

Member Modifiers

```
55 member modifiers:  
56 -----  
57 1. public members:  
58 -----  
59 |
```

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir On 20-02-2018  
1 package pack1;  
2 class A  
3 {  
4     public void m1()  
5     {  
6         System.out.println("Hello it is public method")  
7     }  
8 }
```

The below is an error. The class A is not public and Class B is in diff package, How can it import a non public class in diff package

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir On 20-02-2018  
1 package pack2;  
2 import pack1.A;  
3 class B  
4 {  
5     public static void main(String[] args)  
6     {  
7         A a = new A();  
8         a.m1();  
9     }  
10 }  
11  
D:\durgaclasses>javac -d . B.java  
B.java:2: error: A is not public in pack1; cannot be accessed from outside package  
import pack1.A;  
          ^  
B.java:7: error: cannot find symbol  
        A a = new A();  
        ^  
      symbol:   class A  
      location: class B  
B.java:7: error: cannot find symbol  
        A a = new A();  
        ^  
      symbol:   class A  
      location: class B  
3 errors
```

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir On 20-02-2018  
1 package pack1;  
2 public class A  
3 {  
4     public void m1()  
5     {  
6         System.out.println("Hello it is public method");  
7     }  
8 }
```

```
C:\Windows\system32\cmd.exe
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 20-02-2018

D:\durgaclasses>javac -d . A.java
D:\durgaclasses>javac -d . B.java

D:\durgaclasses>java pack2.B
Hello it is public method
```

Default Member Access – Can only be accessed by classes within the package

Class A is public , Method m1 is default.

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 20-02-2018

1 package pack1;
2 public class A
3 {
4     void m1()
5     {
6         System.out.println("Hello it is public method");
7     }
8 }
```

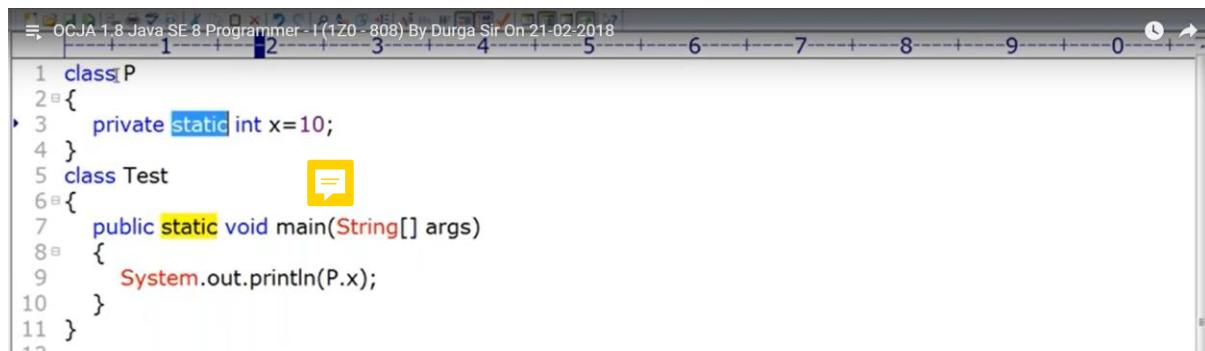
```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 20-02-2018

1 package pack2;
2 import pack1.A;
3 class B
4 {
5     public static void main(String[] args)
6     {
7         A a = new A();
8         a.m1();
9     }
10 }
11
```

```
Select C:\Windows\system32\cmd.exe
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 20-02-2018

D:\durgaclasses>javac -d . B.java
B.java:8: error: m1() is not public in A; cannot be accessed from outside package
        a.m1();
               ^
1 error
```

Private – Only within the class – even the child class cant access. Class Level Access



```
1 class P
2 {
3     private static int x=10;
4 }
5 class Test
6 {
7     public static void main(String[] args)
8     {
9         System.out.println(P.x);
10    }
11 }
```

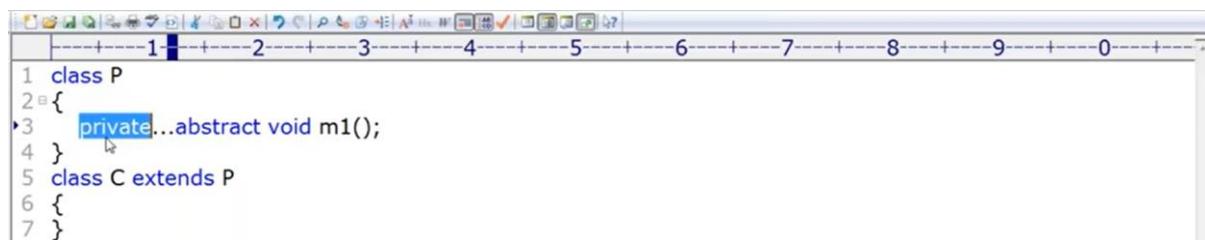


```
D:\durgaclasses>javac Test.java
Test.java:9: error: x has private access in P
        System.out.println(P.x);
                           ^
1 error
```



