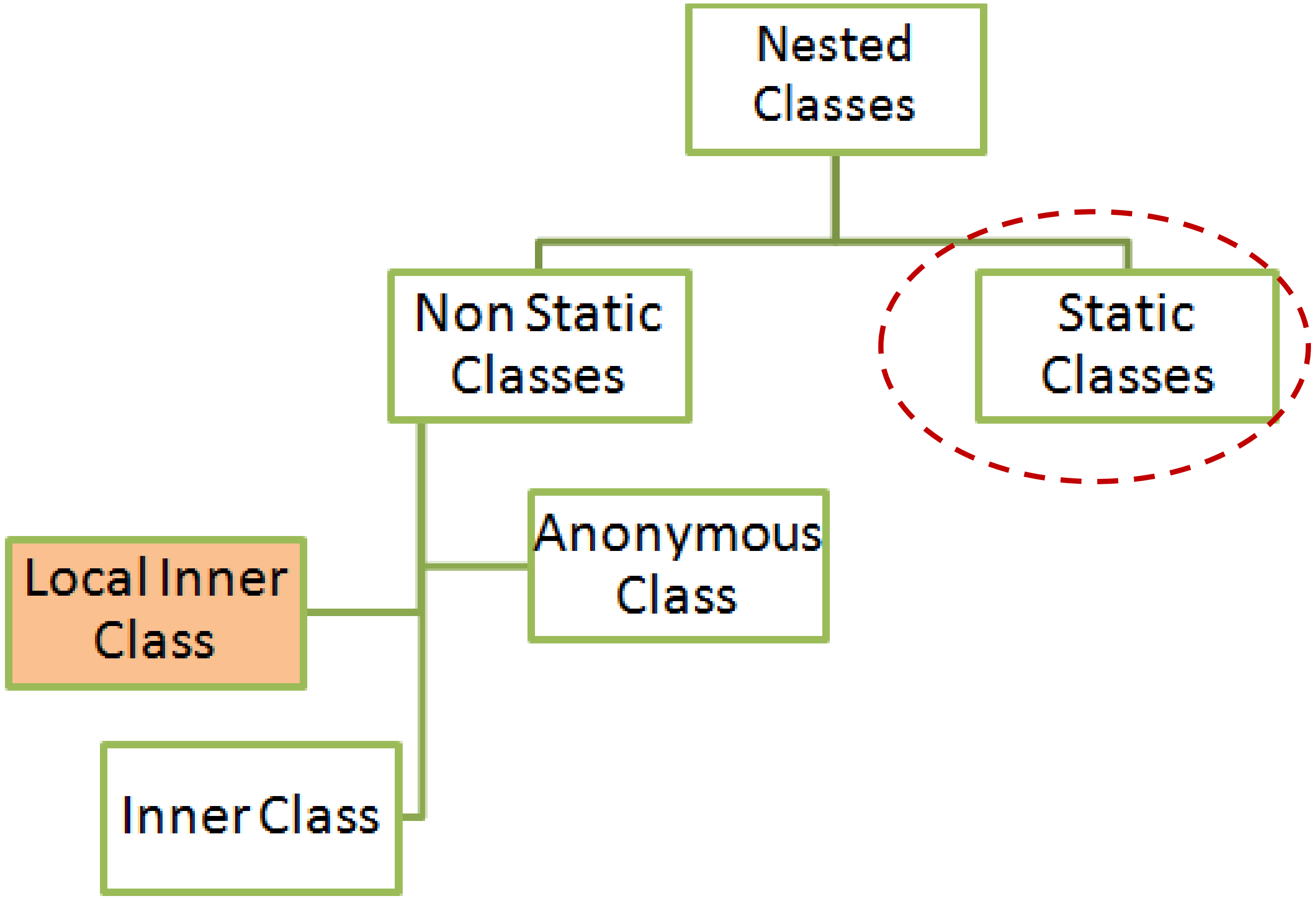
Static Inner Class

Sungchul Lee

Learning Object



Type of Nested Class

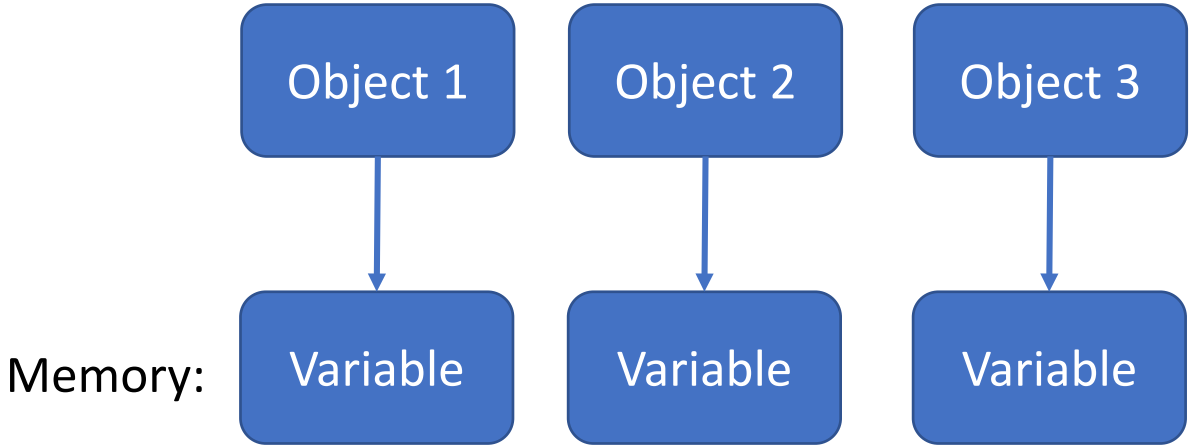
- Four type of nested class
  - Classes are inside of another class

  1. Inner Class
     - Increasing efficient to manage class
  2. **Static Inner Class**
     - static member
  3. Local Inner Class
  4. Anonymous Inner Class

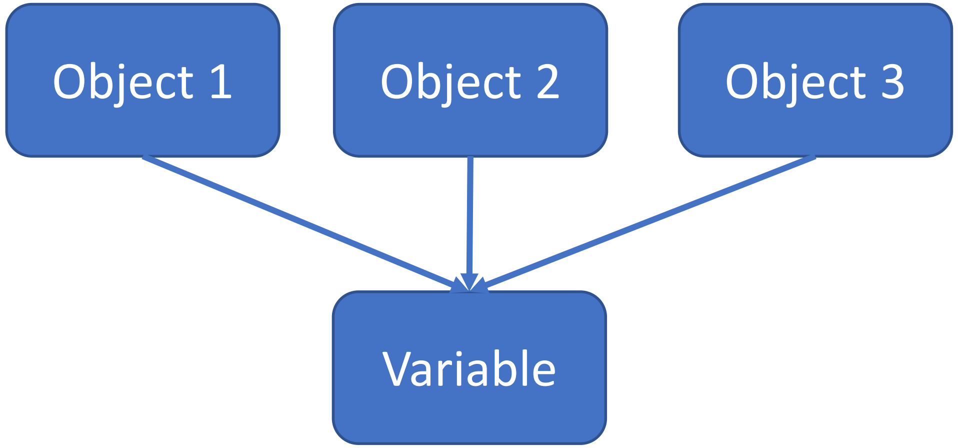
```
class Outer
{
    statement 1
    static class Inner
    {
        statement 1-1
    }
}
```

Instance vs. Static

- A variable of method that is **dependent on a specific instance** of the class should be an **instance** variable or method.
