Software development

Nested classes

2 types

* Static nested class cannot access instance member of parent class
  + Defined using class OuterClass {
    - Static class StaticNestedClass{}
    - }
* Inner class does have access to instance members
  + Defined the same way but without static

Outerclass can only be defined with either public or the default (package private)

Nested class has no such limitations

Class Car {

Static class Gearbox {

}

}

Access gearbox using Car.Gearbox

Car.Gearbox nestedObject = new Car.Gearbox();

Inner classes

Often difficult to spot in code

Produce 2 class files on compilation

OuterClass.class

OuterClass$InnerClass.class

Local classes

Often very difficult to spot within code

Local classes can access effectively final variables, these are variables that have not been changed since they have been initialised but do NOT have the final key word

Anonymous classes

Why use anonymous classes

Makes the code more concise

Can then make sure that you use instantiation and declaration at the same time

Cannot have enumerated class within a local class within a method because it cannot have the static modifier, (everything must be declared as final)