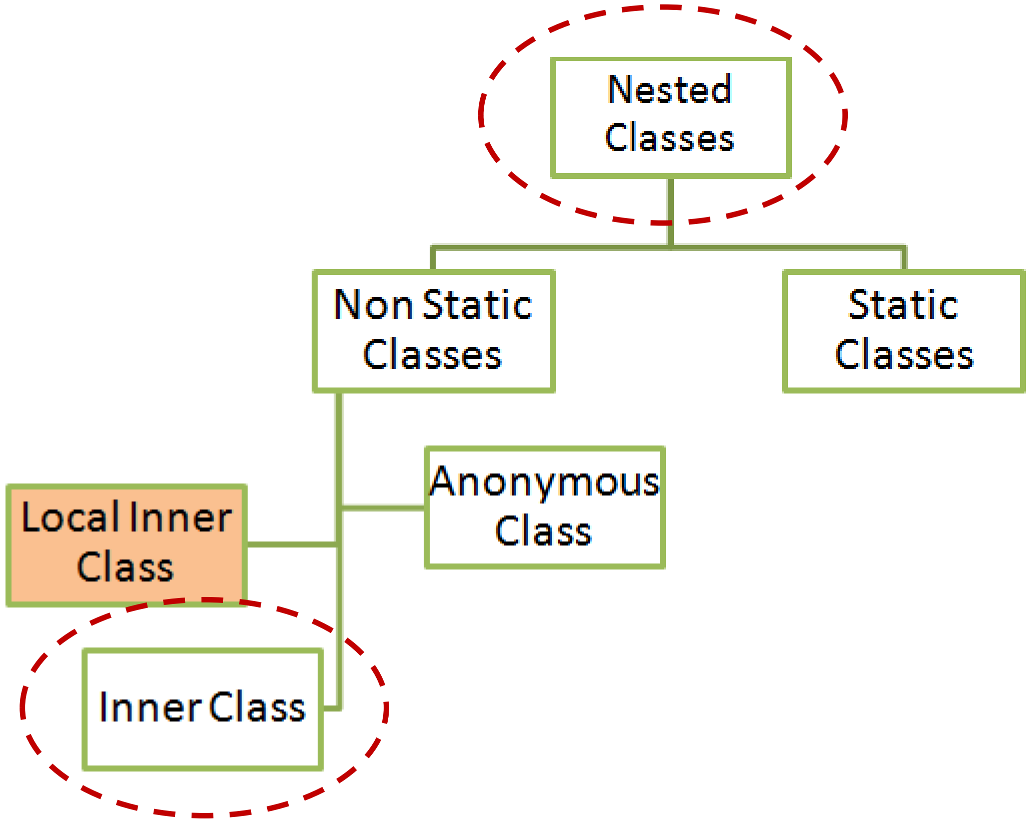
Nested Inner Class

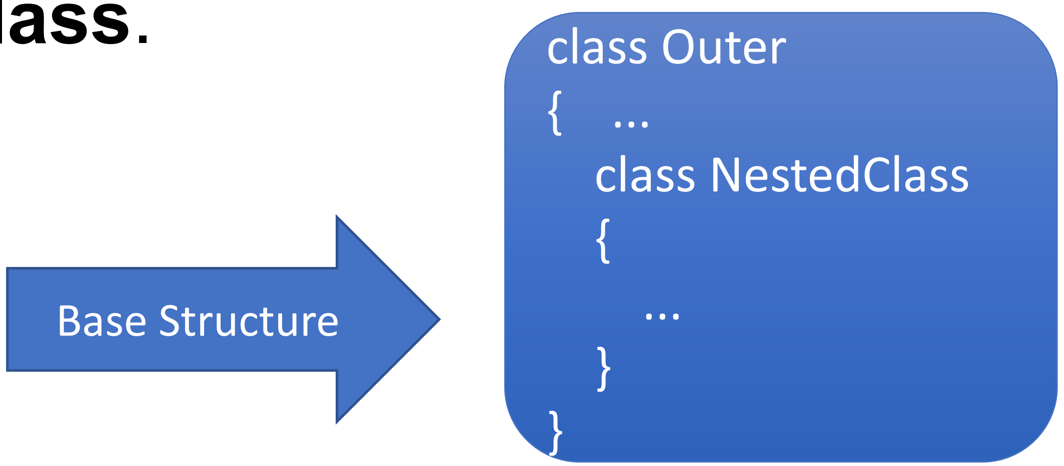
Sungchul Lee

Learning Object



Nested Class

- ➢Classes that are defined inside another class
  - ❑Like nested if statement, if statement inside other if
- ➢Purpose of a nested class
  - ❑Clearly group the nested class with its surrounding class, signaling that these **two classes are to be used together**
  - ❑the nested class is only to be used from **inside its enclosing (owning) class**.



```
class Outer
{   …
    class NestedClass
    {
        …
    }
}
```

Type of Nested Class

- ➢Four type of nested class
  - ❑Classes are inside of another class
  1. **Inner Class**
     - ❖ Increasing efficient to manage class
  2. Static Inner Class
  3. Local Inner Class (Next Lecture)