- A variable or method that is **not dependent on a specific instance** of the class should be a **static** variable or method.



Instance



Static

Static Inner Class

- Using the **static** keyword in front of Class name
- Not dependent on a specific instance
  - Independently create object
- Can make static member
  - static methods and static variables in static Inner
- Only can access static member in Outer class
  - Cannot access instance member (without declare and initialize)

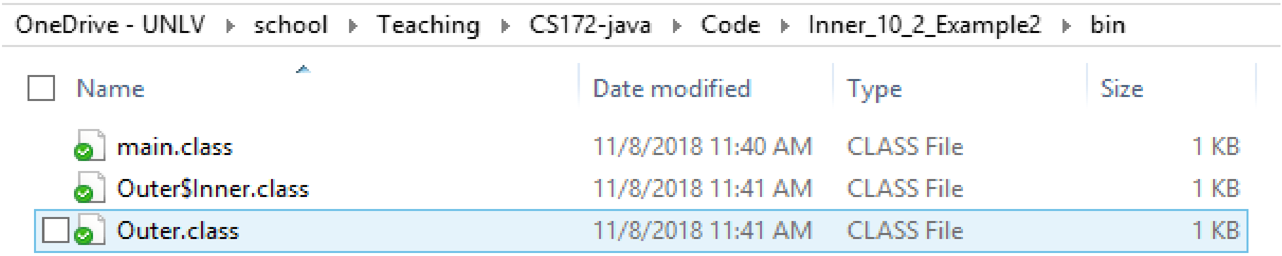
Generate Object (static inner)

- Declare and initialize
  - Create Inner class object without Outer's object
  - Syntax:

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

    - Don't need outer object.

