Inner class

※ Type of Inner Class:

1. Nested Inner class
2. Local inner class
3. Anonymous inner class
4. Static inner class

* ya par jo super class bola ha wo "outer-class" hai and sub-class bola hai wo "Inner-class" hai. Ya sabhi self-understanding ka liya hai.

Inner class

Nested Inner class

```
class Test
{
    p.s.v.main( )
    {
        Outer o=new Outer();
        o.outerDisplay();

        Outer.Inner i=new Outer().new Inner();
    }
}
```

--> Agar hma Inner class ka object ko direct call karna ho main method mai to ya uska mth. hai.

```
class Outer
{
  ① int x=10;

  ③ class Inner
    {
        int y=20;
        void innerDisplay( )
        {
            s.o.p(x);
            s.o.p(y);
        }
    }

  ② void outerDisplay( )
    {
        Inner i=new Inner();
        i.innerDisplay();
        S.o.p(i.y);
    }
}
```

--> Nested Inner class mai ham class ka andar hi one or more than one class ko define karta hai.

--> Jo bhi chij sub-class ka andar hota hai uska use ham class ka bhar object bna kar sakta hai. Lekin jo bhi item super class mai declare hai usa ham kabhi bhi kahi bhi direct use kar sakta hai.

Inner class

Local Inner Class

--> Jab ham method ke andhar kisi class ko define karta hai ushe "local inner class bolta hai ".

--> ise inner class ko ham sirf ushe class kai method mai access kar sakta hai jis mai method define hai (ise case mai class outer{}). Agar outside method access karna ka try krenga to error aata hai.

--> method mai local inner class ko call karna ka liya class ka object normaly create karta hai.

```
class Outer
{

    void Display( )
    {

        class Inner
        {
            void innerDisplay( )
            {
                s.o.p("Hello");
            }
        }

        Inner i=new Inner();
        i.innerDisplay();
    }
}
```

Inner class

Anonymous

--> Ise class ka mostly use abstract class ya phir interface mai hota hai.

--> Jab abstract class ka object create karta hai then use waqt ham abstract class ka method ko override bhi kar deta hai to ek unknown class create hota hai by default jisa anonymous class bolta hai.

--> Same interface ka case mai kaam karta hai.

```
abstract class My
{
  abstract void display();
}
```

```
class Outer
{
    public void meth()
    {
        My m=new My()
        {
            public void display()
            {
                s.o.p("Hello");
            }
        };

        m.display();
    }
}
```

Inner class

Anonymous

```
interface My
{
  void display();
}
```

```
class Outer
{

  public void meth()
  {

    My m=new My()
    {
      public void display()
        s.o.p("Hello");
      }
    };

    m.display();
  }
}
```

--> Same theory jo upper wala slide mai hai.

Inner class

```
class Test
{
    p.s.v.m()
    {
        Outer.Inner i=new Outer.Inner();
        i.display();
    }
}
```

static class ka object create karna ka tareka.

```
class Outer
{
    static int x=10;
        int y=20;

    static class Inner
    {
        void display()
        {
        ✓  s.o.p(x);
        ✗  s.o.p(y);
        }
    }
}
```

--> Ek satic class jo kis class ki ander define hota hai usa static class bolta hai.
--> Outer class ka non-static variable ko ham static class ka andhar access nahi kar sakta hai.
--> Outer class ka instance ko create kia bina ham main class mai direct object create kar sakta hai static class ka.