Questions:  
• 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 the act of making an object/class to behave like other object/class.

• How does polymorphism work in the above program?

In the program, we stored the Cat and Dog instances in an array of Animal, which is their parent class. This is called upcasting since we're casting the child classes into their parent class.

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

introduceYourself() method from the Animal class doesn't run because its child classes, Cat and Dog class have overriden the method with their own implementation of the same method. When called, we'll use the child version of the method since we called the method using the subclass instances (which was upcasted into Animal).

• 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.

The Animal version of the method will be called instead.

• 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?)

Inside of the Animal class as a protected String, so both Cat and Dog subclasses will inherit it.

• How does the code in the test program work?

First, declare an Animal array and an int variable, which acts as a counter later.

Then initialize the array into an Animal array with the size of 3.

Then initialize Cat and Dog objects and store them inside the Animal array. Cat on index 0 and 2, while Dog on index 1.

Initialize the int variable earlier with 0 and start a while loop, which keeps repeating until the counter is no longer less than the length of the array.

Inside the loop, call the introduceYourself() from each object stored inside the array.

Then increment the counter by 1.

• How does an array work?

Arrays work by reserving some sections of consecutive memory based on the type and size of the array. For example, an char array with the size of 8, will cause approximately 8 bytes of memory to be reserved for the array. We can then store the any values we want inside each 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?

By using a for loop, we can reduce the length of the code and avoid using additional variables.

for (int i = 0; i != allAnimals.length; i++) {

allAnimals[i].introduceYourself();

}