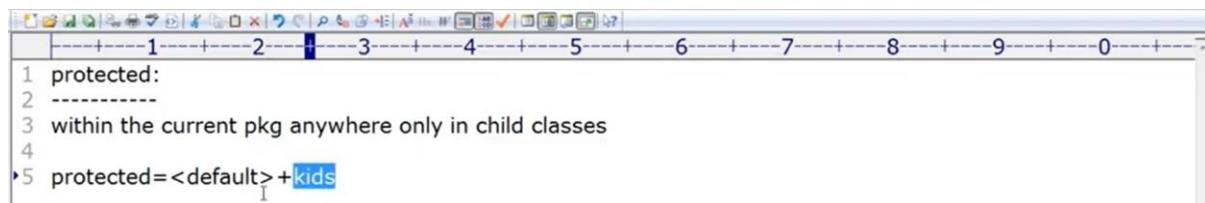
Abstract method cannot be Private and vice versa. Cant be accessed by the extending class and cannot be implemented

```
3 private==>class level access
4
5 abstract ==>private
6
```



```
1 class P
2 {
3     private...abstract void m1();
4 }
5 class C extends P
6 {
7 }
```

Protected – Can be accessed anywhere in the package but only by child classes outside package



```
1 protected:
2 -----
3 within the current pkg anywhere only in child classes
4
5 protected=<default>+kids
```

Parent Reference and Child Ref

The screenshot shows an IDE interface with a code editor and a 'Questions' panel. The code editor contains the following Java code:

```
1 package pack1;
2 public class A {
3 {
4     protected void m1()
5     {
6         System.out.println("Hello it is public method");
7     }
8 }
9 class C extends A
10 {
11     public static void main(String[] args)
12     {
13         A a = new A();
14         a.m1();
15
16         C c = new C();
17         c.m1();
18
19         A a1= new C();
20         a1.m1();
21     }
22 }
```

The line 'A a = new A();' is highlighted with a red box. The line 'C c = new C();' is also highlighted with a red box. The line 'A a1= new C();' is highlighted with a red box.

The 'Questions' panel on the right lists several questions from users like Govindu Rayapur, Poja Chavan, Srinivas Reddy, Amit Kumar, Kammoni Sundarraj, and Deepak Yadav. One question is highlighted: 'A a1=new C(); sir explain this line'. Below the list, a message says 'Just joined today. How many sessions did I miss?'.

Class B though in diff package but is child class so can access both method

The screenshot shows an IDE interface with a code editor. The code editor contains the following Java code:

```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5     public static void main(String[] args)
6     {
7         A a = new A();
8         a.m1();
9
10    }
11 }
12 }
```

Very Important. In diff package only the child class reference can access not the parent class reference

A a = new A();

a.m1() -> is a parent ref and throw and error..

The screenshot shows an IDE interface with a code editor. The code editor contains the following Java code:

```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5     public static void main(String[] args)
6     {
7         A a = new A();
8         a.m1();
9     }
10 }
```



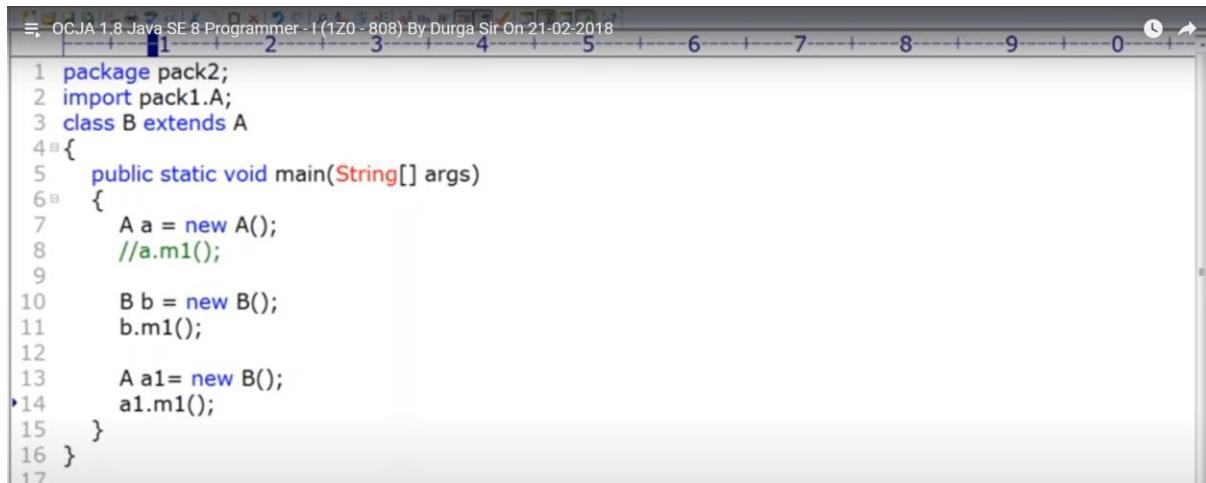
```
C:\Windows\system32\cmd.exe
= OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 21-02-2018
D:\durgaclasses>javac -d . B.java

D:\durgaclasses>java pack2.B
Hello it is protected method

D:\durgaclasses>javac -d . B.java
B.java:8: error: m1() has protected access in A
        a.m1();
               ^
1 error
```

Line 14 will also throw the same error. Parent reference cannot be used. Should use child ref only.

