

Weekly Lab 3 – Inheritance

Maximum Points = 10

The purpose of this lab is to review your study of computer programming and algorithms from CS 1 focusing on classes and ArrayList.

Exploring Inheritance

1) File Dog.java contains a declaration for a Dog class. Save this file to your directory and study it—notice what instance variables and methods are provided. Files Labrador.java and Yorkshire.java contain declarations for classes that extend Dog. Save and study these files as well.

1. Compile the three classes; you should get an error saying something like Cannot find symbol - constructor Dog()

If you look at line 16 of Labrador.java it's just a {, and the constructor the compiler can't find (Dog()) isn't called anywhere in this file.

a. What's going on? (Hint: What call must be made in the constructor of a subclass?)

=>

b. Fix the problem (which really is in Labrador)

c. Recompile Labrador.java; you should get an error saying something like non-static variable breedWeight cannot be referenced from a static context

d. Fix the problem

2) File DogTest.java contains a simple driver program that creates a dog and makes it speak. Study DogTest.java, save it to your directory, and compile and run it to see what it does. Now modify these files as follows:

1. Add statements in DogTest.java after you create and print the dog to create and print a Yorkshire and a Labrador. Note that the Labrador constructor takes two parameters: the name and color of the labrador, both strings. Don't change any files besides DogTest.java. Now recompile DogTest.java

a. Fix any problems so that DogTest.java creates and makes the Dog, Labrador, and Yorkshire all speak.

2. Add code to DogTest.java to print the average breed weight for both your Labrador and your Yorkshire. Use the avgBreedWeight() method for both.

a. What error do you get? Why?

=>

b. Fix the problem by adding the needed code to the Yorkshire class.

3. Add an abstract int avgBreedWeight() method to the Dog class. Remember that this means that the word abstract appears in the method header after public, and that the method does not have a body (just a semicolon after the parameter list). It makes sense for this to be abstract, since Dog has no idea what breed it is. Now any subclass of Dog must have an avgBreedWeight method; since both Yorkshire and Labrador do, you should be all set.

a. Save these changes and recompile DogTest.java. You should get an error in Dog.java (unless you made more changes than described above). Figure out what's wrong and fix this error, then recompile DogTest.java.

b. You should get another error, this time in DogTest.java. Read the error message carefully; it tells you exactly what the problem is. Fix this by changing DogTest (which will mean taking some things out).

3) Compile and run your program.

(Due before 11:59pm on Friday, September 3, 2010) Submit your .java files containing your program to the dropbox in WebCT.

Grades are determined using the following scale:

- Runs correctly.....: ___/3
- Correct output.....: ___/2
- Design of output.....: ___/1
- Design of logic.....: ___/2
- Standards.....: ___/1
- Documentation.....: ___/1

[Grading Rubric](#) ([Word document](#))