  4. Anonymous Inner Class (Next Lecture)

```
class Outer
{
   statement 1
   class Inner
   {
      statement 1-1
   }
}
```

Inner Class

- ➢ Outer Class
  - ❑ More than one inner class
  - ❑ Can not use Inner class member in Outer's method
    - ❖ Need declare and assign new object
- ➢ Inner Class
  - ❑ Can not use other outer's member
    - ❖ Need declare and assign new object
  - ❑ Can not use **static keyword inside block**

```
class Outer
{
   statement 1;
   class Inner 1
   {
      statement 1-1;
   }
  class Inner 2
   {
      statement 2-1;
   }
}
```

Generate Object (inner)

- Declare and initialize
  - Outer Class
    - Same as previous Class declare and initialize

      e.g.) Character character = new Character();

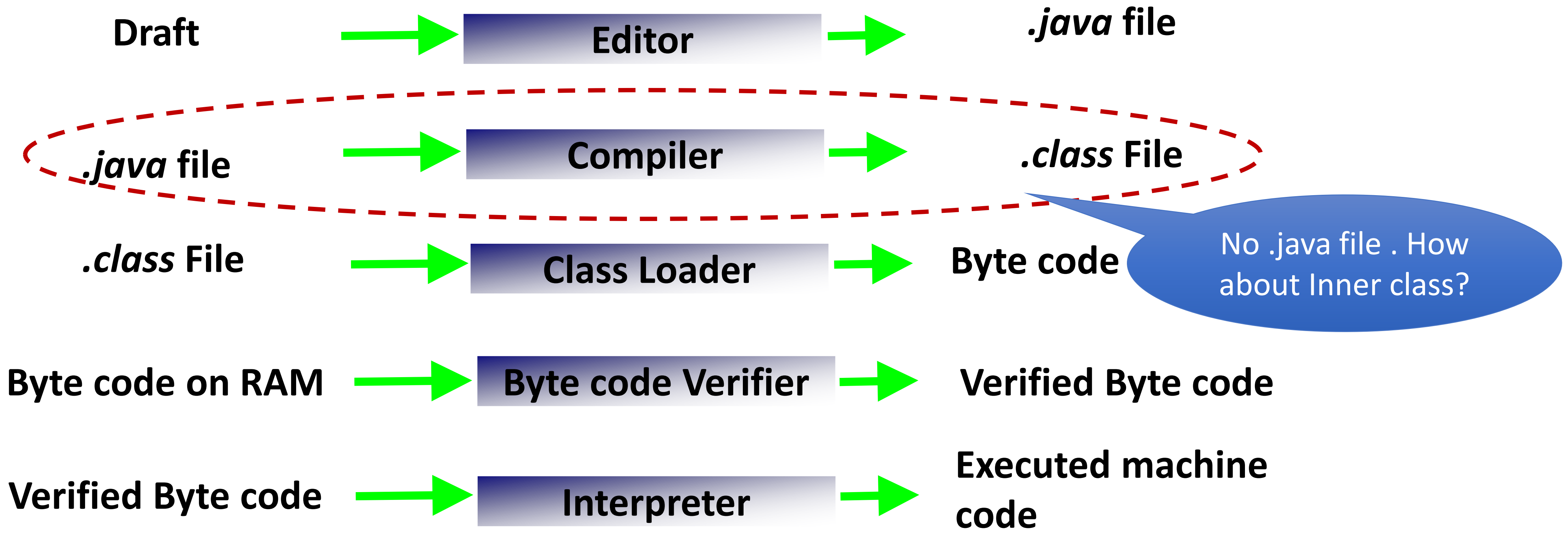
  - Inner Class
    - Use Outer Class Object.

