Q1: What will be printed?

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

and I am 6 years old.

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

and I am 3 years old.

Q2: Explain how it is that the instance variable age of Animal can be used in Cat and Dog when it is declared in Animal.

Because used inheritance that will inherit all the variable from Animal class to Cat and Dog class. Cat and Dog are subclasses of Animal, they obtain all of the parent classes methods through inheritence.

Q3: What actually consists an instance of?

Variables defined within a class are call instance variable. Animal object Kurre and Vilma contains an instance of Cat and Dog respectively. Each instance of the class contains its own copy of the variables. Which Variables defined within a class are call instance variable.

Q4: And what consists a class of?

For the file of Animal, Cat, Dog and Lab1Stage2 consists of a classes with the same name as the file respectively. A class consists of properties and method definitions. A class is declared by the class keyword. A class is consists of data or variables, methods and codes.

Q5: What is the difference between a class and an instance?

A class is a blueprint of an object or instance. A class defines the properties and methods of an object or instance while the instance is the object of a class created to be used in a program.

Class Variables	**Instance Variables**
Class variables are declared with keyword *static*.	Instance variables are declared without *static* keyword.
Class variables are common to all instances of a class. These variables are shared between the objects of a class.	Instance variables are not shared between the objects of a class. Each instance will have their own copy of instance variables.
As class variables are common to all objects of a class, changes made to these variables through one object will reflect in another.	As each object will have its own copy of instance variables, changes made to these variables through one object will not reflect in another object.
Class variables can be accessed using either class name or object reference.	Instance variables can be accessed only through object reference.

Q6: Change the declaration of the instance variable age of Animal to a class variable using static, in this way:
public static int age;

Refer code

Q7: What is the result of the output now? Why?

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

and I am 3 years old.

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

and I am 3 years old.

It is because static fields are special properties that can be accessed without instantiating a class. Only a single instance of it exists and shared between all objects of the same class. So, when we set the age of Vilma to 3, Kurre's age will also be set to 3 since the age is now a static field that belongs to Animal.

Q8: Where is the value of an instance variable stored?

Instance variable were stored under the class it was defined in.

For example, Kurre and Vilma instance variables were defined in the main method of class Lab1Stage2. Therefore, the variables belong to the scope of main method inside Lab1Stage2 class.

Q9: Where is the value of a class variable stored?

The value of class variable was stored inside a property of the class' objects.

For example, age is a property of Animal. Therefore, the class variable age will be stored in an Animal instance.

Q10: What refers the variable this to?

The keyword ‘this’ refers to the class itself.

When two variables or methods with the same name are used, ‘this’ keyword will make sure the current class version of the two will be used instead.