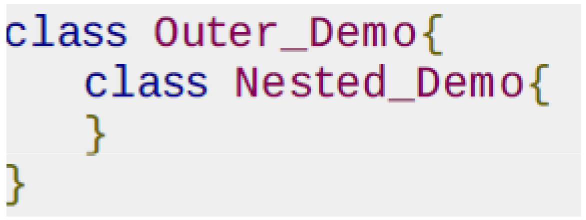
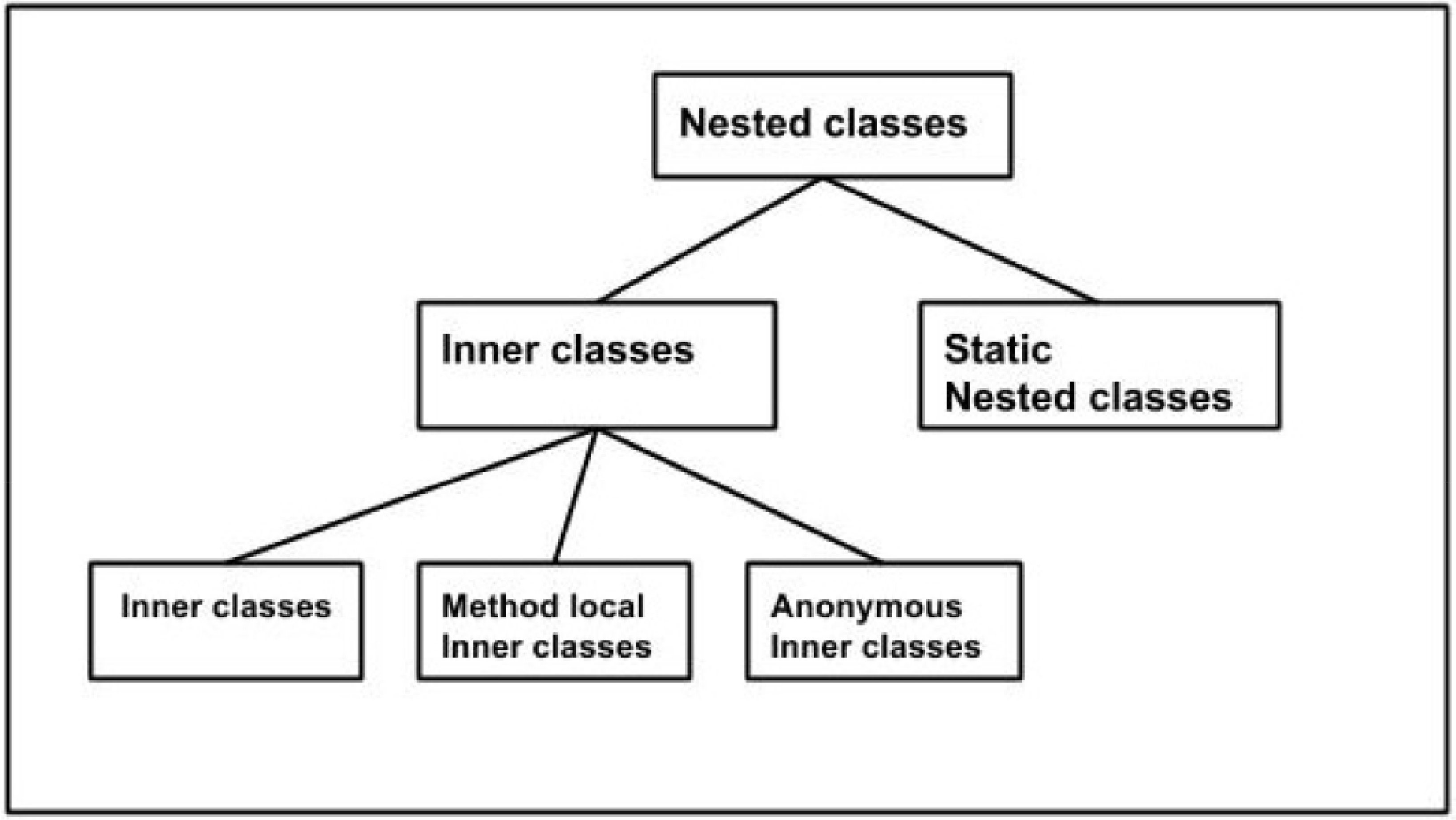
Inner classes

Nested Classes

- In Java, just like methods, variables of a class too can have another class as its member.
- Writing a class within another is allowed in Java.
- The class written within is called the **nested class**, and the class that holds the inner class is called the **outer class.**



```
class Outer_Demo{
   class Nested_Demo{
   }
}
```


Nested classes
Inner classes
Static Nested classes
Inner classes
Method local Inner classes
Anonymous Inner classes

Non static nested class – Inner class

```
class Outer_Demo{
   int num;
   //inner class
   private class Inner_Demo{{
      public void print(){
      System.out.println("This is an inner class");
      }
   }
   //Accessing he inner class from the method within
   void display_Inner(){
      Inner_Demo inner = new Inner_Demo();
      inner.print();
   }
}

public class My_class{
   public static void main(String args[]){
      //Instantiating the outer class
      Outer_Demo outer=new Outer_Dem();
      //Accessing the display_Inner() method.
      outer.display_Inner();
   }

}
```

If you compile and execute the above program, you will get the following result.

```
This is an inner class.
```

Accessing the Private Members

```
class Outer_Demo {
   //private variable of the outer class
   private int num= 175;
   //inner class
   public class Inner_Demo{
      public int getNum(){
         System.out.println("This is the getnum method of the inner class");
         return num;
      }
   }
}

public class My_class2{
   public static void main(String args[]){
      //Instantiating the outer class
      Outer_Demo outer=new Outer_Demo();
   //Instantiating the inner class
      Outer_Demo.Inner_Demo inner=outer.new Inner_Demo();
      System.out.println(inner.getNum());
   }
}
```

If you compile and execute the above program, you will get the following result.

```
The value of num in the class Test is: 175
```

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.
- A method-local inner class can be instantiated only within the method where the inner class is defined.

