Inner Class

What is inner class?

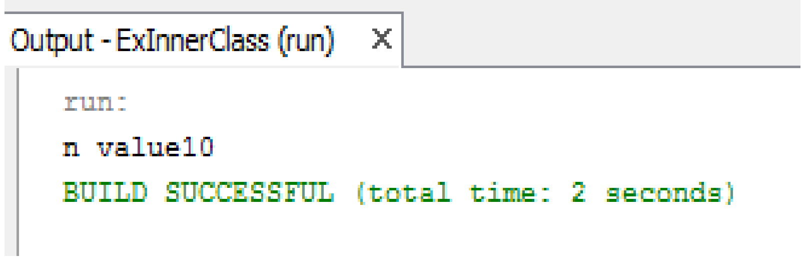
- **Java inner class** or nested class is a class which is declared inside the class or interface.
- Used to logically group classes and interfaces in one place
- They more readable and maintainable.
- Inner class can access all the members of outer class including private data members and methods.

Syntax for Inner Class

```
class Java_Outer_class
{
   //code
   class Java_Inner_class
  {
    //code
  }
}
```

Example1 for (Private)Member Inner Class

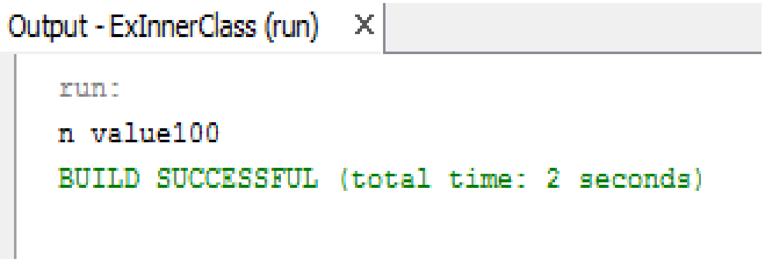
```
class Outer
{   int n;
   private class Inner
   {  void print()
      {         System.out.println("n value"+n);        }
   }
   void display()
   {       Inner in=new Inner();       in.print();    }
}
public class ExInnerClass
{
     public static void main(String[] args)
  {
       Outer out=new Outer();
        out.n=10;
        out.display();
    }
}
```



```
run:
n value10
BUILD SUCCESSFUL (total time: 2 seconds)
```

Example 2 for (Public) Member inner class

```
class Outer
{
  int n=100;
  public class Inner
  { void print()
     { System.out.println("n value"+n);      }
  }
  void display()
  {  Inner in=new Inner(); in.print();    }
}
public class ExInnerClass1
{
public static void main(String[] args)
{
      //Instance for Outer class
      Outer out=new Outer();
     //Instance for Inner class
      Outer.Inner in =out.new Inner();
      in.print();
   }
}
```



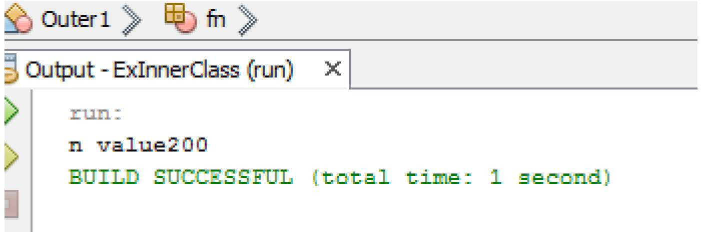
```
run:
n value100
BUILD SUCCESSFUL (total time: 2 seconds)
```

Method-Local Inner Class

- In Java, we can write a class within a method and this will be a local type.
- Like local variables, the scope of the inner class is restricted within the method.
- Method-local inner class can be instantiated only within the method where the inner class is defined.

Example3 for Method-Local Inner Class

```
class Outer1
{
   void fn()
   {    class Inner1
      {
         int n=200;
         void print()
         {   System.out.println("n value"+n);      }
      }
      Inner1 in1=new Inner1();
      in1.print();
   }
}
public class ExInnerClass2
{
   public static void main(String args[])
   {
   Outer1 outer=new Outer1();
   outer.fn();
   }
}
```



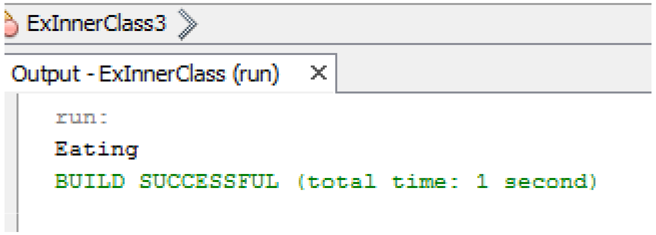
```
run:
n value200
BUILD SUCCESSFUL (total time: 1 second)
```

Anonymous Inner Class

- A class that have no name is known as anonymous inner class in java.
- It is used if necessary to override method of class or interface.
- Java Anonymous inner class can be created by two ways:
  - Class (may be abstract or concrete).
  - Interface

Example4 for Anonymous Inner Class

```
abstract class Person{
 abstract void eat();
}
class ExInnerClass3 {
 public static void main(String args[])
{
 Person p=new Person(){
 void eat(){System.out.println(“Eating");;}
 };
 p.eat();
 }
}
```



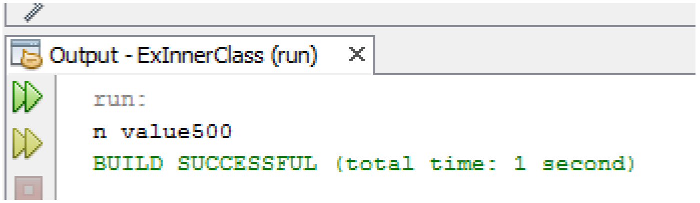
```
run:
Eating
BUILD SUCCESSFUL (total time: 1 second)
```

Static Inner Class

- Inner class can be static class also.
- It cannot access non-static data members and methods.
- It can be accessed by outer class name.
- It can access static data members of outer class including private.
- Static nested class cannot access non-static (instance) data member or method.

Example5 for static inner class

```
class Outer4
{
    static int n=500;
    static class Inner4
    {
        void print()
        {
            System.out.println("n value"+n);
        }
    }
}
public class ExInnerClass4
{
    public static void main(String args[])
    {
   Outer4.Inner4 out=new Outer4.Inner4();
   out.print();
    }
}
```



```
run:
n value500
BUILD SUCCESSFUL (total time: 1 second)
```