

# Weekly Lab 3 – Interfaces and Polymorphism

Maximum Points = 10

The purpose of this lab is to introduce the concept of interfaces and polymorphism.

1) It is often the case that two or more classes share a common set of methods. For programming purposes we might wish to treat the objects of those classes in a similar way by invoking some of their common routines.

For example, the Dog and Cat classes listed below agree on the void method `speak`. Because Dog and Cat objects have the ability to “speak,” it is natural to think of putting both types of objects in an `ArrayList` and invoking `speak` on every object in the list. Is this possible? Certainly we could create an `ArrayList` of Dog that would hold all the Dog objects, but can we then add a Cat object to an `ArrayList` of Dog?

a) Try running the main program below as it is written.

b) Run it a second time after uncommenting the line that instantiates a Cat object and tries to add it to the `ArrayList`.

```
import java.util.*;

public class AnimalRunner
{
    public static void main(String[] args)
    {
        ArrayList<Dog> dogcatList = new ArrayList<Dog>();
        dogcatList.add(new Dog("Fred"));
        // dogcatList.add(new Cat("Wanda"));
        for (Object obj : dogcatList)
        {
            // obj.speak();
        }
    }
}
```

2) Our experiment to add Cat objects to an `ArrayList` of Dog objects failed. Perhaps we should try using the original Java `ArrayList` without generics (e.g. `<Dog>`)?

a) Try running the code after removing the generic references `<Dog>` along with the Dog and Cat classes defined above.

b) Run it a second time after uncommenting the line that invokes `speak`.

3) The experiment shows that we are now able to add Dog and Cat objects to the `ArrayList`, but there is a compile error on the line `obj.speak` because `obj` is an `Object` reference variable and the class `Object` doesn't contain a `speak` method. We need a reference variable that can refer to Dog and Cat objects and which also allows us to invoke `speak`. The solution to the problem uses interfaces.

a) First create an interface called `Speakable` that contains a `void speak()` method signature.

Be sure to modify the Dog and Cat classes to indicate that they implement the `Speakable` interface. (For example, in the case of the Dog class, we will code `public class Dog implements Speakable`. Be sure to make a similar change in the declaration of the Cat class.)

- b) The term `Speakable` can be used to create `Speakable` references.
- i) Using generics, create an `ArrayList` of `Speakable` objects in the `main` method.
  - ii) Modify the `for` loop so that it iterates over `Speakable` objects.
  - iii) Try adding the `Dog` and `Cat` objects and invoking the `speak` method on each object. Does this work?

4) Compile and run your program.

(Due before 11:59pm on Friday, January 28, 2011) Submit ALL 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](#))