So first and third is invalid. Second one is valid



```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5     public static void main(String[] args)
6     {
7         A a = new A();
8         //a.m1();
9
10        B b = new B();
11        b.m1();
12
13        A a1= new B();
14        a1.m1();
15    }
16 }
17
```

So first the below commented are invalid. Lets see the others

The screenshot shows a Java code editor with the following code:

```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5 }
6 class D extends B
7 {
8     public static void main(String[] args)
9     {
10         A a = new A();
11         //a.m1();    I
12
13         B b = new B();
14         b.m1();
15
16         D d= new D();
17         d.m1();
18
19         A a1= new B();
20         //a1.m1();
21
22         A a2= new D();
23         //a2.m1();
24
25         B b1= new D();
26         b1.m1();
27     }
28 }
```

The code demonstrates various access levels and inheritance. It includes comments with 'I' indicating invalid syntax.

The screenshot shows a Java code editor with the following code:

```
1 protected:
2 -----
3 within the current pkg anywhere only in child classes
4
5 protected=<default>+kids
6
7
8 within the current pkg,we can access protected members anywhere
9 either within the child class or outside of child calss
10 we can use either parent reference or child reference
11
12 But from outside pakcage we can access protected members only in child classes
13 and we should use only child reference.]
```

This code illustrates the behavior of the protected access modifier within and outside a package.

If you are using in different package , we can access it from that child class ref only not from its Parent Ref B. u can use in child class but that class reference only

The screenshot shows a Java IDE interface with a code editor and a question list. The code editor contains the following Java code:

```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5 }
6 class D extends B
7 {
8     public static void main(String[] args)
9     {
10         B b = new B();
11         b.m1();
12
13         D d = new D();
14         d.m1();
15
16         B b1= new D();
17         b1.m1();
18     }
19 }
20
21
```

A red circle highlights the class definition 'class D extends B'. A red arrow points from this circle to the line 'D d = new D();' in the code. Another red arrow points from the same line to the variable 'd'. The question list on the right side of the interface shows several questions related to Java inheritance.

The screenshot shows a Java IDE interface with a code editor. The code is identical to the one in the previous screenshot:

```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5 }
6 class D extends B
7 {
8     public static void main(String[] args)
9     {
10         B b = new B();
11         //b.m1();
12
13         D d = new D();
14         d.m1();
15
16         B b1= new D();
17         //b1.m1();
18     }
19 }
20
21
```

Annotations are present in the code editor: a cursor is positioned at the end of the line 'B b = new B();', and a comment 'I' is placed after the line. The status bar at the bottom indicates the file is 'Untitled3.java' and the current line is 'In 18 col 11'.

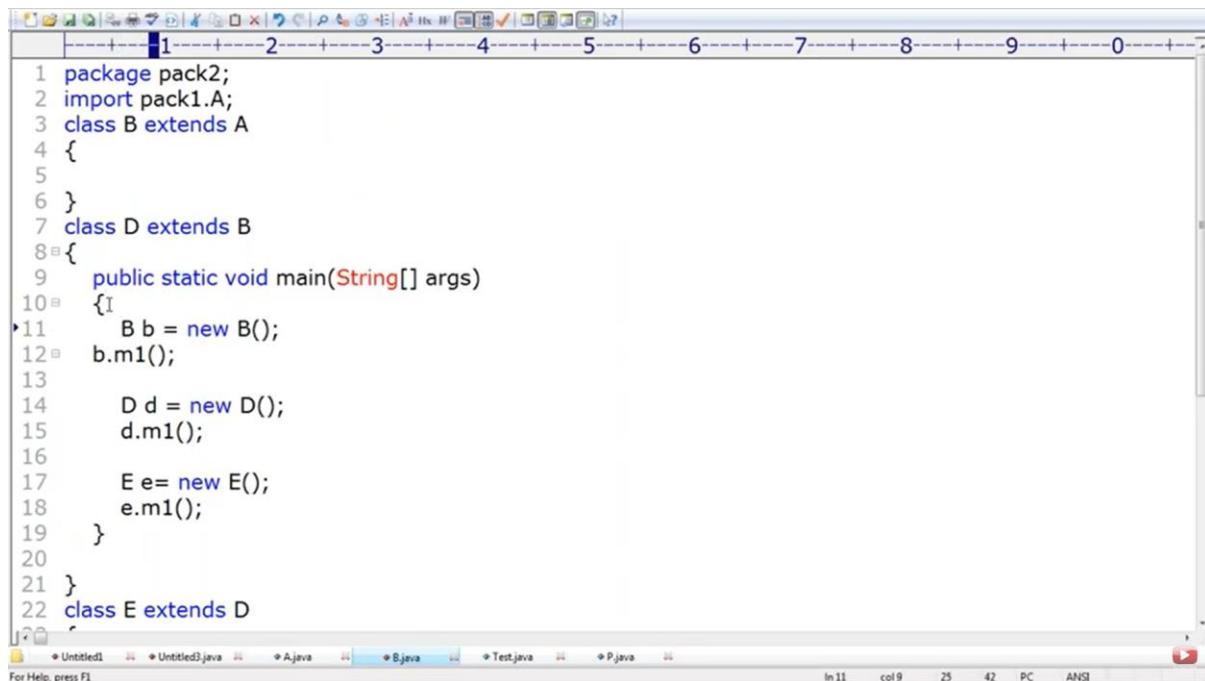
Extra care, if a member is protected, it can be accessed by the child class in diff package by using its ref or using its child ref or its child class reference

