**Term Definition**

**base class**

One of two classes in an inheritance relationship, the members of the base class are inherited by the derived class.

**delegation**

Where an aggregation/composition relationship is used instead of inheritance, and the implementation of one classes methods are 'delegated' to an object of a second class.

**derived class**

One of two classes in an inheritance relationship, the members of the derived class are inherited from the base class.

**generalisation**

A view of inheritance, where the base class represents a more general version of the derived class.

**inheritance**

A class relationship where one class, the derived class, is able to build upon, or extend, the functionality defined in another class, the base class.

**inheritance hierarchy**

A collection of two or more classes that are related by an inheritance relationship, forming a hierarchical (tree-like) structure.

**Liskov Substitution Principle**

A way of testing whether or not it is appopriate to use an inheritance relationship, where an object of the derived class should be able to be used wherever an object of the base class would have been used.

**method hiding**

Occurs when a method is defined in a derived class with the same name as a method defined in the base class, thereby hiding the method in the base class for objects of the derived type.

**refactoring**

The process of modifying the design of an existing program to either improve the existing design, or to carry out necessary design changes to facilitate more functionality.

**specialisation**

A view of inheritance, where the derived class represents a more specific version of the base class.