism. *Overloading* is used in the text of the lesson and some students may confuse overloading with overriding. Overloading a method means creating a new method with the same name but a different signature, i.e. different parameters. This is the same thing as early binding. *Overriding a method* means providing a different implementation for a method in a subclass without changing the signature. This permits late binding, i.e. choosing the particular implementation of the method to execute depending on the type of the object. Late binding is sometimes referred to as *dynamic binding*.

The concepts of abstract classes and interfaces are rather advanced topics. They are, however, extremely important when creating designs of complex systems. If a designer can do a good job of designing what the interface to a class looks like without specifying all of the implementation details, then it is likely the system can be extended more easily.

Inheritance is often referred to as an “is-a” relation. A subclass “is a” kind of superclass. A Pig “is-a” Animal, a Chick “is-a” Animal, a Cow “is-a” Animal.

Some students think that the arrows in a class hierarchy diagram should point from the superclass to the subclass to indicate inheritance. A good way to think of the direction is that the subclass can see the superclass and is dependent on it. If Fish is a subclass of SeaCreature, then Fish can see SeaCreature and is dependent on SeaCreature, but SeaCreature has no knowledge of or dependency on Fish.