Inner Class

```
public class OuterClass {
    private int x = 99;
    public OuterClass() {
        System.out.println("constructor...x=" + x);
    }
    @Override
    public String toString() {
        final int inString = 111; //has to be final, because used in Local Class
        class LocalClass {
            private int y = 999;
            /* static not allowed */
            public String xy() {
                x += inString;
                return ("[local class] x+y=" + x + "+" + y);
            }
        }
         //inString++;
        LocalClass lc = new LocalClass();
        return "toString() in OuterClass; " + lc.xy() + "; " + lc.getClass();
    }
    public class InnerClass {
        private int y = 101;
        /* static not allowed */
        public void xy() {
            x++;
            System.out.println("[inner class] x+y=" + x + "+" + y);
        }
    }
}
```

Usage

```
System.out.println("...new OuterClass...");
OuterClass oc = new OuterClass();
System.out.println("...new InnerClass...");
OuterClass.InnerClass ic = oc.new InnerClass();
System.out.println("...2nd new InnerClass...");
OuterClass.InnerClass iic = new OuterClass().new InnerClass();
System.out.println("...2nd new OuterClass...");
OuterClass ooc = new OuterClass() {
    @Override
    public String toString() {
        return "x2==unknown";
    }
};
System.out.println("...toString Outerclass...");
System.out.println(oc.toString() + "," + oc.getClass().toString());
System.out.println("...toString Innerclass...");
System.out.println(ic.toString() + "," + ic.getClass().toString());
System.out.println("...toString Outerclass-new...");
System.out.println(ooc.toString() + "," + ooc.getClass().toString());
//Syntax: ic instanceof OuterClass?“is instance“:“is not“;outcome:
```

result:

```
...new OuterClass...
constructor...x=99
...new InnerClass...
...2nd new InnerClass...
constructor...x=99
...2nd new OuterClass...
constructor...x=99
...toString Outerclass...
toString() in OuterClass; [local class] x+y=210+999; class pkg01innerclass.OuterClass$1LocalClass,class
pkg01innerclass.OuterClass
...toString Innerclass...
pkg01innerclass.OuterClass$InnerClass@70dea4e,class pkg01innerclass.OuterClass$InnerClass
...toString Outerclass-new...
x2==unknown,class pkg01innerclass.Main$1
```

Anonymous Class

```
public class Mole extends Animal {
    private String color = "black";
…
    @Override
    public String toString() {
        return super.toString() + " ... Mole{" + "color=" + color + '}';
    }
    public int callAnonym() {
        Comparable ano = new Comparable()  {
            @Override
            public int compareTo(Object o) {
               System.out.println("in compare: " + color + "," + getName() +
                                 "," + toString()+ "," + super.toString());
               color = "no black"; //but NOT this.color !!
               return 22;
            }
        };
        ano.compareTo(null);
        System.out.println("after compare(): " + color);
        return ano.compareTo(null);
        }
    }
```

result:

```
== Animal{name=Jerry}
== Animal{name=Jerry} ... Mole{color=black}
in compare:
black,Jerry,pkg02anonym.Mole$1@7852e922,pkg02anonym.Mole$1@7852e922
after compare(): no black
in compare: no black,Jerry,pkg02anonym.Mole$1@78522,pkg02anonym.Mole$1@78522
== 22
➔toString() (= class info) of Animal/Mole lost;
  attributes accessible
```