Programming Paradigms

Procedural: Set of processes that are connected by a hierarchy

Object-Oriented: Set of objects working together to achieve the goals of the application

Object: a primary modelling element in OOP

Classes: these define attributes and behaviours and work as a template for creating objects of the same type

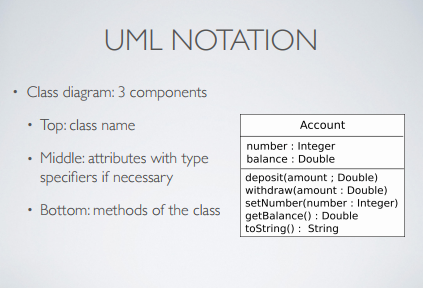
Attributes: these are properties of an object that stores data about it. The state of an object is the set of values of each of its attributes

Behaviour: This is a service provided by the object. It is an action that the object performs

Accessors: allow access to the state of an object. A simple method that returns different attributes. Also known as getters

Mutators: modify the state of an object. Known as setters.

An object is known as an instance of a class



Abstraction takes the essential characteristics of an object that distinguishes it from other objects to provide well defined boundaries.

Encapsulation is the bundling of data and methods that create the behaviours of the object.

Encapsulation is usually achieved by information hiding. Information Hiding is the process of hiding all the secrets of an object that do not contribute to its essential characteristics.

Constructors are used to create an object and/or initialise its state

Method Overloading: The process of writing methods in the same class with the same name but with different method signatures

Instance Variables: a space for part of the state of an object

Class Variables: Part of the state of a class. A class variable is shared by all instances of the same class  
eg. Private static int sensorIDCounter

Relationships

Relationships between classes are established to either:  
- Indicate some sort of sharing between the classes  
- Indicate a semantic connection between the classes

Types of Relationships

Association/Dependency (uses)  
Generalisation/Specialisation (is-a)  
Composition/Aggregation (part-of)

Associations: This denotes a semantic dependency between objects. This can be bidirectional or in one direction

Inheritance is a relationship among classes wherein one class shares the structure and/or behaviour defined in one or more other classes

A class from which another class gets its structure from is called a super class. A class that inherits from one or more classes is called a subclass

String class is an immutable class. No mutators and not possible to change the state of the class after it has been created.

Looking at the equals() method of the Object class:  
public Boolean equals(Object obj)

This method returns true if the current object is stored at the same memory address as obj and false otherwise

An aggregation relationship models a whole-part relationship between two classes. It is also referred to as a part-of relationship.

Consider two classes A and B. A which is considered as the whole consists of instances of B referred to as parts

Composition is a form of aggregation with stronger ownership between the whole and its parts

Where as with the regular form of aggregation, the whole and the parts may live on without each other but in composition the parts in the relationship live and die with the whole. In other words, the whole is responsible for the creation and destruction of its parts.