**When a class declare within the a method is known as Local inner class.**

class A

{

    int x = 10;

    void show()

    {

        int y = 10;

    }

}

class Q01\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        System.out.println( a.x );

        System.out.println( a.y );

    }

}

error: cannot find symbol

    System.out.println( a.y );

                         ^

class A

{

    int x = 10;

    void show()

    {

        int y = 20;

    }

    void show2()

    {

        System.out.println( x );

        System.out.println( y );

    }

}

class Q02\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show2();

    }

}

error: cannot find symbol

    System.out.println( y );

                        ^

class A

{

    int x = 10;

    void show()

    {

        class B

        {

            void show2()

            {

                System.out.println("B - show2");

            }

        }

        System.out.println( "A - show");

    }

}

class Q03\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

A - show

class A

{

    int x = 10;

    void show()

    {

        class B

        {

            void show2()

            {

                System.out.println("B - show2");

            }

        }

        B b = new B();

        b.show2();

        System.out.println( "A - show");

    }

}

class Q04\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

B - show2

A – show

class A

{

    void show()

    {

        class B

        {

            void show2()

            {

                System.out.println("B - show2");

            }

        }

    }

    void show1()

    {

        B b = new B();

        b.show2();

        System.out.println( "A - show1");

    }

}

class Q05\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show1();

    }

}

Q05\_Local\_Inner\_Class.java:15: error: cannot find symbol

                B b = new B();

                ^

After Compile :

**A.class**

**A$1B.class**

**Q05\_Local\_Inner\_Class.class**

class A

{

    void show1()

    {

        class B

        {}

        System.out.println("show - 1");

    }

    void show2()

    {

        class B

        {}

        System.out.println("show - 2");

    }

}

class Q06\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show1();

    }

}

show - 1

After Compile :

**A.class**

**A$1B.class**

**A$2B.class**

**Q06\_Local\_Inner\_Class.class**

class A

{

    void show1()

    {

        class B

        {}

        System.out.println("show - 1");

    }

    void show2()

    {

        class C

        {}

        System.out.println("show - 2");

    }

}

class Q07\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show1();

    }

}

show - 1

After Compile :

**A.class**

**A$1B.class**

**A$1C.class**

**Q07\_Local\_Inner\_Class.class**

class A

{

    void show1()

    {

        class B

        {}

        class C

        {}

        System.out.println("show - 1");

    }

    void show2()

    {

        class C

        {}

        System.out.println("show - 2");

    }

}

class Q08\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show1();

    }

}

show - 1

After Compile :

**A.class**

**A$1B.class**

**A$1C.class**

**A$2C.class**

**Q08\_Local\_Inner\_Class.class**

**class ko Constructor ke andr bhi define kar sakte he.**

class A

{

    A()

    {

        class B

        {

            void show2()

            {

                System.out.println("A - Constructor Show - 2");

            }

        }

B b = new B();

        b.show2();

    }

}

class Q09\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

    }

}

A - Constructor Show - 2

**class ko Instance Block ke andr bhi define kar sakte he.**

class A

{

    {       // Instance Block

        class B

        {

            void show2()

            {

                System.out.println("B Instance Block Show - 2");

            }

        }

        B b = new B();

        b.show2();

    }

}

class Q10\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

    }

}

B Instance Block Show - 2

**class ko Static Block ke andr bhi define kar sakte he.**

class A

{

    static

    {       // Static Block

        class B

        {

            void show2()

            {

                System.out.println("B Static Block Show - 2");

            }

        }

        B b = new B();

        b.show2();

    }

}

class Q11\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

    }

}

B Static Block Show – 2

**class ko if- else ke andr bhi define kar sakte he.**

class A

{

    void show()

    {

        if( true )

        {

            class B

            {

                void show2()

                {

                    System.out.println("B If. Show - 2");

                }

            }

            B b = new B();

            b.show2();

            System.out.println("A show if - else");

        }

    }

}

class Q12\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

B If. Show - 2

A show if - else

**class ko Loops ke andr bhi define kar sakte he.**

class A

{

    void show()

    {

        for( int i = 0; i < 2; i++)

        {

            class B

            {

                void show2()

                {

                    System.out.println("B If. Show - 2 ");

                }

            }

            B b = new B();

            b.show2();

            System.out.println("A show if - else");

        }

    }

}

class Q13\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

B If. Show - 2

A show if - else

B If. Show - 2

A show if - else

class A

{

    int x = 10;

    static int y = 20;

    void show()

    {

        class B

        {

            void show2()

            {

                System.out.println("X : " + x);

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        System.out.println("AA");

    }

}

class Q14\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

X : 10

Y : 20

AA

class A

{

    int x = 10;

    static int y = 20;

    static void show()

    {

        class B

        {

            void show2()

            {

                System.out.println("X : " + x);

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        System.out.println("AA");

    }

}

class Q15\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

error: non-static variable x cannot be referenced from a static context

                                System.out.println("X : " + x);

                                                            ^

**class ko static method ke andr bhi define kar sakte he.**

class A

{

    static int y = 20;

    static void show()

    {

        class B

        {

            void show2()

            {

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        System.out.println("AA");

    }

}

class Q16\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

Y : 20

AA

Local Class ke andr se jab direct call kar sakte he (compile + run).

class A

{

    void show()

    {

        int y = 201;

        class B

        {

            void show2()

            {

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        System.out.println("AA");

    }

}

class Q17\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

(this output after 1.8 version)

Y : 201

AA

javac-source 1.7 Q17\_Local\_Inner\_Class.java

Local variable y is accessed from within inner class need to declared final

**Local Class ke andr se change kare to Error aayega (Error).**

class A

{

    void show()

    {

        int y = 201;

        class B

        {

            void show2()

            {

                y = 1000;

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        System.out.println("AA");

    }

}

class Q18\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

error: local variables referenced from an inner class must be final or effectively final

    y = 1000;

    ^

Method ke andar variable by default final hota he.

we can't modify final variable.

**Local Class ke bahar or Method ke andar change kare OR local ke andar se call na kare no Error( Compile + run ).**

class A

{

    void show()

    {

        int y = 201;

        class B

        {

            void show2()

            {

                System.out.println("BB");

            }

        }

        B b = new B();

        b.show2();

        y = 1000;

        System.out.println("Y : " + y);

    }

}

class Q19\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

BB

Y : 1000

**Local Class ke bahar or Method ke andar change kare OR local ke andar se call kare tab Error( Compile + run ).**

**Jab v class me local me final variable ko likhate ke bad change karenge to error aayega But hum call karenge to Error nai aayega.**

**But hum change ke bad call karenge to Error aayega.**

class A

{

    void show()

    {

        int y = 201;

        class B

        {

            void show2()

            {

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        y = 20;

        System.out.println("Y : " + y);

    }

}

class Q20\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

error: local variables referenced from an inner class must be final or effectively final

                                System.out.println("Y : " + y);

                                                            ^

class A

{

    void show()

    {

        final int y = 201;

        class B

        {

            void show2()

            {

                System.out.println("Y : " + y);

            }

        }

        B b = new B();

        b.show2();

        y = 20;

        System.out.println("Y : " + y);

    }

}

class Q21\_Local\_Inner\_Class

{

    public static void main( String args[])

    {

        A a = new A();

        a.show();

    }

}

error: cannot assign a value to final variable y

                y = 20;

                ^