      Syntax:

```
Outer outerName= new Outer();
Outer.Inner innerName = outerName.new Inner();
```

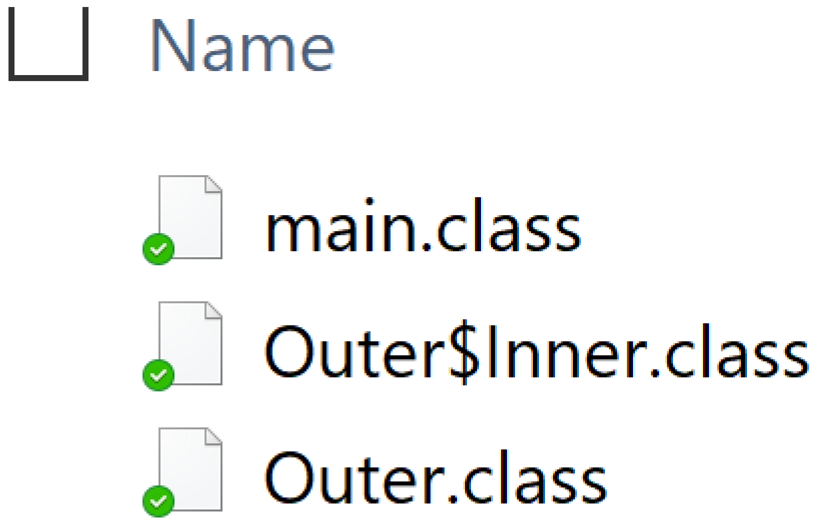
Java Environment

- Java programs normally go through five phases



Inner class in bin folder

- In project folder, inner class is generated by compiler
- "$" symbol is used to distinguish inner class.


Name

main.class

Outer$Inner.class

Outer.class

Practice

1. Make a new project (Reference: Create Project and Class File)
   - ❑Project name: Inner
2. Create a new Class File
   - ❑Class name: Main
   - ❑Class name: Outer
3. Coding:

