# Teacher’s guide

# Inheritance

­­­­­­

**OBJECTIVES:** The student will learn about single inheritance.

The student will learn how to use inheritance to build class hierarchies.

The student will use inheritance and subclasses in a program.

The student will learn how to used method overriding to modify the behavior of a subclass. The student will learn how to use **super** in an overridden method. The student will learn how to use the access modifier **protected**.

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

**MATERIALS:** Student Lesson A11: *Inheritance*

Worksheet A11.1, *Inheritance Review*

Lab Assignment A11.1, *BackToSchool*

*BackToSchool* Starter Code: *Person.java, Student.java*

Lab Assignment A11.2, *GraphicPolygon*

Teacher's Guide, Lesson A11: *Inheritance*

Worksheet A11.1, *Answer Sheet*

Lab Assignment A11.1 - Answers: *BackToSchool.java, Person.java, Student.java, CollegeStudent.java, Teacher.java*

Lab Assignment A11.2, Answers: *GraphicPolygon.java, RegularPolygon.java*

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

A simple example of inheritance.

**INSTRUCTOR**

**NOTES:** Inheritance is one of the three fundamental principles of Object-Oriented Programming (the other two are encapsulation and polymorphism). Inheritance is a very powerful technique for reuse of existing code. Once behavior and attributes are defined for a superclass, a programmer can easily reuse them by defining a subclass.

Inheritance is sometimes called an “is a” relationship. If Vertebrate is a superclass of Bird, then you can say that a Bird “is a” Vertebrate. Saying this out loud can help when trying to define good inheritance hierarchies. For example, one might be tempted to derive a Nose class from a Face class. However, the statement “a Nose is a Face” does not make sense. In this case, the statement, “a Face has a Nose” makes more sense. This is another kind of relationship, often called a “has a” relationship. In this case, Nose would actually be an attribute of Face rather than a subclass.

Lab Assignment A11.2, *GraphicPolygon*, asks the students to extend the RegularPolygon class to produce pictures of polygons. This is a very fun lab for students.

**WORKSHEET**

**NOTES:** Inheritance is a very important part of Java. Worksheet A11.1, *Inheritance Review* gives students a simple illustration of how Inheritance can be used. In practice, programmers have wide access to many classes - all through Inheritance! This would also be a great chance to practice using Javadoc. Javadoc creates *.html* files that show the classes in tree format – like Sun’s website for Java classes.

The parameter lists for some of the constructors are quite long - please explain that they were used to give a clear example of Inheritance, rather than as an example of good programming.