

Java means DURGA SOFT..

CORE JAVA

Material



India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Inner Classes

- Declaring a class inside a class is called as an Inner Class.
- In Java applications, Inner classes are able to provide the following Advantages.

- 1.Modularity
- 2.Abstraction
- 3.Share-ability
- 4.Security
- 5.ReUseAbility

1.Modularity:

- If we declare an Inner class inside a class then that inner class is treated as a module of that entity class.

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

JAVA MEANS DURGASOFT

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS

#202 2nd FLOOR
www.durgasoft.com

040-64512786
+91 9246212143
+91 8096969696

Ex:

```
class Account{
    classStudentAccount{
    }

    classEmployeeAccount{
    }

    classLoanAccount{
```

```
}  
  
}
```

2.Abstraction:

- If we declare any variable or method inside an inner class then that variable and method will have scope upto the respective inner class only,which are not available to outside of that inner class.

3.Security:

- In Java applications,private keyword is able to improve security.In Java applications,it is not possible to declare a class as private but it is possible to declare an inner class as private.Therefore Inner classes are able to improve security.

4.Share-ability:

- In Java applications,static keyword is able to improve share-ability.
- In Java applications,it is not possible to declare a class as static but it is possible to declare an inner class as static.Therefore inner classes are able to improve share-ability.

5.ReUseAbility:

- In Java applications,inheritance is able to improve ReuseAbility.In Java applications,it is possible to extend one inner class to another inner class,sothat,inner classes are able to improve ReuseAbility.

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

Types of Inner classes:

There are four types of Inner classes in Java.

- 1.Member Inner class
- 2.Static Inner class
- 3.Method Local Inner class
- 4.Anonymous Inner class.

1.Member Inner class:

- Declaring a non-static class inside a class is called as Member Inner class.

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



- If we want to access members of member Inner class then we have to create an Object for the respective member inner class. To create object for member inner class, we have to use the following syntax.

Outer.Innerref_Var=new Outer.newInner();

- In the case of member Inner classes, by using outer class reference variable we are able to access only outer class members, it is not possible to access inner class members.
- In the case of member Inner classes by using Inner class reference variables we are able to access only inner class members, it is not possible to access outer class members.
- In the case of member inner classes, all the outer class members are available to inner classes automatically but all the inner class members are not available to outer classes.
- In general, inner classes are not allowing static declarations, but inner classes are able to allow 'static' keyword along with 'final' keyword.

www.durgasoftonlinelearning.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinelearning@gmail.com

Example programme on Member Inner class:

```
class A{
int i=10;

void m3(){

System.out.println("m3-A");

//System.out.println(j); ---->error

}
```



```
class B{  
    int j=20;  
    //static int k=30; --->error  
    static final int k=30;  
    void m1(){  
        System.out.println("m1-B");  
        System.out.println(i);  
    }  
    void m2(){  
        System.out.println("m2-B");  
    }  
}  
  
class Test{  
    public static void main(String args[]){  
        A.B ab=new A().new B();  
        ab.m1();  
        ab.m2();  
        //ab.m3(); --->error  
        A a=new A();  
        a.m3();  
        //a.m1(); --->error  
        //a.m2(); --->error  
    }  
}
```

In the case of member inner classes, we are able to define inheritance relation in the following cases:

Diagram(OOPS--INNERCLASSES1.png)

Q)Is it possible to provide an interface inside a class?

- Ans: Yes, it is possible to provide an interface inside a class but the respective implementation class must be provided in the same outer class.

Example programme on, to provide an interface inside a class :

```
class A{  
    interface I{  
        void m1();  
        void m2();  
    }  
    class B implements I{  
        public void m1(){  
            System.out.println("m1-B");  
        }  
        public void m2(){  
            System.out.println("m2-B");  
        }  
    }  
}
```



```

class Test{

public static void main(String args[]){

A.I ai=new A().new B();

ai.m1();

ai.m2();

}

}

```

2.Static Inner class:

- Declaring Static class inside a class is called as Static Inner class.

```

class Outer{

static class Inner{

}

}

```

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

JAVA MEANS DURGASOFT

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS

#202 2nd FLOOR
www.durgasoft.com

040-64512786
+91 9246212143
+91 8096969696

- If we want to access the members of static inner classes then we have to create object for the static inner class.

Outer.Innerref_Var=new Outer.Inner();

- In general inner classes will allow all the members of outer classes directly but static inner classes are able to allow only static members of the outer class, it will not allow non-static members of the outer class.

- In general, inner classes will not allow static declarations but static inner classes are able to allow the static declarations.

Example programme on Static Inner classes:

```
class A{
    int i=10;
    static int j=10;
    static class B{
        void m1(){
            System.out.println("m1-B");
            //System.out.println(i); ---->error
            System.out.println(j);
        }
    }
}
```

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

```
void m2(){
    System.out.println("m2-B");
}
static void m3(){
    System.out.println("m3-B");
}
}
```

```

class Test{

public static void main(String args[]){

A.B ab=new A.B();

ab.m1();

ab.m2();

A.B.m3();

}

}

```

Q) Is it possible to provide a class inside an Interface?

