**Anonymous class:**

Anonymous classes enable you to make your code more concise. They enable you to declare and instantiate a class at the same time. They are like local classes except that they do not have a name. Use them if you need to use a local class only once. An anonymous class is an expression.

Parentheses that contain the arguments to a constructor, just like a normal class instance creation expression. (when we override other class: abstract or concrete) **Note**: When you implement an interface, there is no constructor, so you use an empty pair of parentheses, as in this example.

**Scope of anonymous class**

* An anonymous class has access to the members of its enclosing class.
* An anonymous class cannot access local variables in its enclosing scope that are not declared as final or effectively final.
* Like a nested class, a declaration of a type (such as a variable) in an anonymous class shadows any other declarations in the enclosing scope that have the same name.

**Restriction of Anonymous class:**

* You cannot declare static initializers or member interfaces in an anonymous class.
* An anonymous class can have static members provided that they are constant variables.

Note that you can declare the following in anonymous classes:

* Fields
* Extra methods (even if they do not implement any methods of the supertype)
* Instance initializers
* Local classes

However, you cannot declare constructors in an anonymous class.