

## 1. Explain the programming case in that video!

In this programming case, there are 5 classes, Utama, Hewan, Anjing, Burung and Kucing. The main class here would be the class Utama which will contain our public static void main to run our program. Next there is a class called Hewan which will be a parent class or super class to 3 other classes which are Anjing, Burung and Kucing. The class Hewan then becomes abstract by adding the keyword abstract in front of public class Hewan so this class can not be instantiated. In the class Hewan, there are 4 private attributes which are umur that is an integer, then nama that is a string, then mata and also mulut that are both integers. It also has setter getters for the private attributes of umur, nama and mulut. In the class Hewan and its sub-classes, they all also have constructors that are empty. Then in the class Hewan, there is also then a method that is made abstract with the name suara(); where then all of the sub-classes of Hewan must override this method. In the class Anjing, the method suara() is overridden and made to print "anjing menggongong", while in the class Burung, it is overridden and made to print "burung berkicau" and finally in the class Kucing, it is overridden and made to print "Kucing mengeong". Then there is also a method that is made to print "Makan" in the class Hewan, where then it is overridden in the class Kucing to print "Kucing makan" to show that we can still override methods that are not abstract. Inside the class Kucing, there is also a method that is called suara() however has a parameter of (String s) to create an overloading method from the method suara() with no parameters. However, since the abstract class does not recognize this method, it is unable to be used. Finally, there is also 2 interfaces that are made where the class Hewan will implement these 2 interfaces. The 2 interfaces created are Trick and Komunikasi. In the interface Trick, it will hold a constant attribute which is a variable named bintang with the type being an integer and the variable will contain the number 5. Then it also has a method called duduk() where then the class that implements the interface Trick will be able to use this function. However, since the class Hewan contains this method that is implemented from an interface, this method must be overridden in all of the sub classes of Hewan. In particular, the method is overridden in the class Anjing to print "anjing duduk" when called. Finally the other interface called Komunikasi will contain nothing. This is just to show that a class can implement more than one interface.

The interface is implemented on the abstract class Hewan so that it can be automatically be inherited by all of the subclass of Hewan. There is also the use of an abstract class to achieve polymorphism in the program to increase efficiency of the program as well as code re-usability. It achieves polymorphism as an object in the main class Utama may be re-instantiated to another sub class of Hewan where Hewan itself can not be instantiated.

## 2. Explain the polymorphism concept in that program!

The polymorphism concept in that program is that there are 5 classes. The first class is the main class called Utama, then a class named Hewan where it is then turned into an abstract class, so that we cannot make an instance of Hewan, with 3 sub classes that extends Hewan which are Anjing, Burung and Kucing. In the Hewan class, there is also an abstract method called suara() where then all the sub classes must override this to be able to use it. In the abstract class Hewan, there is an abstract method that is initialized where then the sub classes will have to override this method as it is an abstract method, the abstract method is called suara(). In this program, the method will print "anjing menggongong" if an object is an instance of the class Anjing, or will print "burung berkicau" if an object is an instance of the class Burung or will print "Kucing mengeong" if an object is an instance of the class Kucing. The method makan() in Hewan class is also then overridden in each subclass although it does not have the keyword abstract, it is still able to be overridden. In the class Hewan, the method makan() will print "Makan" where the in the subclass named Kucing, it will then print "Kucing makan".

From this, when we create an object in the class Utama, we can create one by writing as follows, `Hewan h = new Anjing();` or `Hewan h = new Burung();` or `Hewan h = new Kucing();` and by this, we can change the instance by simply writing `h = new Kucing();` if `h` was an instance of `Anjing`. From this we can achieve polymorphism where an object can use methods of the same name however from different sub classes of `Hewan`. Then there is also an interface that is created and named `Trick`, which has an attribute named `bintang` with integer type where it holds a value of 5. It is also has a method called `duduk();` to be implemented on a class, in this case to the abstract class `Hewan` where it then gets overridden by its sub classes. In the sub class `Anjing`, this method is overridden and functions to print "anjing duduk".

### 3. Why we need abstract class, interface & implements?

We would need abstract class to help in defining sub class features efficiently as we do not have to rewrite repeated codes, from this we can also say that abstract class helps in code re-usability with the help of generalization of the methods when a sub class extends a parent class where the subclass will inherit all attributes and methods from the parent class. Aside from that, it also allows us to be able to group certain classes together as siblings and allow the code to be more organized and understandable especially when we need a concrete behaviour of the super-class. However, if we do not need a concrete behaviour of the super-class, we could use interfaces, especially when in Java, it does not support multiple inheritance but it does support multiple interfaces so this would be useful if we need more than one behaviour that a class must share with its sub classes and achieve multiple inheritance efficiently. As for implements, we need this to implement an interface to an existing class so that the class may access the methods contained in it mainly used to declare a class with only abstract methods. Different from extends as extends would allow a class to inherit attributes and methods that the super class owns, this would implement an interface to a class.