- **Ans:** Yes, it is possible to declare a class inside an interface. If we declare a class inside an interface then that class is becoming as static inner class, where we can access the members like static inner class members.

```

interface I{

class A{

void m1(){

System.out.println("m1-A");

}

}

```



```

void m2(){

System.out.println("m2-A");

}}

class Test{

```

```

public static void main(String args[]){

    I.A ia=new I.A();

    ia.m1();

    ia.m2();

}

}

```



3.Method local Inner Class:

- Declaring a class Inside a method is called as Method Local Inner class.
- If we declare a class inside a method then that class is having scope upto the respective method.
- In the case of method local Inner classes,we have to create object for the Inner class and we have to access members of the inner class inside the respective method only.

Example Programme on Method local Inner Class:

```

class A{
    void m1(){
        class B{
            void m2(){
                System.out.println("m2-B");
            }
        }
        void m3(){

```

```

System.out.println("m3-B");

}

}

B b=new B();

b.m2();

b.m3();

}

}

class Test{

public static void main(String args[]){

A a=new A();

a.m1();

}}

```



4. Anonymous Inner class:

- In general, in Java applications, if we declare any abstract class with abstract methods then we have to take a sub class for the abstract class, where we have to provide implementation for all the abstract methods.
- In the case of interfaces, we have to take an implementation class and we have to provide implementation for all the abstract methods declared in the interface.
- In the above two cases, subclasses or implementation classes may allow their own methods and if we create object for sub class or implementation class then that object may have the respective sub class identity or implementation class identity.

- In Java applications, if we want to provide implementations for abstract classes and interfaces without allowing extra methods and if we want to create object with abstract class identity and with interface identity then we have to use "Anonymous Inner Class".

Syntax:

```
abstractclassName / Interface Name
{
    ---abstract methods-----
}
class Outer{
    Name ref_Var=new Name(){
        -----impl for abstract methods-----
    };
}
```



LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

JAVA MEANS DURGASOFT

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

<small>AN ISO 9001:2008 CERTIFIED</small> DURGA <small>SOFTWARE SOLUTIONS</small>	#202 2nd FLOOR www.durgasoft.com	040-64512786 +91 9246212143 +91 8096969696
--	--	---

Example programme on Anonymous Inner class for Interface:

```
interface I{
    void m1();
    void m2();
}
class A{
    I i=new I(){
```



```

public void m1(){
    System.out.println("m1-A");
}

public void m2(){
    System.out.println("m2-A");
}

};

}

class Test{

    public static void main(String args[]){
        A a=new A();
        a.i.m1();
        a.i.m2();
    }
}
    
```

www.durgasoftonlinetraining.com



Online Training

Pre Recorded Video

Classes Training

Corporate Training

Ph: +91-8885252627, 7207212427
+91-7207212428

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

Example programme on Anonymous Inner class for Abstract Class:

```
abstract class A{  
    void m1(){  
        System.out.println("m1-A");  
    }  
    abstract void m2();  
    abstract void m3();  
}  
class B{  
    A a=new A(){  
        void m2(){  
            System.out.println("m2-AIC");  
        }  
        void m3(){  
            System.out.println("m3-AIC");  
        }  
    };  
}
```

www.durgajobs.com
Continuous Job Updates for every hour

Fresher Jobs	Govt Jobs	Bank Jobs
Walk-ins	Placement Papers	IT Jobs
Interview Experiences		

Complete Job information across India

```

class Test{

public static void main(String args[]){

    B b=new B();

    b.a.m1();

    b.a.m2();

    b.a.m3();

}

}

```



Utilization of Anonymous Inner class:

- In Java applications, when we have any requirement to pass interface or abstract class references as parameters to the methods then we have to pass anonymous Inner class as parameter to the methods instead of taking implementation classes and their references values.

Ex:

```

class MyFrame extends Frame{

    MyFrame(){

        this.addWindowListener(new WindowAdapter(){

            public void windowClosing(WindowEvent we){

                System.exit(0);
            }
        });
    }
}

```

```

    }
    });
} }

```

Example programme on Utilization of Anonymous Inner class:

```

interface I{
    void m1();
}
class B{
    void m2(I i){
        System.out.println("m2-B");
        i.m1();
    }
}
class Test{
    public static void main(String args[]){
        B b=new B();
        b.m2(new I()
        {
            public void m1(){
                System.out.println("m1-AIC");
            }
        });
    }
}

```

NOTE:Passing Anonymous Inner class as parameter to the methods is suggestable only in the case of less no.of abstract methods in side the interface.If the interface

contains more no.of methods then it is suggestable to use implementation class approach.

A vertical banner advertisement for Durga Software Solutions. It features a dark blue background with a yellow central area containing red and white text. The text lists various technologies and services offered, including Java, .NET, Testing Tools, Oracle, D2K, MSBI, Share Point, Hadoop, Android, C, C++, DS, UNIX, and CRT & Aptitude Training. It also mentions ISO 9001:2008 certification and provides contact information for their Hyderabad office.

LEARN FROM EXPERTS ...

COMPLETE JAVA
CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...

COMPLETE .NET
C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF

TESTING TOOLS
MANUAL + SELENIUM

ORACLE | D2K

MSBI | SHARE POINT

HADOOP | ANDROID

C, C++, DS, UNIX

CRT & APTITUDE TRAINING

AN ISO 9001:2008 CERTIFIED
DURGA
Software Solutions®

202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,
9246212143, 8096969696

www.durgasoft.com