CSE1007 – JAVA PROGRAMMING

Module - 2

Inner Class

Inner Classes

- Class within another class is called an **inner class** or nested class
- Inner classes are the nested classes
- classes that are defined inside other classes
- The scope of a nested class is bounded by the scope of its enclosing class.
- One class can be defined within another class and the class may be non-static, static class.
- **Syntax:**

```
[access_modifier] class Outerclassname
{
  //code
    ….
    …
  [access_modifier] class Innerclassname
    {
        //code
    }
}
```

Inner Classes

Properties of inner classes:

- Inner class has access to the member, including private members, of the class in which it is nested.
- The enclosing class does not have access to the member of the nested class.
- If inner class has the same variable name, then the outer class variable can be accessed like

outerclassname.this.variable_name

Types of inner classes:

- **Non-static inner classes**
- **Static inner classes**
- **Method local inner classes**
- **Anonymous inner classes**

Inner Classes

1.Non-static inner Classes:

- Non-static nested classes
- Creating an inner class is quite simple

Inner Classes

1.Non-static inner Classes:

Non-static nested classes

Creating an inner class is quite simple

Syntax:

```
access_modifier class Outerclass
{
    //code
     class innerclassname
     {
         //code
     }
    //code
}
```

Example:

```
Class A
{
    ….
     class B
      {
          ……
      }
    ……
}
```

Inner Classes

Example:

```
package CSE1007_MODULE2;
class Outer
{
    int outer_x = 100;
    void test()
    {
        Inner inner = new Inner();
        inner.display();
    }
    class Inner
    {
        void display()
        {
            System.out.println("outer_x = " + outer_x);
        }
    }
}
```

```
public class Innerclass1
{
  public static void main(String args[])
  {
      Outer outer = new Outer();
      outer.test();
  }
}
```

```
outer_x = 100
BUILD SUCCESSFUL (total time: 0 seconds)
```

Inner Classes

- Classes cannot be created as private. But, the inner classes can be created as private
- So, it cannot be accessed from an object outside the class.

Inner Classes

Ex: Non-static inner class

```
class A
{
    private class B
     {
         public void print()
         {
            System.out.println("Inner class");
         }
     }
    void display()
    {
       B b=new B();
       b.print();
    }
}
```

```
public class innerclassprivate
{
    public static void main(String args[])
    {
       A a =new A();
       a.display();
    }

}
```

```
Inner class
BUILD SUCCESSFUL (total time: 0 seconds)
```

Inner Classes

2.Static inner classes:

- It has the static modifier
- It cannot refer to members of its enclosing class directly
- It must access the members of its enclosing class through an object
- Static members of outer class are visible to the static inner class
- The non-static members of the outer class are not visible to inner class

Inner Classes

Syntax:

```
[access_modifier]  class Outerclass
{
   //code
   static class Innerclass
     {
          //code
     }
}
```

Example:

```
class A
{
      ….
      static class B
        {
            ……
        }
      ……
}
```

Inner Classes

Ex:Static inner class

```
package CSE1007_MODULE2;
public class innerclassstatic
{
   static class B
    {
         public void print()
        {
            System.out.println("Inner class");
        }
    }
    public static void main(String args[])
    {
      B b=new B();
      b.print();
    }
}
```

```
Inner class

BUILD SUCCESSFUL (total time: 0 seconds)
```

Inner Classes

3.Method local inner classes:

Class is defined inside the body of the method/function

Syntax:

```
[access_modifier] class Outerclass
{
    //code
     [access_modifier] return_type  fnname(arguments)
      {
           class innerclass
           {
                //code
           }
      }
}
```

Example:

```
class A
{
     ….
     void AB()
      {
           class B
           {
                ……
           }
          ……
      }
}
```

Inner Classes

Ex:Method local inner class

```
package CSE1007_MODULE2;
class D
{
   void mtdA()
    {
       class B
       {
          public void print()
         {
             System.out.println("Inner class");
         }
       }
      B b=new B();
      b.print();
    }
}
```

```
public class innerclassmethodlocal
{
    public static void main (String args[])
    {
       D a =new D();
       a.mtdA();
    }

}
```

```
Inner class
BUILD SUCCESSFUL (total time: 0 seconds)
```

Inner Classes

4. Anonymous classes:

- This class is the local class without any name
- This is one-shot-class created exactly where needed
- The anonymous class is created in following situations
  - » When the class has very short body
  - » Only one instance of the class is needed
  - » Class is used immediately after defining it

Syntax:

```
A a=new A()
{
  ……
  …….
};
```

Inner Classes

Ex:Anonymous inner class

```
package CSE1007_MODULE2;
class innerclassannoymous
{
    public static void main(String args[])
    {
      A a=new A()
      {
        public void display()
        {
          System.out.println("Annoymous");
        }
      };
      a.display();
    }
}
```

```
Annoymous

BUILD SUCCESSFUL (total time: 0 seconds)
```