

Solution Review: Displaying Message Using Inheritance

In this review, solution of the challenge 'Displaying Message Using Inheritance' from the previous lesson is provided.

We'll cover the following

- Solution: Displaying message using inheritance
 - Understanding the Code
- *
- Lines 1 - 19: Creating the Animal class
 - Lines 21 - 36: Creating the Zebra & Dolphin class

Solution: Displaying message using inheritance

```
class Animal {  
  
    private int age;  
    private String name;  
  
    void set_data(int a, String b) {  
  
        age = a;  
        name = b;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}  
  
class Zebra extends Animal {  
  
    String message_zebra(String str) {  
        str = "The zebra named " + getName() + " is " + getAge() + " years old. The zebra comes from " + name;  
        return str;  
    }  
}  
  
class Dolphin extends Animal {
```

```
String message_dolphin(String str) {  
    str = "The dolphin named " + getName() + " is " + getAge() + " years old. The dolphin come  
    return str;  
}  
}
```



Understanding the Code

Lines 1 - 19: Creating the *Animal* class

- On lines 3,4 the member variables are declared that are **private** to the student class. These include an **int** type variable `age` and a second variable is also declared, `name` which is of type **String**.
- From lines 6-10 is the *method* named `set_data`. This according to the requirements, takes in parameters and sets both `age` and `name` to the passed parameters.
- From lines 12-14, is the method `getName()` which returns a **String** type of value.
- Lines 16-18 show the method `getAge()` which returns a **int** type of value.

Lines 21 - 36: Creating the *Zebra* & *Dolphin* class

- The `Zebra` & `Dolphin` class **inherit** from the `Animal` class, each contain only one method named `method_zebra` and `method_dolphin` respectively. A variable `str` is being passed to the methods to store the output **String** according to the respective classes. Getters are used to showcase the inheritance because we are using them without instantiating the `Animal` class.

Let's wrap up this chapter by solving a quiz in the upcoming lesson.