Method-local Inner Class

```
public class Outerclass{

   //instance method of the outer class
   void my_Method(){
   int num=23;

   //method-local inner class
   class MethodInner_Demo{
      public void print(){
         System.out.println("This is method inner class "+num);
      }
   }//end of inner class

   //Accessing the inner class
   MethodInner_Demo inner=new MethodInner_Demo();
   inner.print();
   }

   public static void main(String args[]){
      Outerclass outer =new Outerclass();
      outer.my_Method();
   }
}
```

If you compile and execute the above program, you will get the following result.

```
This is method inner class
```

Anonymous Inner Class

- An inner class declared without a class name is known as an **anonymous inner class.**
- **In case of** anonymous inner classes, we declare and instantiate them at the same time.
- Generally **they are used** whenever you need to override the method of a class or an interface.

Anonymous Inner Class

```
abstract class AnonymousInner{
   public abstract void mymethod();
}

public class Outer_class {
   public static void main(String args[]){
   AnonymousInner inner= new AnonymousInner(){
      public void mymethod(){
         System.out.println("This is an example of anonymous inner class");
      }
   };
   inner.mymethod();
 }
}
```

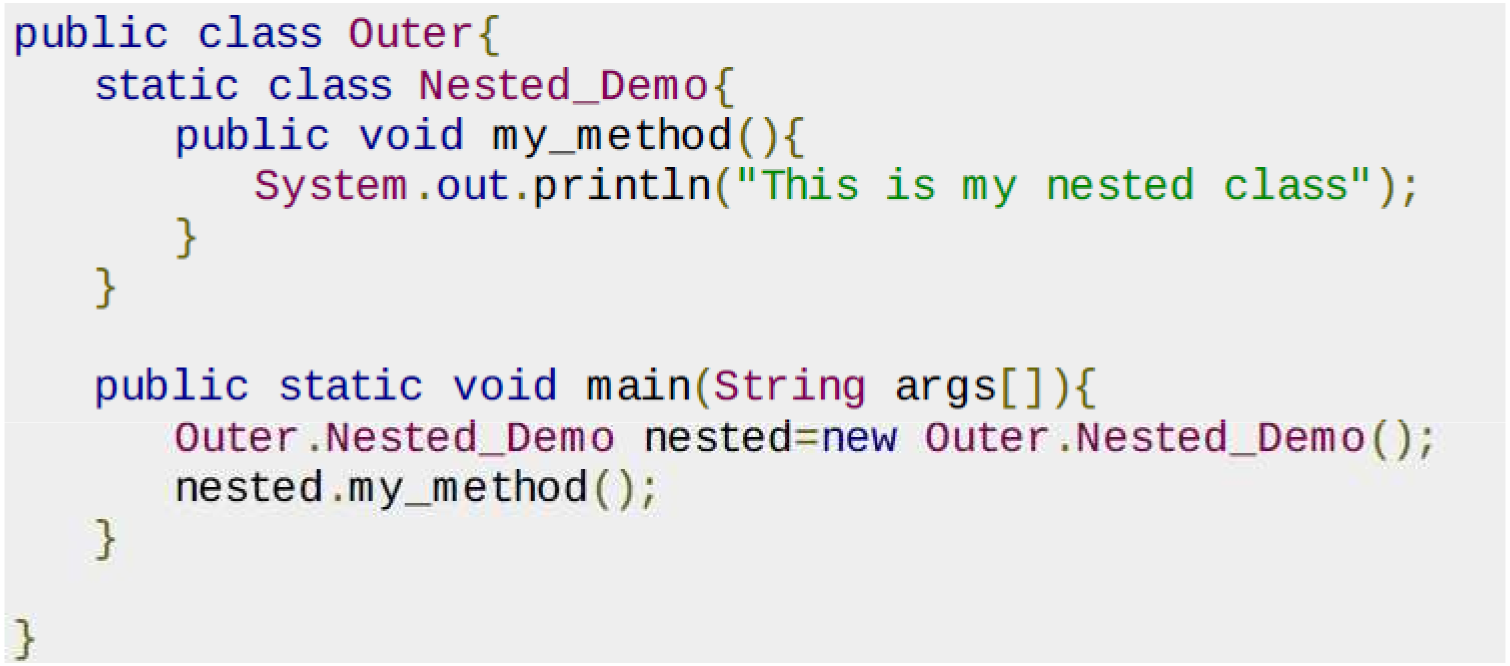
If you compile and execute the above program, you will get the following result.

```
This is an example of anonymous inner class
```

Static Nested class

- A static inner class is a nested class which is a static member of the outer class.
- It can be accessed without instantiating the outer class, using other static members.
- Just like static members, a static nested class does not have access to the instance variables and methods of the outer class.

Static Nested class



```
public class Outer{
   static class Nested_Demo{
      public void my_method(){
         System.out.println("This is my nested class");
      }
   }

   public static void main(String args[]){
      Outer.Nested_Demo nested=new Outer.Nested_Demo();
      nested.my_method();
   }

}
```