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 21-02-2018
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5     public static void main(String[] args)
6     {
7         D d = new D();
8         d.m1(); I
9     }
10 }
11 class D extends B
12 {
13 }
14
15 }
16
```

Below is also correct, its reference or its child class reference

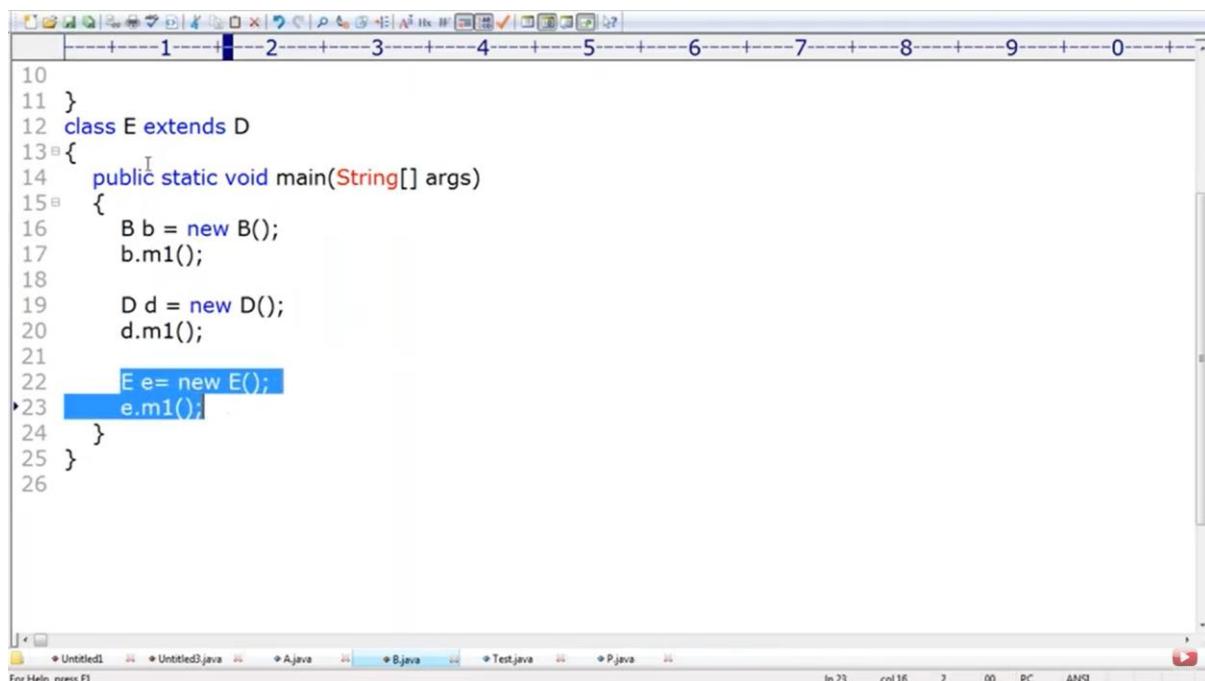
```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 21-02-2018
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5     public static void main(String[] args)
6     {
7         D d = new D();
8         d.m1(); I
9
10     E e = new E();
11     e.m1(); I
12 }
13 }
14 class D extends B
15 {
16 }
17
18 class E extends D
19 {
20 }
21
```

Here line 12 alone will throw error as its using parent ref



```
1 package pack2;
2 import pack1.A;
3 class B extends A
4 {
5
6 }
7 class D extends B
8 {
9     public static void main(String[] args)
10    {
11        B b = new B();
12        b.m1();
13
14        D d = new D();
15        d.m1();
16
17        E e = new E();
18        e.m1();
19    }
20
21 }
22 class E extends D
```

Only Line 23 is valid, no parent ref is not possible



```
10
11 }
12 class E extends D
13 {
14     public static void main(String[] args)
15    {
16        B b = new B();
17        b.m1();
18
19        D d = new D();
20        d.m1();
21
22        E e = new E();
23        e.m1();
24    }
25 }
```

Protected

7
8 within the current pkg, we can access protected members anywhere
9 either within the child class or outside of child class
10 we can use either parent reference or child reference
11
12 But from outside package we can access protected members only in child classes
13 and we should use only that particular child reference or its child classes.