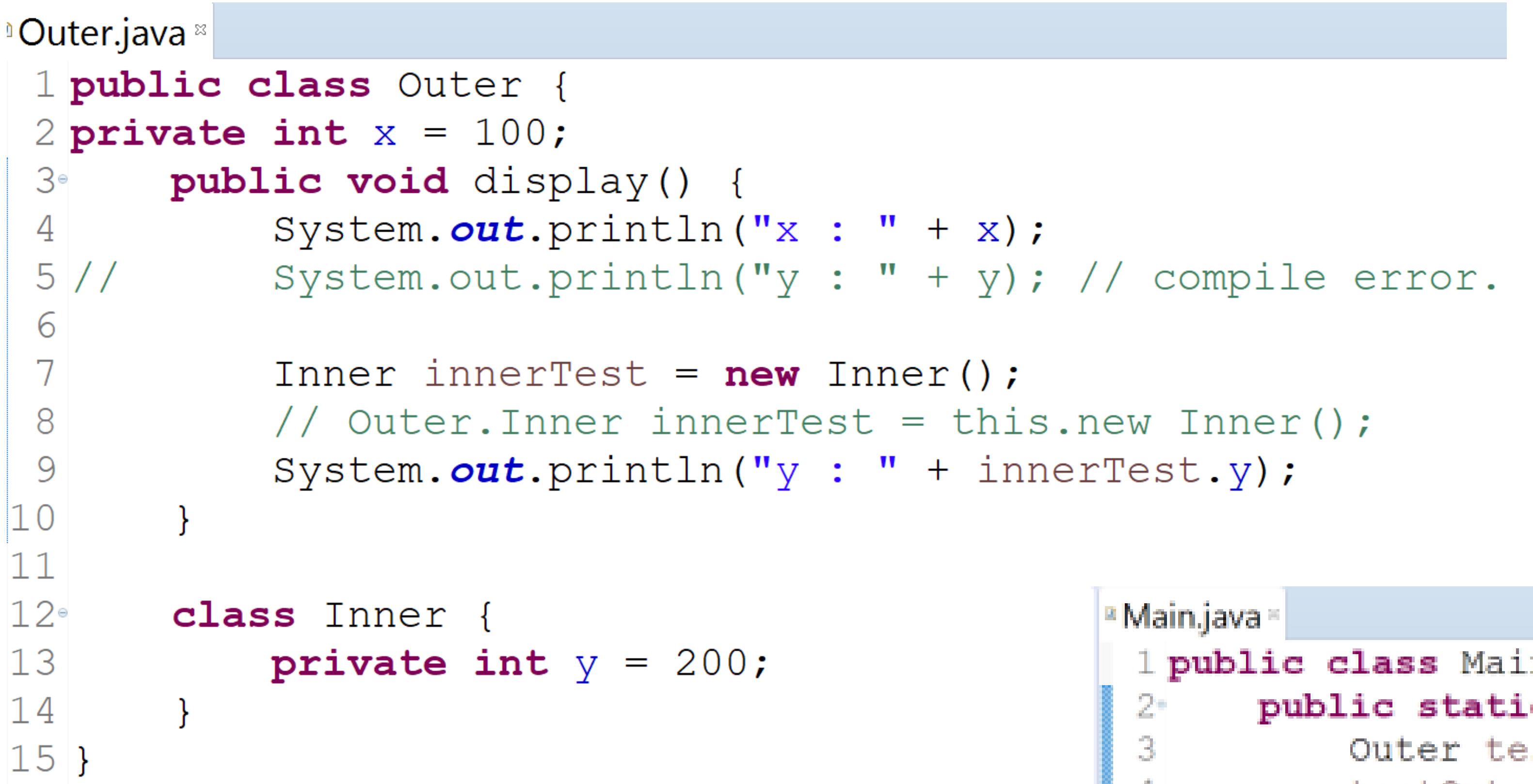
Practice – Code (Main)

```
public class Main {
    public static void main(String[] args) {
        Outer testOutet= new Outer();
        testOutet.display();

        Outer.Inner innerTest = testOutet.new Inner();
        System.out.println("y:" + innerTest.y);
    }
}
```

Practice – code (Outer)

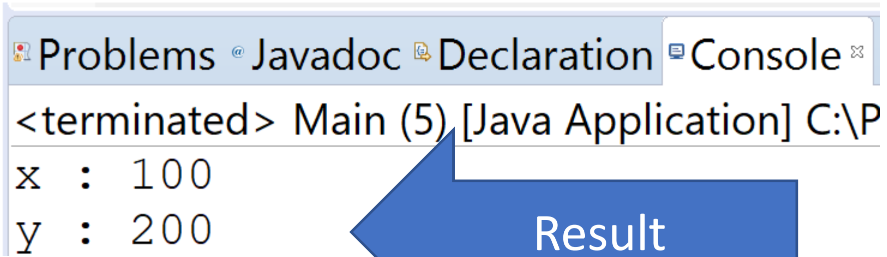
```
public class Outer {
private int x = 100;
    public void display() {
        System.out.println("x : " + x);
         Inner innerTest = new Inner();
System.out.println("y : " + innerTest.y);
    }
class Inner {
        public int y = 200;
    }
}
```

