**ABSTRACT CLASS**

An abstract class is a class used to define the general structure of an object; it contains abstract methods that can be implemented by subclasses to define what they do in the specific instance.

The Account class in my project is an abstract class and will define, the basic structure of the savings and current account.

**CONCRETE CLASS**

A concrete class is a java class that has an implementation for all its methods.

Concrete classes will be used in my project to define the Bank(main), BankManager, BankEmployee, currentAccount, Customer, SavingsAccount

**INHERITANCE**

Inheritance is an object-oriented concept of a subclass being able to access the methods and properties of the super class, it can be used to create natural extensions of an object

In my project, BankEmployee inherits from BankManager, because the employee is an extension of the managers services,

Savings account and Current account, both inherit from Account because they are different types of account

**POLYMORPHISM**

Polymorphism is the ability of an object to be represented in different forms, due to the inheritance of properties and method from other classes, an object can be represented or type casted into another object

I use polymorphism in my project in the BankManager class, the BankManager is type casted into a BankEmployee object, this object is then passed into the customer methods, deposit, withdraw and transaction history, to attend the customer needs

**PUBLIC**

Public access modifier, is used to indicate a method or property is accessible to all classes in the project,

Public method are used in all constructor in the project so an object can be instantiated in any class

**PRIVATE**

Private access modifier, is used to indicate a method or property is accessible to only the class that defines it, it is used to protect the method from being accidentally or intentionally called or property used by other classes

Private method are used in the BankManager class to initializeCustomerDatabase() method, because no other class is permitted to create the customers.txt file

**PROTECTED**

Protected access modifier, is used to indicate a method or property is only accessible to subclasses, this is used to protect the method or property from being accessed by classes that do not need it or fit the use of it

Protected is used in this project to define methods and properties in BankManager and BankEmployee, because BankEmployee is a subclass of BankManager