Anonymous means nameless.

**Anonymous Inner Class**

It is a class without name.

**public** **interface** MyInterface {

**public** **void** m1();

}

In below class, we are implementing m1 method, without inheritance (see that there is no implements keyword).

Where is implemented class name? The highlighted yellow color is implemented class (a class without name)

**public** **class** AnnonymousClassExample {

**public** **static** **void** main(String[] args) {

MyInterface s = **new** MyInterface() {

**public** **void** m1() {

System.***out***.println("I am inside annonymous class");

}

};

s.m1();// calling method

}

}

It can extend abstract and concrete class.

It can implement an interface that contains any number of abstract methods.

Inside this we can declare instance variables.

Anonymous inner class can be instantiated.

Inside Anonymous inner class, “this” always refers to current anonymous inner class object but not to outer object.

It is the best choice if we want to handle multiple methods.

At the time of compilation, a separate .class file will be generated.

Memory allocation is on demand, whenever we are creating an object.

**Anonymous function**

Anonymous function means a method with no name and no modifier.

We can see how to implement Anonymous function using lambda (next section)