

Q1 • 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.

```
PS D:\Users\Auntie Goh\Desktop\Quiz> & 'c:\Users\Auntie Goh\.vscode\extensions\vsc
\jdk-12.0.2\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\Auntie Goh\AppData
at.java\jdt_ws\Quiz_7586e21\bin' 'Lab1Stage1'
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.
PS D:\Users\Auntie Goh\Desktop\Quiz> []
```

Q2 • What is meant by polymorphism?

Polymorphism is the concept of a function or an object having more than one form or implementation. The main use of **polymorphism** in Object Oriented **Programming** is the ability of objects of different types to respond to the same method call, implementing a specific type of behaviour.

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Inheritance lets us inherit attributes and methods from another class. **Polymorphism** uses those methods to perform different tasks. This allows us to perform a single action in different ways.

Q3 • How does polymorphism work in the above program?

For example, think of a superclass called Animal that has a method called introduceYourself (). Subclasses of Animals could be Cat, Dog - And they also have their own implementation of an animal introduceYourself (the cat meow and name Kurre and Bamse, the dog woof and name Vilma, the animal morr).

Class Cat and Dog have inheritance from animals. Each class have 2 methods which is constructor and introduceYourself (). Parent class is Animal while the child class are Cat and Dog. We stored the Cat and Dog instances in array of Animal. We have upcasting which mean casting the child classes into their Parent class.

Q4 • The method introduceYourself of Animal appears to be never called? Why not?

Because of, introduceYourself() method from the Animal class doesn't run because its child classes, Cat and Dog class have overridden 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).

Q5 • Comment out the method introduceYourself in Dog. What happens now when you run the program?

the output:

Meow. I am a cat. My name is Kurre

Morr. I am an animal.

Meow. I am a cat. My name is Bamse

The output will change.

The second line of output will change from "Woof. I am a dog. My name is Vilma." to "Morr. I am an animal.". This is because there is no method introduceYourself in Dog to override the method introduceYourself of Animal. So, the method introduceYourself of Animal is called.

Q6 • 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.

Q7 • How does the code in the test program work?

The superclass is Animal and subclass is Cat, Dog. The subclass inherits from the superclass. The method introduceYourself of subclass override the method introduceYourself in superclass. Constructors in created in subclass to initialize object Cat and Dog. Animals(Cat/Dog) with name initialized are stored in array. The data stored in array printed out using loop by calling the method introduceYourself.

First it made an array called allAnimals and set the array size to 3. Then fill the array with calling dog and cat classes with their name in the constructor. Initialize integer i to zero while i is less than the length of allAnimals array, it will call the method introduceYourself() in each animal child class in array. then Add +1 to i. Repeat until value of i is greater than the length of the allAnimals array. then stop.

Q8 • How does an array work?

Array used to stored elements with the data type declared same as the array.
Maximum number of elements to stored in array depends on the size of array.

The method introduceYourself of Animal appears but never called because at the main class the array is only 3 index. For example,

```
allAnimals[0] = new Cat("Kurre");
```

```
allAnimals[1] = new Dog("Vilma");
```

```
allAnimals[2] = new Cat("Bamse");
```

the code is only consists of `allAnimals[i].introduceYourself();`

```
    i = i + 1;
```

that's mean by looping while, start from for $i=0$ or index 0 which call the Cat class and insert name as Kurre, then $i+1 = 0+1=1$.

$i=1$ or index 1 which call the Dog class and insert name as Vilma, then $i+1 = 1+1=2$.

$i=2$ or index 2 which call the Cat class and insert name as Bamse, then $i+1 = 2+1=3$.

Then, $1 < \text{allAnimals.length}$ which is 3 so the while loop end

Q9 • 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 loop.

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