• What will be the output when running the above code?

Meow. I am a cat. My name is Kurre.

Woof. I am a dog. My name is Vilma.

Meow. I am a cat. My name is Bamse.

• What is meant by polymorphism?

Polymorphism is a form that it occurs when we have many classes that are related to each other by inheritance.

• How does polymorphism work in the above program?

The program will stored the Cat and Dog, not in an Animal array, which is superclass of Cat and Dog.

• The method introduceYourself of Animal appears to be never called? Why not?

• Comment out the method introduceYourself in Dog. What happens now when you run the program?

Meow. I am a cat. My name is Kurre.

Morr. I am a cat.

Meow. I am a cat. My name is Bamse.

• Where is the name stored for the instances of Cat and Dog? (In what / which classes did you put the instance variable that refers to the name of the animal? Both Cat and Dog, or just in Animal?)

Name	Classes
Kurre	Animal.Cat(String newName)
Vilma	Animal.Dog(String newName)
Bamse	Animal.Cat(String newName)

• How does the code in the test program work?

At the beginning, program declare an array of Animal and int i. And it initialize the array with size of 3 and . The object Cat and Dog will be stored in Animal Array

which Cat in index 0 and 2, Dog in index 1. Then, program use variable i which initialized with 0 in while loop. In while loop, it call introduceYourself() from each object stored inside the array and then the variable i will increase by 1. While loop will stop until the i is more than length of Animal array.

• How does an array work?

An array is a container object that holds a fixed number of values of a single type. Its memory is based on the data type and size. For example, an array declared with data type int and size of 4. It can be stored int value in any array index.

• In the above programs we have used a while loop to step through the array and to get information about the animals. But there is a more appropriate loop statement here. What is it?

```
for ( int i = 0; i < allAnimals.length; i++){
        allAnimals[i].introduceYourself();
}
```