**Static Nested , Inner and Local classes**

**Static Nested Class**

* Defining a class within another class is known as nested classes. A static class which is created inside a class is called static nested class. A static nested class cannot access non-static data members and methods. It can only access the static members of its outer class. It does not have access to the instance members of the Outer class.
* A static nested class cannot refer directly to instance variables or methods defined in its Outer class. It can use them only through an object reference using the following syntax OuterClass.StaticNestedClass.

class Stuinfor{

  static String stuname;

  static int age;

  static class Studet {

     public void Showdet(String name, int a){

         stuname=name;

         age=a;

         System.out.println("The student name:"+ name);

         System.out.println("The student age:"+

age);

      }

  }

}

public class Staticnested {

 public static void main(String args[]){

     Stuinfor.Studet in=new Stuinfor.Studet();

     in.Showdet("Richard",15);

 }

}

**Inner Class (Non-Static Nested Class)**

* A non-static nested class is called as Inner class in Java. Nested Inner class can access private instance variable of outer class. We cannot define any static method inside the nested inner class because an inner class is implicitly associated with an object of its outer class.
* The nested inner class exists within the outer class, and hence to instantiate an inner class, we must first instantiate the outer class. Then, create the inner object within the outer object using the following syntax as OuterClass.InnerClass innerobject =outerobject.new InnerClass()

class StudentClass{

 int extMark, total;

String rollNo;

 final int internalMark=18;

  class InnerStudent{

         public void StudentMarks(String rNo,int externalMark){

             rollNo=rNo;

             extMark=externalMark;

             total=externalMark+internalMark;

             System.out.println("Student Rollnumber:" + rNo);

             System.out.println("Total marks: " + total);

             }

      }

}

public class InnerClass {

 public static void main(String args[]){

  StudentClass outerClass=new StudentClass();

  StudentClass.InnerStudent InnerClass=outerClass.new InnerStudent();

  InnerClass.StudentMarks("SCS1024",68);

}

}

**Local Inner Class**

* The Local Inner class is a class which is declared within a method body of an outer class. The scope of the Local inner class is restricted to the block/method where they are defined. This Local inner class cannot be instantiated from outside the block/method where it is created. A local class can access the members of its outer class.
* To invoke the methods of local inner class, we must instantiate this local inner class inside the method itself. The outer class methods can be invoked only by creating objects for the outer class.

class Localinnerclass {

 int b=20;

 public void outerMethod(){

 int a=10;

     class Innerclass{

         public void innerMethod(){

             System.out.println("You are inside Inner Local class");

             System.out.println("Sum of a + b is = " +(a+b));

         }

     }

     Innerclass ob = new Innerclass();

     ob.innerMethod();

 }

 public static void main(String args[]){

     Localinnerclass ob1= new Localinnerclass();

     ob1.outerMethod();

 }

}