Polymorphism & Composition

What does the word 'polymorphism' mean?

Polymorphism comes from two Greek words, poly meaning many, and morph meaning change, in that it can have many forms.

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

It increases the reusability of code, wrapping objects in an enclosing type that defines a contract between them all.

```
public interface IConnect {
    public String connectionStatus(String networkName);
}


public class Printer implements IConnect {
    public String print(String data){
        return "printing: " + data;
    }
    public String connectionStatus(String network) {
        return String.format("Ink Printer connected to network %s.", network);
    }
}
```

What can we use to implement polymorphism in Java?

Polymorphism can be implemented using abstract classes through inheritance, or by an implemented interface.

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

Many forms where a parent class refers to a child class object.

Give an example of when you could use polymorphism.

When a parent class is used to refer to a child class object.

Composition

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

Composition is a relationship where an object is constructed of one or more other objects.

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

If/when there is a HAS-A relationship between objects.

```
public class TypeWriter extends PrintingDevice {
 public TypeWriter(String make, String model) {
     super(make, model);
  }
}
```

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

To allows a class to use behaviours from a group of other classes, reusing code.

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

An object composed of other behaviours owns the behaviours and when it is destroyed, its behaviours are destroyed.