```
Select C:\Windows\system32\cmd.exe
Hello it is protected method

D:\durgaclasses>javap java.lang.Object
Compiled from "Object.java"
public class java.lang.Object {
    public java.lang.Object();
    public final native java.lang.Class<?> getClass();
    public native int hashCode();
    public boolean equals(java.lang.Object);
    protected native java.lang.Object clone() throws java.lang.CloneNotSupportedException;
    public java.lang.String toString();
    public final native void notify();
    public final native void notifyAll();
    public final native void wait(long) throws java.lang.InterruptedException;
    public final void wait(long, int) throws java.lang.InterruptedException;
    public final void wait() throws java.lang.InterruptedException;
    protected void finalize() throws java.lang.Throwable;
    static {};
}

D:\durgaclasses>
```

These two protected methods are not properly implemented

Eg clone method does shallow cloning not deep cloning

Finalize method garbage collector – finalize class has empty implementation

```
1 public final native java.lang.Class<?> getClass();
2 public native int hashCode();
3 public boolean equals(java.lang.Object);
4 public java.lang.String toString();
5 public final native void notify();
6 public final native void notifyAll();
7 public final native void wait(long) throws java.lang.InterruptedException;
8 public final void wait(long, int) throws java.lang.InterruptedException;
9 public final void wait() throws java.lang.InterruptedException;
10
11
12 protected native java.lang.Object clone() throws java.lang.CloneNotSupportedException;
13 protected void finalize() throws java.lang.Throwable;
14
```

V

```

  OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 21-02-2018
  1 +-----+ 2 +-----+ 3 +-----+ 4 +-----+ 5 +-----+ 6 +-----+ 7 +-----+ 8 +-----+ 9 +-----+ 0
546 * Any exception thrown by the {@code finalize} method causes
547 * the finalization of this object to be halted, but is otherwise
548 * ignored.
549 *
550 * @throws Throwable the {@code Exception} raised by this method
551 * @see java.lang.ref.WeakReference
552 * @see java.lang.ref.PhantomReference
553 * @jls 12.6 Finalization of Class Instances
554 */
555 protected void finalize() throws Throwable { }
556 }

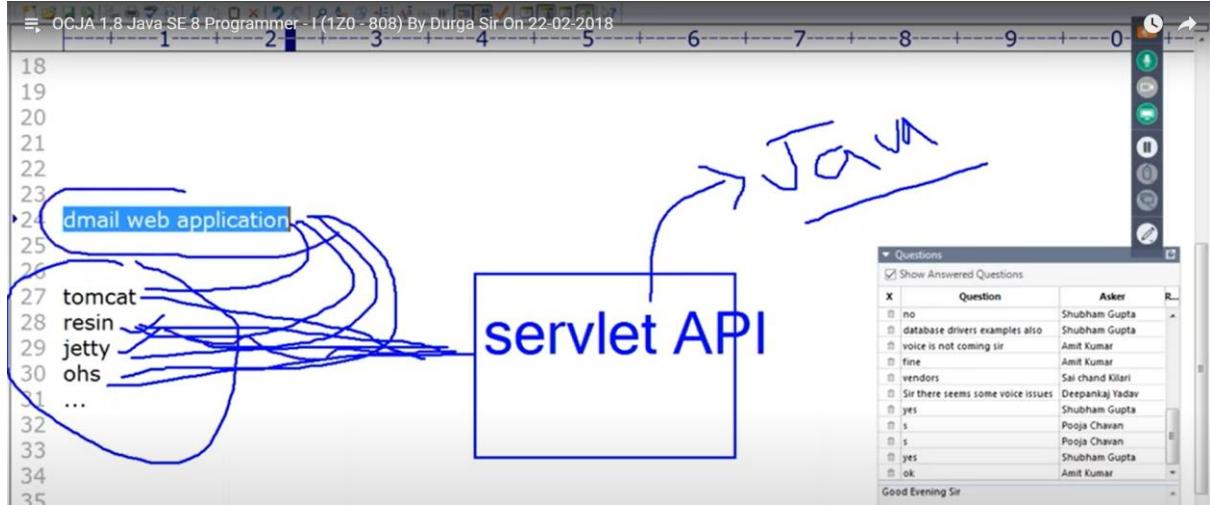
```

Highly recommended to override. So use protected method if you don't have proper implementation . If we declare this abstract then I have to override it so not needed.