By Compiler



OneDrive - UNLV ▸ school ▸ Teaching ▸ CS172-java ▸ Code ▸ Inner_10_2_Example2 ▸ bin

□ Name	Date modified	Type	Size
main.class	11/8/2018 11:40 AM	CLASS File	1 KB
Outer$Inner.class	11/8/2018 11:41 AM	CLASS File	1 KB
□ Outer.class	11/8/2018 11:41 AM	CLASS File	1 KB

Practice

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

Practice – code (Main)

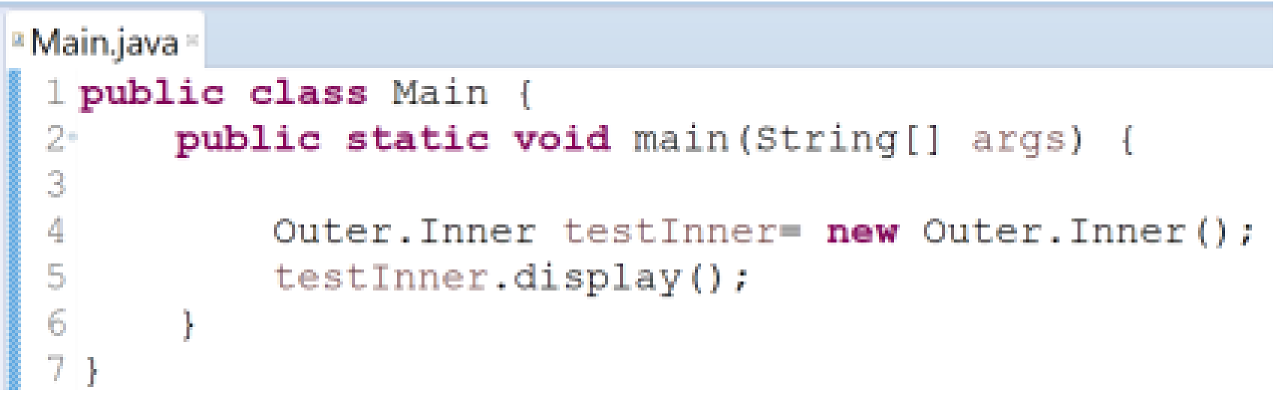
```
public class Main {
    public static void main(String[] args) {

        Outer.Inner testInner= new Outer.Inner();
        testInner.display();
    }
}
```

Practice – code (Outer)

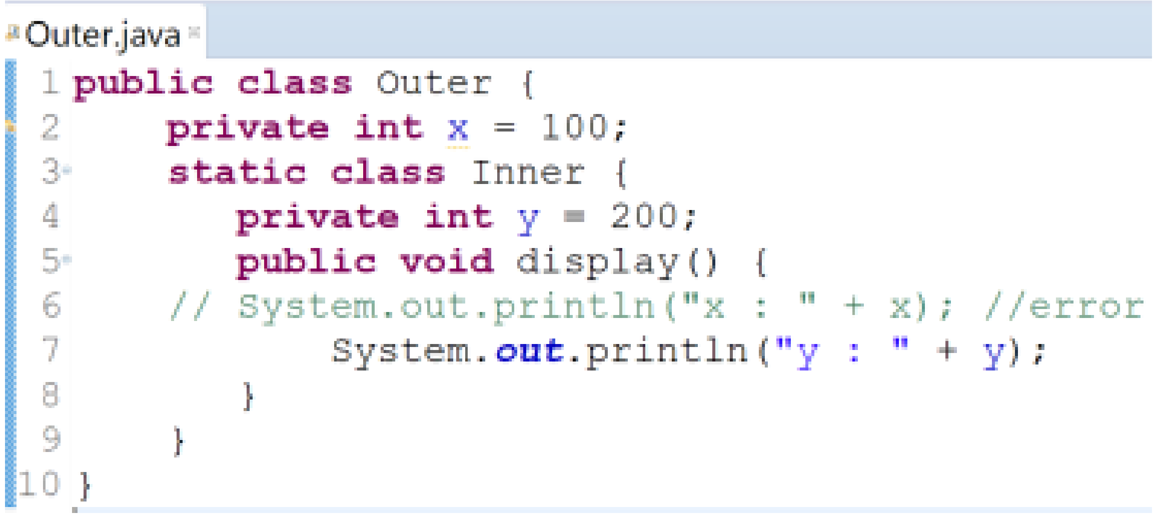
```
public class Outer {
    private int x = 100;
    static class Inner {
              private int y = 200;
              public void display() {
                    // System.out.println("x : " + x); //error
                  System.out.println("y : " + y);
             }
     }
}
```

