**Implementation independent** system models (such as the analysis models) that are not tied to a particular software or hardware platform and can be implemented in a variety of different ways.

**Incremental development** a life cycle model in which the system is partitioned according to areas of functionality. Each major functional area is developed and delivered independently to the client.

**Information hiding** making the internal details of a module inaccessible to other modules.

**Inheritance** a relationship between two classes where one is a refinement of the other; sometimes referred to as the 'is-a' relationship. A mechanism that allows a class to reuse features already defined in another class.

**Initiating actor** the actor who starts off the sequence of events in a use case.

**Input** data which is entered into the system by the user.

**Instance** an object that belongs to a particular class.

**Instantiation** the creation of a new instance of a class (i.e. an object).

**Interaction** a set of messages exchanged between objects to achieve a specific goal.

**Interaction diagram** diagram showing a set of messages that take place between objects to achieve a specific goal.