**Polymorphism**

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

Polymorphism means that many objects of different types can take on the same form through one interface.

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

An indicator of polymorphism is using the ‘is a’ in-between two objects. For example; a car is a vehicle, a helicopter is a vehicle, a boat is a vehicle. Each of these vehicle objects have similar ‘vehicle’ behaviours and variables, however all three can also have their own specific behaviours and variables.

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

We can use interfaces and abstract classes in Java to achieve polymorphism.

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

Many objects can take on many forms.

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

You could use polymorphism to create relationships between data and objects. An example could also be animals (as the parent class) and dog, cat, bird (as child classes).

**Composition**

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

Composition is when an object is ‘part of’ another object, one object cannot practically exist without the other. For example a wall is part of a house, a roof is part of a house.

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

A car is made of and engine, gearbox, wheels.

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

Advantages include: more efficient code, flexible to adaptation.

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

All the objects its composed of are also deleted.