Review of Concepts

1. What is inheritance?

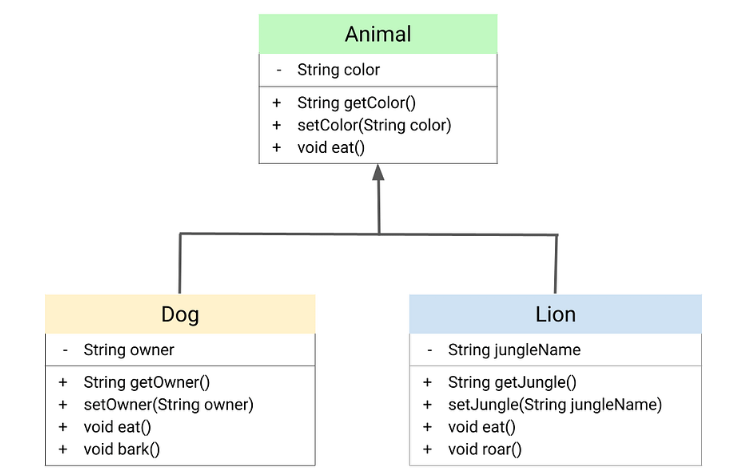
* It is one of the core concepts of OOP.
* It allows developer to create a child/sub/derived class based on a parent/super/base class.
* Main benefit of inheritance is that it allows child class to reuse the attributes and behaviours (methods) available in the parent.

1. Can you think of suitable examples where it can be applied?

A diagram of a diagram of a diagram

Description automatically generated

Another one



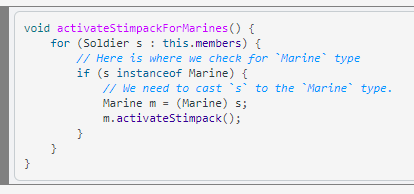
1. What is polymorphism?

“Polymorphism allows us to treat objects of different types as if they were of a common type. This means that we can write code that can work with multiple types of objects through a shared interface or base class, without having to know the specific details of each object. This simplifies our code and makes it more adaptable to changes. “

* Polymorphism is the process of treating objects of different sub types same as long as they have a common super type and they confirm to a uniform interface.
* It allows developer ton write code more maintainable and flexible code
  + Polymorphic code works with different sub types of same inherence hierarchy without any special consideration -Flexibility/Adaptable
  + Polymorphic code works with newly introduced sub types as long as they have same parent class – Maintainable
* To write polymorphic code at a minimum
  + You must inheritance
    - A parent classes
    - 2 or more subclasses

1. What is the purpose of the instanceof keyword?

* instanceof keyword can be used to check for actual/dynamic type of an object
* Example - There are times where we want to determine the actual concrete type of a particular object. If you look at the members attribute of the Squad class, there currently isn't a way for us to know whether a given element is a Marine or a Firebat. To obtain this information, Java provides the instanceof keyword. For example, let's write a method that only applies a Stimpack to Marines within the Squad:



1. Do you understand the code in Soldier.java?