Practice – Code and Result



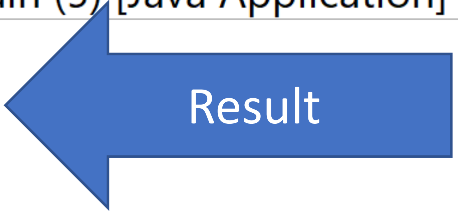
Outer.java

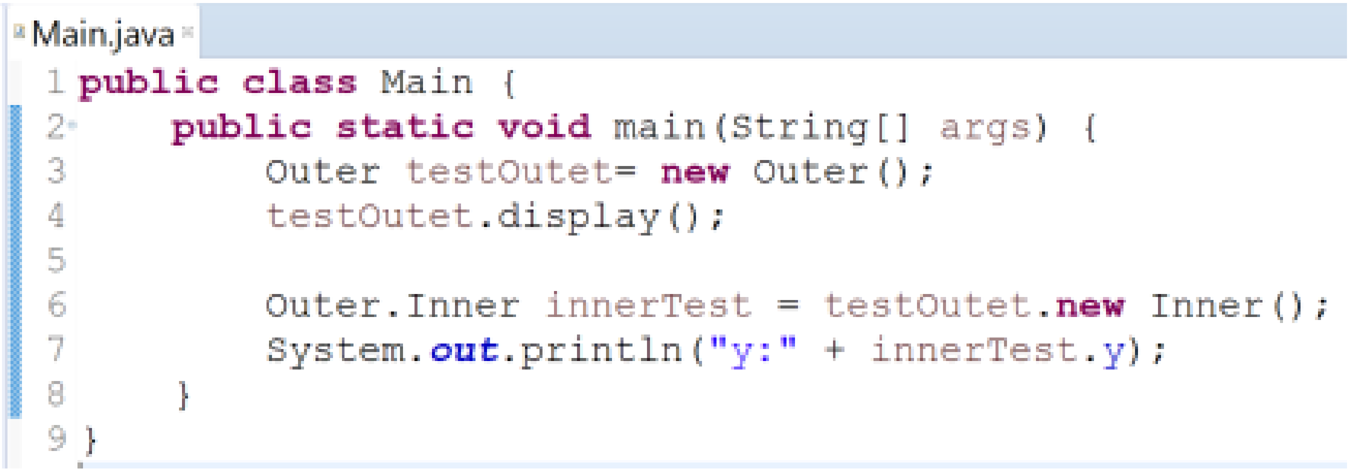
```java
public class Outer {
private int x = 100;
    public void display() {
        System.out.println("x : " + x);
//      System.out.println("y : " + y); // compile error.

        Inner innerTest = new Inner();
        // Outer.Inner innerTest = this.new Inner();
        System.out.println("y : " + innerTest.y);
    }

    class Inner {
        private int y = 200;
    }
}
```



Problems Javadoc Declaration Console

```
<terminated> Main (5) [Java Application] C:\P
x : 100
y : 200
```





Main.java

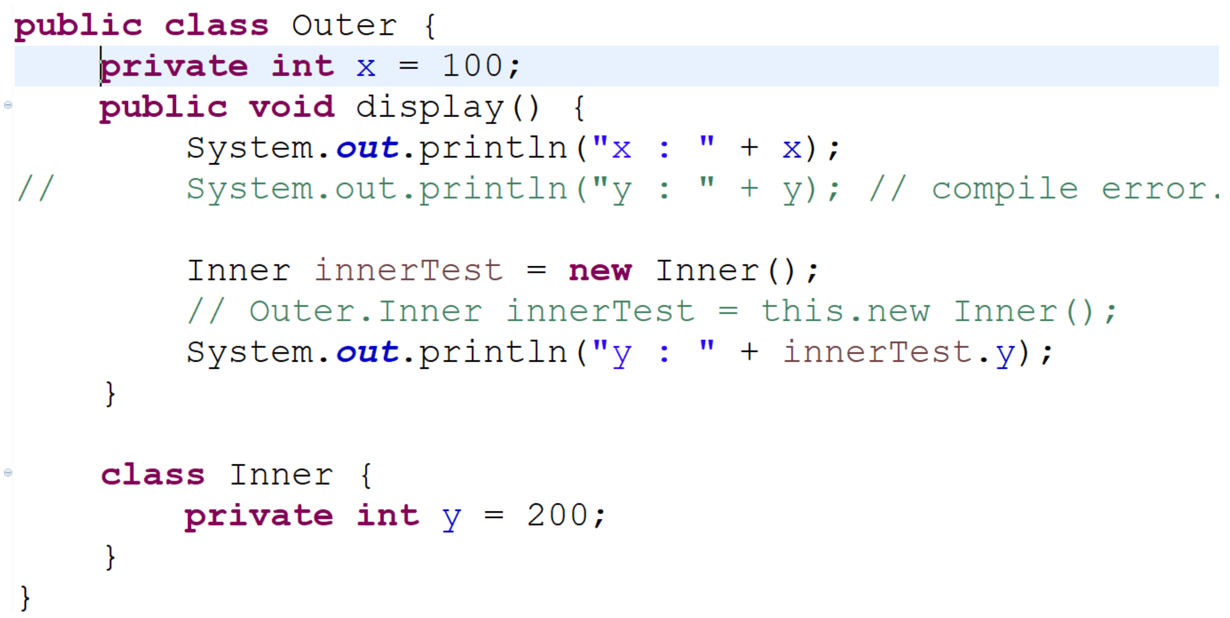
```java
public class Main {
    public static void main(String[] args) {
        Outer testOutet= new Outer();
        testOutet.display();

        Outer.Inner innerTest = testOutet.new Inner();
        System.out.println("y:" + innerTest.y);
    }
}
```

