Course code : **CSE2005**

Course title : **Object Oriented Programming**

Inner class

Objectives

This session will give the knowledge about

- Inner class
- Static inner class
- Local inner class
- Anonymous inner class

Inner Class

- Like variables and methods, class can also be defined inside a class.
- An inner class is a class defined inside the scope of another class.
- Classes that were covered so far were top-level classes.
- The class inside which the inner class is defined is called outer class.
- Inner class can access even the private members of the outer class. Similarly outer class can also access the private members of inner class.
- Similar to inner class, inner interface can also be created.

Inner class example

- Inside the Outer class scope, Inner class name is same as what is declared in Outer class.

- Outside the Outer class scope, Inner class name is combination of Outer class name and Inner class name and a dot (.) separating them. Inner class name can be same as outer class name.

- The name of the inner class’s .class file name:

  - OuterClass$InnerClass.class

Types of Inner Class

- Member class
  - Static Inner Class/ Top-Level nested classes
  - Non Static Inner Class
- Local Inner Class
- Anonymous Class
- Non Static Inner Class, Local Inner Class, Anonymous Class are generally called inner class.
- Static inner class are considered to be top-level class.

Types of Inner Class

- Non static inner class object cannot be created without a outer class instance.
- The private fields and methods of the member classes are available to the enclosing class and other member classes
- All the private fields and methods of the outer classes are also available to inner class.
- Non-static inner class cannot have static members.
- Other modifier applicable here are abstract, final, public, protected, private

Non Static inner class : Syntax

Class definition:

```
class Outer{
   class Inner{
   }
}
```

Instance creation inside the Outer class

```
Inner innerObj=new Inner();
```

Instance creation outside the Outer class

```
Outer outerObj=new Outer();
Outer.Inner innerObj=outerObj.new Inner();
```

Creating instance outside the outer class

Outside outer class non-static inner class creation requires outer class instance also.

There are 2 ways to do this.

If you don’t need outer class instance , then create it like line 1,

If you need outer class instance or already have one, create it like line 2

```
class A
{
C.B b= new C(). new B(); //line 1
C c= new C();
C.B b1= c. new B(); // line 2
}
```

If Inner class was defined in a package say p, then it can be created using the syntax: new p.C(). new B();

Non Static inner class : Example

```
package vit.demo;
class Person{
   String name;
   Person(String name){
            this.name=name;
   }
   class Address{
            String street;
            public Address(String street) {
                        this.street = street;
            }
   }
}
```

```
public class Main {
       public static void main(String s[]){
            Person person=new Person("john");
            Person.Address add=person.new
       Address("kk nagar");

            System.out.println(person.name);
            System.out.println(add.street);
       }
}
```

Quiz

```
class C {
  private int i;
  static private int k;
  void m() {
        B b = new B();
        b.j = 10;
  }
  class B {
        private int j;
        static private int l;
        void m() {
              i = 10;
              k = 15;
              j = 12;
        }
    }
}
```

Quiz

```
class C {
   int i;
   static int k;
   void m() {
            B b = new B();
            b.j = 10;
   }
   class B {
            int j, q;
            void m() {
                        i = 10; k = 15;  j = 12;
            }
   }
}

public class Main {
   public static void main(String s[]) {
            C c=new C();
            C.B b=c.new B();
            System.out.println(b.j);
   }
}
```

Quiz

```
package vit.demo;

class C {
  int i=10;

  class B {
        int i=i+10;
  }
}
```

```
public class Main {
    public static void main(String s[])
    {
        C c = new C();
        C.B b = c.new B();
        System.out.println(b.i);
    }
}
```

Name conflict

If the name of the members in Outer class and inner class are same, then how to refer to the name of the outer class member in the inner class?

This can be done using Outer class name dot (.) this dot (.) member name.

```
class C {
  int i=10;
  class B {
        int i=C.this.i+10;
  }
}
```

Static Inner Class

- A static inner class is a class that’s a static member of the outer class
- It can access only all static members of the outer class.
- But like main method, instances of outer class can be created inside static inner class and using this private members can be accessed.
- It is created without an instance of the outer class unlike the regular inner classes. static classes are also called top-level nested classes.
- Other modifier applicable to member classes abstract, final, public, protected, private

Static inner class : Syntax

Class definition:

```
public class Outer{
   public static class Inner{
   }
}
```

Instance creation inside the Outer class

```
Inner innerObj=new Inner();
```

Instance creation outside the Outer class

```
Outer outerObj=new Outer();
Outer.Inner innerObj=Outer.new Inner();
```

Local Inner class

- An inner class that is defined inside a method is called local inner class (or method local inner class).
- A local inner class can be instantiated only by the method which defined it.
- Therefore no access specifier is applicable for the local inner class declaration. Only abstract and final modifiers are allowed.
- Also like other inner classes, local inner class can access all the members of the outer class including private members.

Local Inner class

- Apart from the above, the local inner class can also access local variables which are final.

```
class C {
        int i=10;
        void process(){
              class B{
                            B(){
                                          System.out.println("I am local inner class");
                            }
              }
              B b=new B();
        }
}
```

Anonymous Inner Classes

- Inner class without a class name is an anonymous inner class.
- Allows creation of one time use object !
- Anonymous inner class can be created either inside a method or outside a method. It is implicitly final.
- No modifier is allowed anywhere in the class declaration
- Also declaration cannot have an implements or extends clause.
- No constructors can be defined.

Anonymous Inner Classes

- An anonymous inner class is either inherited from an interface or from a class and so polymorphism is applicable. It cannot inherit from more than one class directly.

```
Test t = new Test()     // Test can be interface,abstract/concrete class
{
   public void test_method()          // data members and methods
   {
       ........
    }
}
```

Anonymous Inner class - Example

```
package vit.demo;
interface B{
   public void display();
}

class C {
   int i=10;
   B b=new B(){
            public void display(){
                  System.out.println("i am anony");
            }
   };
}

public class Main {
   public static void main(String s[]) {
            new C().b.display();
   }
}
```

Quiz: Guess the output

```
package vit.demo;

public class Main {
   Main() {
            System.out.print("E");
   }
   static class Z {
            Z() {
                        System.out.print("Z");
            }
   }
      public static void main(String args[])
      {
               new Main.Z();
      }
   }
```

Quiz

Which variable (in red) cannot be substituted for “???” without causing a compile-time error?

```
public class Main {
  private static String s1 = "s1";
  final String s2 = "s2";

  Main() {
        new Z("s5", "s6");
  }
```

Quiz

```
    static class Z {
        final String s3 = "s3";
        static String s4 = "s4";

        Z (final String s5, String s6) {
            System.out.print(???);
            }
    }
    public static void main(String args[]) {
        new Main();
    }
}
```

Quiz: Guess the output

```
public class Main {
  public void m1() {
        Z.m1();
  }

  private static class Y {
      private static void m1() {
        System.out.print("Y.m1 ");
      }
  }

  private static class Z {
      private static void m1() {
            System.out.print("Z.m1 ");
            Y.m1();
      }
  }

  public static void main(String[] args) {
      new Main().m1();
  }
}
```

Quiz

Compile-time errors are generated at which lines?

```
class Outer {
  static class StaticNested {
      static final int a = 25; // 1
      static final int b; // 2
      static int c; // 3
      int d; // 4
      static { b = 42; } // 5
  }
```

```
  class NonStaticInner {
      static final int e = 25; // 6
      static final int f; // 7
      static int g; // 8
      int h; // 9
      static { f = 42; } // 10
  }
}
```

Summary

We have discussed about

- Inner class
- Static inner class
- Local inner class
- Anonymous inner class