

ZITE1102 – Programming Fundamentals

Practical 5

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.

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*; you should get an error saying something like

```
./Labrador.java:18: Dog(java.lang.String) in Dog cannot be
applied to ()
{
^
```

1 error

If you look at line 18 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*) 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. What error do you get? Why?

=>

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.

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*. 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).