```

  OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 22-02-2018
  1 +-----+ 2 +-----+ 3 +-----+ 4 +-----+ 5 +-----+ 6 +-----+ 7 +-----+ 8 +-----+ 9 +-----+ 0
6 interfaces:
7 -----
8 1. Any service requirement specification
9
10

```



Interfaces are no more 100% abstract. All methods are by default public

```
49 topic-1
50 topic-10
51
52
53
54 100 % pure abstract class
55 -----
56 interface Interf
57 {
58     void m1();
59     void m2();
60     default void m3()
61     {
62     }
63     public static void m4()
64     {
65     }
66
67 }
```

1. 8V

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 22-02-2018
49 topic-1
50 topic-10
51
52
53
54 100 % pure abstract class
55 -----
56 interface Interf
57 {
58     void m1();
59     void m2();
60     default void m3()
61     {
62     }
63     public static void m4()
64     {
65     }
66     private void m5()
67     {
68     }
69 }
```

1. 9V

```

1 interface Interf
2 {
3     void m1();
4     void m2();
5 }
6 class ServiceProvider implements Interf
7 {
8     void m1()
9     {
10 }
11 }
12

```

Questions

X	Question	Asker	R...
1	void m1() in Interf and m2 not overridden	Shruti sahu	
2	ohh	Shubham Gupta	
3	sir we should implement each method	Deepankaj Yadav	
4	overridden*	Shruti sahu	

When overriding an interface method, we cannot decrease the scope of the method. By default an interface method is public, whereas the access modifier is default in the class. So it should be public. ELSE CE.

In the below abstract class implementing interface needs not implement all methods the child classes can implement. Next level child class should implement

Every method should be public in the implementing class

```

1 interface Interf
2 {
3     void m1();
4     void m2();
5 }
6 abstract class ServiceProvider implements Interf
7 {
8     void m1()
9     {
10 }
11 }
12

```

```

C:\Windows\system32\cmd.exe
2 errors
D:\durgaclasses>javac ServiceProvider.java
ServiceProvider.java:8: error: m1() in ServiceProvider cannot implement m1() in Interf
    void m1()
           ^
attempting to assign weaker access privileges; was public
1 error

```

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 22-02-2018
1 interface Interf
2 {
3     void m1();
4     void m2();
5 }
6 abstract class ServiceProvider implements Interf
7 {
8     public void m1()
9     {
10 }
11 }
12 class SubServiceProvider extends ServiceProvider
13 {
14     I
15 }
16
```

```
D:\durgaclasses>javac ServiceProvider.java
D:\durgaclasses>javac ServiceProvider.java
ServiceProvider.java:12: error: SubServiceProvider is not abstract and does not override abstract method
in Interf
class SubServiceProvider extends ServiceProvider
^
1 error
```

Again the below line should be public

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 22-02-2018
1 interface Interf
2 {
3     void m1();
4     void m2();
5 }
6 abstract class ServiceProvider implements Interf
7 {
8     public void m1()
9     {
10 }
11 }
12 class SubServiceProvider extends ServiceProvider
13 {
14     | void m2()
15     {
16     }
17 }
18 }
19
```

```
1 2 3 4 5 6 7 8 9 0
76 implements
77
78
79 1. A class can extend only one class at a time
80 2. A class can implements any number of interfaces
81 3. A class can extends another class and can implements any no of interfaces
82 simultaneously
83 4. An interface can extend any number of interfaces at a time
```

```
87
88 A extends B
89
90 A and B should be either classes or interfaces
91
92 2. A extends B,C I
```

A , B and C should be interfaces only. Only an interface can extend any no of interfaces

```
93 3. A implements B,C
94     A-->class
95     B,C should be interfaces
96
97
98 4. A extends B implements C,D I
```

A & B- Class

C,D -> Interfaces

```
100 5. A implements C,D extends B
101
```

CE Error -> when implements and extends order is important order first and implements next

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 22-02-2018
1 extends ===>we are getting something from the parent
2 implements==>We have to giveI
3
4 -----
```

```

1 interface A
2 {
3     m1();
4     m2();
5 }
6 class B
7 {
8     m2()
9     {
10    }
11 }
12 class C implements A extends B
13 {
14     m1()
15     {
16    }
17     m2()
18     {
19    }
20 }

```

If implement can be accepted above case when compiler sees implements key word it would prompt for the implementation of both the methods before checking m2 which is implemented by B which the c class extends . so it should be extends followed by implements

```

1 interface A
2 {
3     m1();
4     m2();
5 }
6 class B
7 {
8     m2()
9     {
10    }
11 }
12 class C extends B implements A
13 {
14     m1()
15     {
16    }
17 }
18

```

Until java 7 every method is always by default u don't have to specify

```

101
102
103 interface methods:
104 -----
105 public and abstract|

```

Interface methods cannot be private protected static ...

```
107  
108 public & abstract==>private,,protected,static,synchronized,final,strictfp  
109 |
```

```
1 interface A  
2 {  
3     void m1();  
4     public void m1();  
5     abstract void m1();  
6     public abstract void m1();  
7 }
```

Last 1 is only valid

```
1 interface A  
2 {  
3     public void m1(){}
4     private void m1();
5     public final void m12();
6     public static void m1();
7     public abstract void m1();  
8 }
```

Interface variable should be always public static and final whether specified or not. They are constants

The screenshot shows a video player interface with a code editor and a question board.

Code Editor:

