**Polymorphism & Composition Homework - Quiz**

1. What does the ***word*** 'polymorphism' mean?

Polymorphism in programming means that an object can be many things.

1. What does it mean when we apply polymorphism to OO design? Give a simple Java example.

It means that the polymorphic object can take on the attributes of it’s own class as well as the attributes of the other classes it belongs to.

An example of this in Java would be a bedroom in a simple hotel application. The *bedroom* class would have its own properties and methods upon which it can call, but because it is also a *room*, it can call upon the methods available in the *room* class.

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

We can use inheritance and abstract classes to implement polymorphism in Java.

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

This would depend on the type of polymorphism being used, but generally, as the word suggests, they can take on multiple.

For clarity though, you should probably only really use two.

1. Give an example of when you could use polymorphism.

You could use polymorphism in a game application taking in a Sports team. There would be certain attributes that all *Players* would have, and then have separate classes for *Forwards*, *Defencemen*, and *Goalie*, which encapsulate the individual functions specific to those positions and classes.

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

Composition in OOP is when an object is composed of other objects.

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

An example of composition in Java would be if we were modelling vehicles. *Vehicle* would be the superclass from which *Car* inherits many attributes, however composition would be used when constructing the *Car* to pass in the *Engine* class.

*Engines* do not have the same functions and attributes as a *Car*, and therefore cannot inherit from the *Car* class or *Vehicle* superclass., but a *Car* still has an *Engine* and composition affords us the opportunity for these classes to interact.

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

As aforementioned, composition affords us the opportunity to allow classes to interact with each other when there is no inheritance or polymorphism.

1. What happens to the behaviours when the object composed of them is destroyed?

They are also destroyed.