Summary

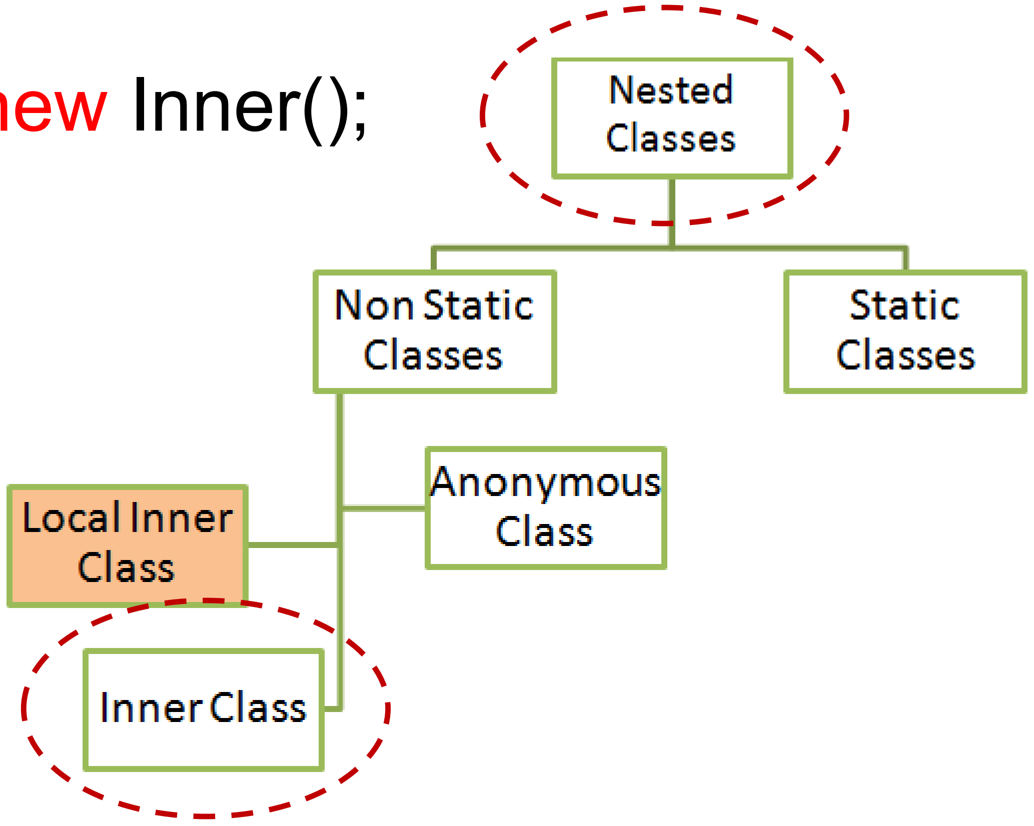
- Inner Class
  - Syntax:

Outer outerName= new Outer();

Outer.Inner innerName = outerName.new Inner();



```
public class Outer {
    private int x = 100;
    public void display() {
        System.out.println("x : " + x);
//      System.out.println("y : " + y); // compile error.

        Inner innerTest = new Inner();
        // Outer.Inner innerTest = this.new Inner();
        System.out.println("y : " + innerTest.y);
    }

    class Inner {
        private int y = 200;
    }
}
```



Nested Classes
Non Static Classes
Static Classes
Anonymous Class
Local Inner Class
Inner Class