```
10 interface variables:  
11 -----  
12 any requirement specification...  
13  
14 cname=DURGASOFT  
15  
16 getMarks()  
17 extends  
18 updateMarks()  
19 getAttendance()  
20 updateAttendance()  
21  
22 LOC=HYD  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32
```

Hand-drawn annotations:

- A red box highlights the class declaration: `cname=DURGASOFT`.
- Red text inside the box lists methods: `getMarks()`, `updateMarks()`, `getAttendance()`, and `updateAttendance()`.
- A blue circle labeled `LOC=HYD` is at the bottom of the box.

Question Board:

X	Question	Asker	Rec'd
1	extend	Amit Kumar	9:42 PM
2	extends	Pooja Chavan	9:48 PM
3	s	Pooja Chavan	9:48 PM
4	s sir	Amit Kumar	9:48 PM
5	s	Pooja Chavan	9:49 PM
6	yes	Govindu Rayapuri	9:49 PM
7	final variables	Mohammad Alim Osmani	9:49 PM
8	SRS	Shubham Gupta	9:49 PM
9	requirement specification	John acharya	9:49 PM
10	s	Shubham Gupta	9:51 PM
11	s	Pooja Chavan	9:52 PM
12	s	Amit Kumar	9:52 PM
13	clear	Govindu Rayapuri	9:52 PM
14	yes sir	Mohammad Alim Osmani	9:52 PM
15	constant variable	Amit Kumar	9:53 PM

It should be public to be accessed within the package or outside

The screenshot shows a video player interface with a code editor and a question board.

Code Editor:

```
13  
14 public  
15 static  
16 final  
17  
18 int x=10;  
19 public int x =10;  
20 static int x =10;  
21 final int x =10;  
22 public static final int x =10;  
23  
24  
25  
26
```

Hand-drawn annotations:

- A blue arrow points from the `final` keyword in line 16 to the `x` variable in line 18.

Question Board:

X	Question	Asker	Rec'd
1	extend	Amit Kumar	9:48 PM
2	extends	Pooja Chavan	9:48 PM
3	s	Pooja Chavan	9:48 PM
4	s sir	Amit Kumar	9:48 PM
5	s	Pooja Chavan	9:49 PM
6	yes	Govindu Rayapuri	9:49 PM
7	final variables	Mohammad Alim Osmani	9:49 PM
8	SRS	Shruti sahu	9:49 PM
9	requirement specification	John acharya	9:49 PM
10	s	Shubham Gupta	9:51 PM
11	s	Shubham Gupta	9:52 PM
12	s	Pooja Chavan	9:52 PM
13	s	Amit Kumar	9:52 PM
14	clear	Govindu Rayapuri	9:52 PM
15	yes sir	Mohammad Alim Osmani	9:52 PM
16	constant variable	Amit Kumar	9:53 PM

Its not possible to create an object for an interface and to access the variables it should be static

The screenshot shows a video player interface with a code editor and a question board.

Code Editor:

```
22 public static final int x =10;  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38
```

Hand-drawn annotations:

- A blue box highlights the `final` keyword in line 22.
- Red text inside the box says `final = X`.
- Three red circles are drawn below the code editor, connected by lines to the `final` keyword in the code.

Question Board:

X	Question	Asker	Rec'd
1	final variables	Shruti sahu	9:49 PM
2	SRS	Amit Kumar	9:49 PM
3	requirement specification	John acharya	9:49 PM
4	s	Shubham Gupta	9:51 PM
5	s	Shubham Gupta	9:52 PM
6	s	Pooja Chavan	9:52 PM
7	s	Amit Kumar	9:52 PM
8	clear	Govindu Rayapuri	9:52 PM
9	yes sir	Mohammad Alim Osmani	9:52 PM
10	constant variable	Amit Kumar	9:53 PM
11	all r valid	Amit Kumar	9:53 PM
12	s	Pooja Chavan	9:53 PM
13	s	Amit Kumar	9:54 PM
14	any no	Pooja Chavan	9:54 PM
15	base on requirements	Amit Kumar	9:55 PM
16	s	Amit Kumar	9:55 PM

Since its public it cant be private or protected

Since we cant create object it cant be serialized we cant use transient

Not volatile

The screenshot shows a Java code editor with the following code in a file named `Interf.java`:

```
1 interface Interf
2 {
3     private static final int x=10;
4 }
```

Below the code, the terminal output shows the compilation command and the resulting error:

```
2 D:\durgaclasses>javac Interf.java
3 D:\durgaclasses>javac Interf.java
Interf.java:3: error: modifier private not allowed here
    private static final int x=10;
                           ^
1 error
```

The screenshot shows a Java code editor with the following code in a file named `Test.java`:

```
1 class Test
2 {
3     static int x;
4     public static void main(String[] args)
5     {
6         System.out.println(x);
7     }
8 }
```

If its final JVM would not specify default value for static member, what value is final JVM does not know CE. For instance and Static variable JVM will provide default but final static and instance variable JVM will not provide

The screenshot shows a Java code editor with the following code in a file named `Test.java`:

```
1 class Test
2 {
3     final static int x;
4     public static void main(String[] args)
5     {
6         System.out.println(x);
7     }
8 }
```

A yellow warning icon is visible next to the declaration of the `x` variable.

Interface variable is by default static and final and hence it should be initialized . Hence below error