Practice – Code and Result

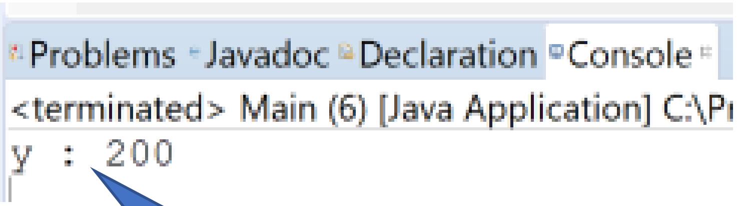


```
public class Main {
    public static void main(String[] args) {

        Outer.Inner testInner= new Outer.Inner();
        testInner.display();
    }
}
```



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



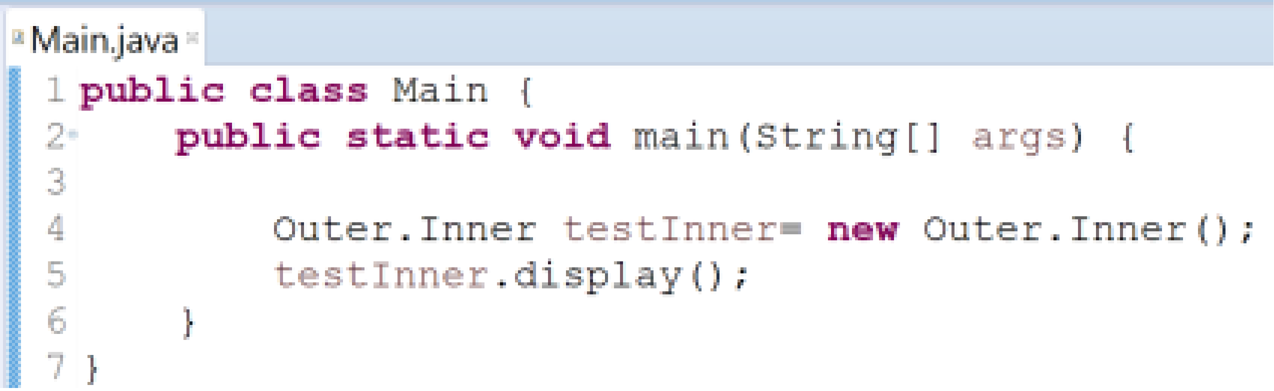
```
<terminated> Main (6) [Java Application] C:\Pr
y : 200
```

Result

Summary

➢Static Inner Class

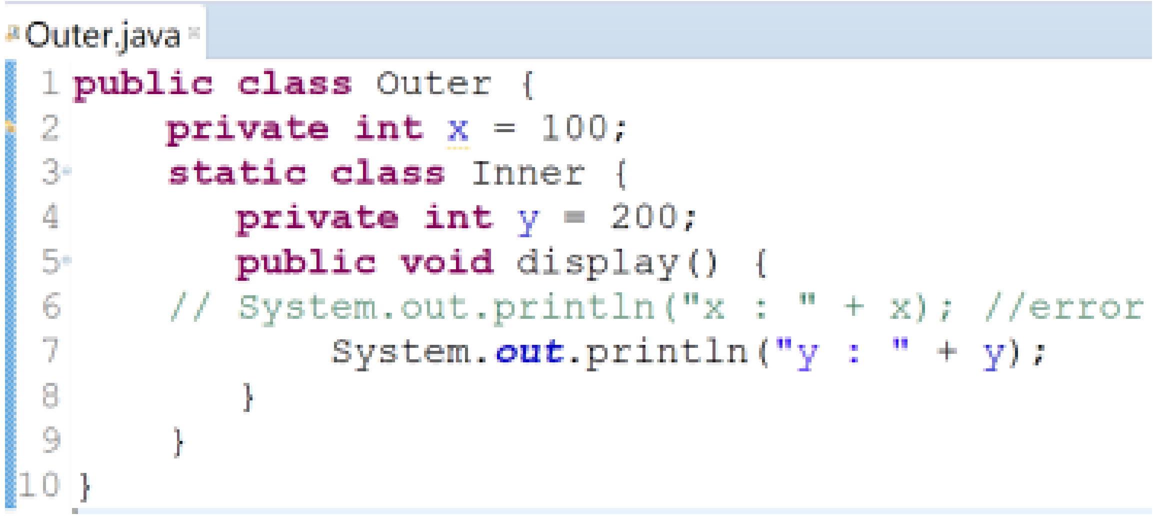
- ❑Using the **static** keyword in front of Class name
- ❑Not dependent on a specific instance
  - ❖Independently create object
- ❑Only can access static member in Outer class



Main.java

```
public class Main {
    public static void main(String[] args) {

        Outer.Inner testInner= new Outer.Inner();
        testInner.display();
    }
}
```



Outer.java

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