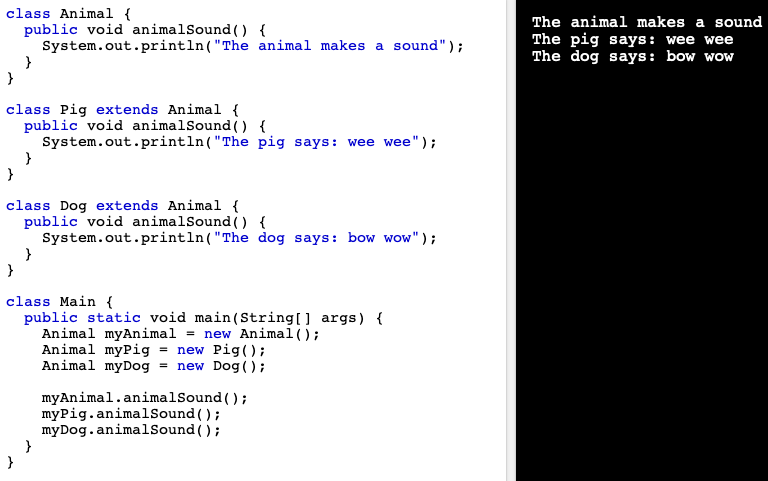
**Polymorphism & Composition Homework - Quiz**

**Polymorphism**

1. What does the ***word*** 'polymorphism' mean? The condition of occurring in several different forms.
2. What does it mean when we apply polymorphism to OO design? Give a simple Java example.

**Polymorphism** uses those methods to perform different tasks. This allows us to perform a single action in different ways.

For example, think of a superclass called Animal that has a method called animalSound(). Subclasses of Animals could be Pigs, Cats, Dogs, Birds - And they also have their own implementation of an animal sound (the pig oinks, and the cat meows, etc.):

1. What can we use to implement polymorphism in Java?

**Upcasting**

If the reference variable of Parent class refers to the object of Child class, it is known as upcasting.

For upcasting, we can use the reference variable of class type or an interface type. For Example:

1. interface I{}
2. class A{}
3. class B extends A implements I{}

1. How many 'forms' can an object take when using polymorphism?

*Polymorphism in****Java****has two types:****Compile****time polymorphism (****static****binding) and Runtime polymorphism (dynamic binding).****Method overloading****is an****example****of****static****polymorphism, while method overriding is an****example****of dynamic polymorphism.*

*NB: an object can only be one form - https://www.polymorphism.co.uk/about/polymorphism*

1. Give an example of when you could use polymorphism. java - a paymentCard is a parent class, the child classes may consist of debit card and credit card, they child classes then share an interface, such as IScan, for the behaviour of scanning the card.

**Composition**

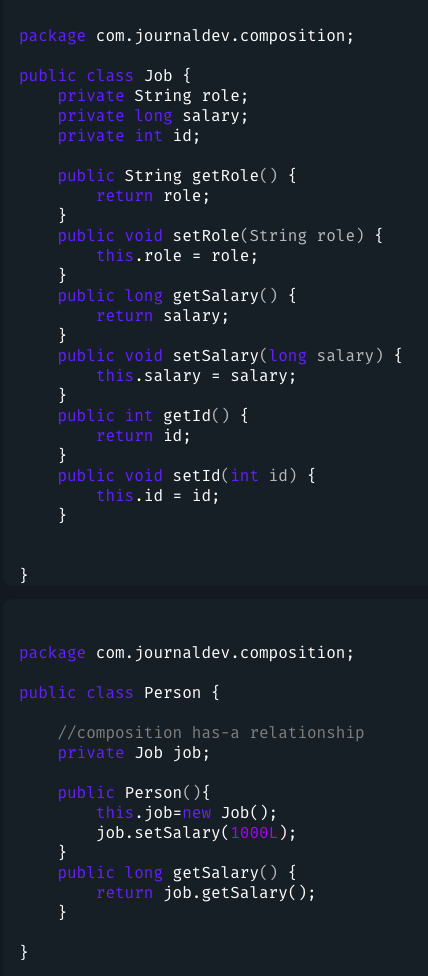
1. What do we mean by 'composition' in reference to object-oriented programming?

It describes a class that references one or more objects of other classes in instance variables. This allows you to model a *has-a* association between objects.

1. When would you use composition? Provide a simple example in Java.

when there is a ‘has-a’ relationship between objects. I.e. a Person object, has-a Job (object).

1. What is/are the advantage(s) of using composition?

* reuse existing code
* change the implementation of a class used in a composition without adapting any external clients

1. When an object is destroyed, what happens to all the objects it is composed of?

the objects still exists.