

Source Code:

```
class OuterClass {
    int x = 10;

    class InnerClass {
        int y = 5;
    }
}

public class Inner{
    public static void main(String[] args) {
        OuterClass myOuter = new OuterClass();
        OuterClass.InnerClass myInner = myOuter.new InnerClass();
        System.out.println(myInner.y + myOuter.x);
    }
}
```

Static Inner Class:

```
class OuterClass {
    int x = 10;

    class InnerClass {
        int y = 5;
    }
}

public class Inner {
    public static void main(String[] args) {
        OuterClass myOuter = new OuterClass();
        OuterClass.InnerClass myInner = myOuter.new InnerClass();
        System.out.println(myInner.y + myOuter.x);
    }
}
```

Access Outer Class From Inner Class

```
class OuterClass {
    int x = 10;

    class InnerClass {
```

```

        public int myInnerMethod() {
            return x;
        }
    }
}

public class Inner {
    public static void main(String[] args) {
        OuterClass myOuter = new OuterClass();
        OuterClass.InnerClass myInner = myOuter.new InnerClass();
        System.out.println(myInner.myInnerMethod());
    }
}

```

Outputs:

The screenshot shows a Java IDE with a code editor on the left and an output console on the right. The code editor contains the following code:

```

1 package demo;
2
3
4
5 class OuterClass {
6     int x = 10;
7
8     static class InnerClass {
9         int y = 5;
10    }
11 }
12
13 public class Inner {
14     public static void main(String[] args) {
15         OuterClass.InnerClass myInner = new OuterClass.InnerClass();
16         System.out.println(myInner.y);
17     }
18 }
19

```

The output console on the right shows the result of the program execution:

```

<terminated> Inner [Java Appli
5

```

```

1 package demo;
2
3
4 class OuterClass {
5     int x = 10;
6
7     class InnerClass {
8         int y = 5;
9     }
10 }
11
12 public class Inner {
13     public static void main(String[] args) {
14         OuterClass myOuter = new OuterClass();
15         OuterClass.InnerClass myInner = myOuter.new InnerClass();
16         System.out.println(myInner.y + myOuter.x);
17     }
18 }

```

▲ <terminated> Inner [Java App
15

```

1 package demo;
2
3
4 class OuterClass {
5     int x = 10;
6
7     class InnerClass {
8         public int myInnerMethod() {
9             return x;
10         }
11     }
12 }
13
14 public class Inner {
15     public static void main(String[] args) {
16         OuterClass myOuter = new OuterClass();
17         OuterClass.InnerClass myInner = myOuter.new InnerClass();
18         System.out.println(myInner.myInnerMethod());
19     }
20 }

```

▲ <terminated> Inner [Java Application] C:\User
10