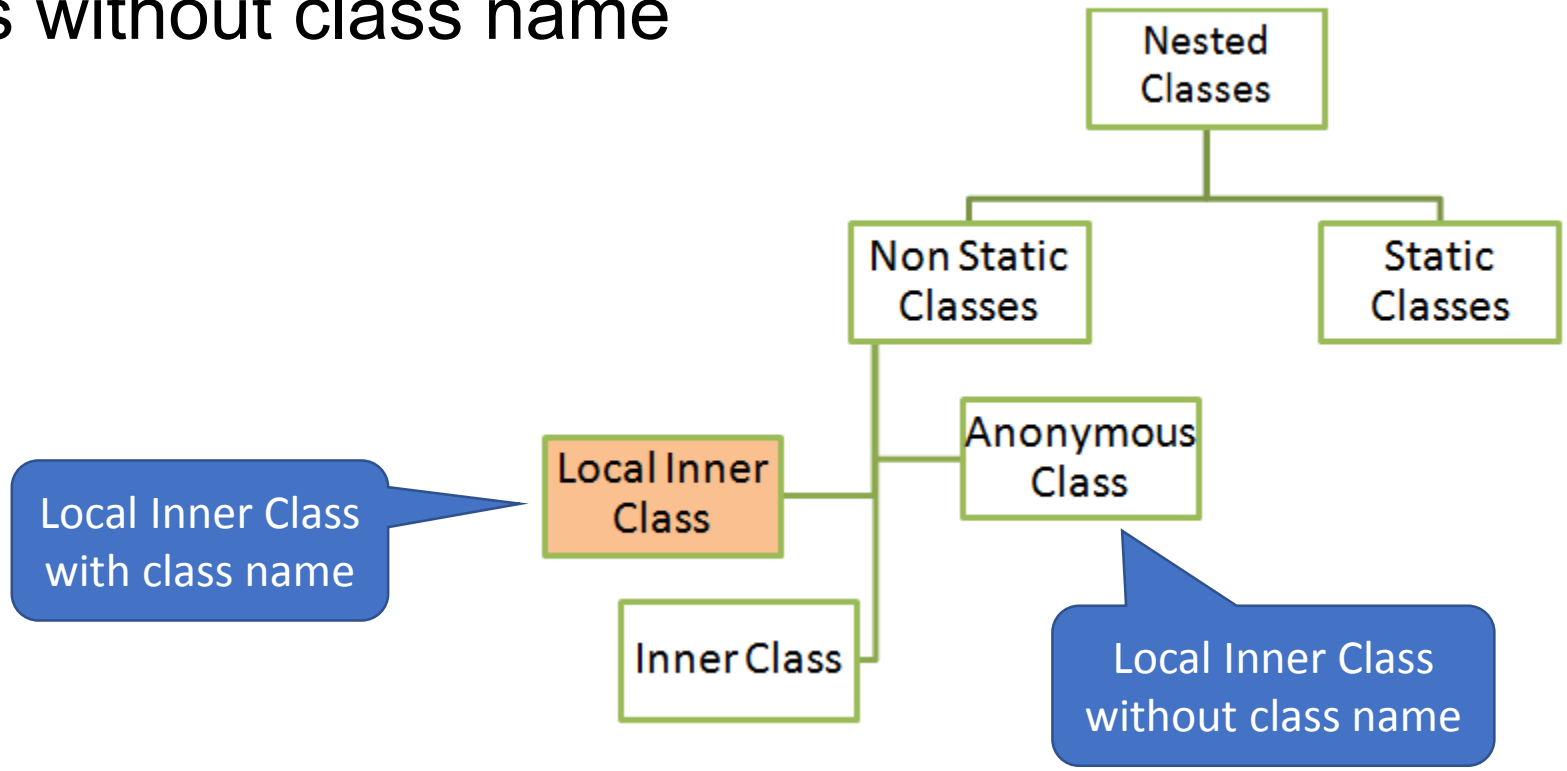
Anonymous Inner Class

Sungchul Lee

Learning Object

- Anonymous Inner Class
  - Local Inner Class without class name



Type of Nested Class

- Four type of nested class
  - Classes are inside of another class
  1. Inner Class
  2. Static Inner Class
     - static member
  3. Local Inner Class
     - Class in Method
  4. **Anonymous (Local) Inner Class**
     - **Overloading method in class**
     - **Derived from Local Inner**

```
 statement
}
class Outer
{
   statement 1
            method(){
                        new
Inner()
                        {

            statement 1-1
                        };
            }
```

Local Inner Class

- Anonymous Inner class is also local inner class
- The inner classes that are defined inside a block.
  - Cannot create the object outside a block
  - Working on only in the method
- Not a member of any enclosing classes
- Cannot have any access modifiers
  - JDK 7- Local inner class can access only final local variable of the **enclosing block**
  - JDK 8- it is possible to access the non-final local variable of enclosing block in local inner class.

Anonymous Inner Class

- It is an inner class without a name and for which only a single object is created.
- Useful when making an instance of an object with certain "extras" such as overloading methods of a class
  - **Cannot use static member**
- Only new keyword and statements
  - Overloading an existing class

Generate Object (Anonymous Inner)

- Declare and initialize
  - Syntax:

    Inner innerName =new Inner() { Overriding code }; // in method
  - ‘$’ and number for Anonymouse inner class
  - Overriding the existing class to modify
    - Save code

Practice

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

Practice – Code (Main)

main.java

```
1
2  public class main {
3
4      public static void main(String[] args) {
5          // TODO Auto-generated method stub
6          int x = 100;
7          Justclass jc= new Justclass();
8          jc.display();
9
10         Justclass jc_inner = new Justclass() {
11             public void display() {
12                 System.out.println("Overriding");
13                 System.out.println("x: " +x);
14                 System.out.println("y: "+y);
15             }
16             public int add(int a, int b) {
17                 System.out.println("A+B = "+ (a+b));
18                 return a+b;
19             }
20         };
21         jc_inner.display();
22     }
23
```

Practice – Code (Justclass)

Justclass.java

```
1
2  public class Justclass {
3      int y = 200;
4      public Justclass() {
5          this.display();
6      }
7
8      public void display() {
9          System.out.println("Justclass");
10     }
11
12     public int add(int a, int b) {
13         return a+b;
14     }
15 }
```

Summary

- Anonymous (Local) Inner Class
  - Local Inner Class in method
  - No name
  - Override the existing class

Justclass.java

```
1
2  public class Justclass {
3      int y = 200;
4      public Justclass() {
5          this.display();
6      }
7
8      public void display() {
9          System.out.println("Justclass");
10     }
11
12     public int add(int a, int b) {
13         return a+b;
14     }
15 }
```

main.java

```
1
2  public class main {
3
4      public static void main(String[] args) {
5          // TODO Auto-generated method stub
6          int x = 100;
7          Justclass jc= new Justclass();
8          jc.display();
9
10         Justclass jc_inner = new Justclass() {
11             public void display() {
12                 System.out.println("Overriding");
13                 System.out.println("x: " +x);
14                 System.out.println("y: "+y);
15             }
16             public int add(int a, int b) {
17                 System.out.println("A+B = "+ (a+b));
18                 return a+b;
19             }
20         };
21         jc_inner.display();
22     }
23
24 }
```