```
1 class Test
2 {
3     final static int x;
4     public static void main(String[] args)
5     {
6         System.out.println(x);
7     }
8 }
```

Order of modifiers are not important in java

This is valid as its by default is static

```
1 interface Interf
2 {
3     int x=777;
4 }
5 class Test implements Interf
6 {
7     public static void main(String[] args)
8     {
9         System.out.println(x);
10    }
11 }
```

```
1 interface Interf
2 {
3     int x=777;
4 }
5 class Test implements Interf
6 {
7     public static void main(String[] args)
8     {
9         int x = 888;
10        System.out.println(x);
11    }
12 }
```

Annotations and a question list are visible on the right side of the IDE.

X	Question	Asker	Rec'd
final	Shruti sahu	Amit Kumar	10:15 P...
we can't change the value	Amit Kumar	Govindu Rayapur	10:15 P...
cont override	Govindu Rayapur	Pooja Chavan	10:15 P...
can reass	Pooja Chavan	Amit Kumar	10:15 P...
of final var	Amit Kumar	Mohammad Alim Os...	10:15 P...
final	Mohammad Alim Os...	Sai chand Kilari	10:15 P...
if int x=888; o/p=888	Sai chand Kilari	john acharya	10:15 P...
int	john acharya	Amit Kumar	10:15 P...
s	Amit Kumar	Pooja Chavan	10:15 P...
s	Pooja Chavan	Shruti sahu	10:15 P...
yes	Shruti sahu	Mohammad Alim Os...	10:15 P...
s sir	Mohammad Alim Os...	Sai chand Kilari	10:16 P...
888	Sai chand Kilari	Pooja Chavan	10:16 P...
s	Pooja Chavan		10:16 P...

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 23-02-2018  
1 interface Interf  
2 {  
3     int x=777;  
4 }  
5 class Test implements Interf  
6 {  
7     public static void main(String[] args)  
8     {  
9         int x =888;  
10        System.out.println(x);  
11        System.out.println(Interf.x);  
12    }  
13 }
```

```
TRANSIENT  
28 volatile  
29  
30 Interface Naming conflicts:  
31 -----  
32 1. Method naming conflicts:  
33 -----
```

Same args one implementation is enough

```
1 interface Left  
2 {  
3     public void m1();  
4 }  
5 interface Right  
6 {  
7     public void m1();  
8 }  
9 class Test implements Left,Right  
10 {  
11     public void m1()  
12     {  
13     }  
14 }  
15
```

Questions

X	Question	Asker	Rec'd
ohhhhh	Shubham Gupta	10:17 P...	✓
Interf.x	Shubham Gupta	10:17 P...	
ok.	Shubham Gupta	10:17 P...	
clear	Shubham Gupta	10:18 P...	
sir	Shubham Gupta	10:18 P...	
2	Pooja Chavan	10:19 P...	
2	Sai chand Kilar	10:19 P...	
2	Shubham Gupta	10:19 P...	
2	Govindu Rayapur	10:19 P...	
two	Amit Kumar	10:19 P...	
2	Mohammad Alim Os...	10:19 P...	
2	john acharya	10:19 P...	
m1()	Amit Kumar	10:19 P...	
1	Shubham Gupta	10:20 P...	
for one	Mohammad Alim Os...	10:20 P...	
1	Pooja Chavan	10:20 P...	
one	Govindu Rayapur	10:20 P...	

sir then how we can use interface variable and local var same time

Overloaded methods

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 23-02-2018
1 interface Left
2 {
3     public void m1(int i);
4 }
5 interface Right
6 {
7     public void m1();
8 }
9 class Test implements Left,Right
10 {
11     public void m1()
12     {
13     }
14     public void m1(int i)
15     {
16     }
17 }
18
```

Same Signature Interfaces CE:

If we implement two methods with different return types. What will be used

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 23-02-2018
1 interface Left
2 {
3     public void m1();
4 }
5 interface Right
6 {
7     public int m1();
8 }
9 class Test implements Left,Right
10 {
11     public void m1() // Method 1
12     {
13     }
14     public int m1() // Method 2
15     {
16         return 10;
17     }
18 }
```

Test t = new Test();
t.m1();

X	Question	Asker	Rec'd
1	two?	Shruti sahu	10:28 P...
1	yes	john acharya	10:28 P...
1	s	Amit Kumar	10:28 P...
1	yes	Mohammad Alim Os...	10:28 P...
2	2	Amit Kumar	10:28 P...
2	2	Shruti sahu	10:28 P...
2	2	Mohammad Alim Os...	10:28 P...
2	2	Govindu Rayapur	10:28 P...
2	2	Sai chand Kilari	10:28 P...
2	2	Pooja Chavan	10:28 P...
2	s	Pooja Chavan	10:29 P...
2	s	Mohammad Alim Os...	10:29 P...
2	s	Amit Kumar	10:29 P...
2	s sir	Sai chand Kilari	10:29 P...
2	sir, method signature??	Govindu Rayapur	10:29 P...
2	hahhaaa	Mohammad Alim Os...	10:29 P...

if imple of m1() diff for both interfaces??

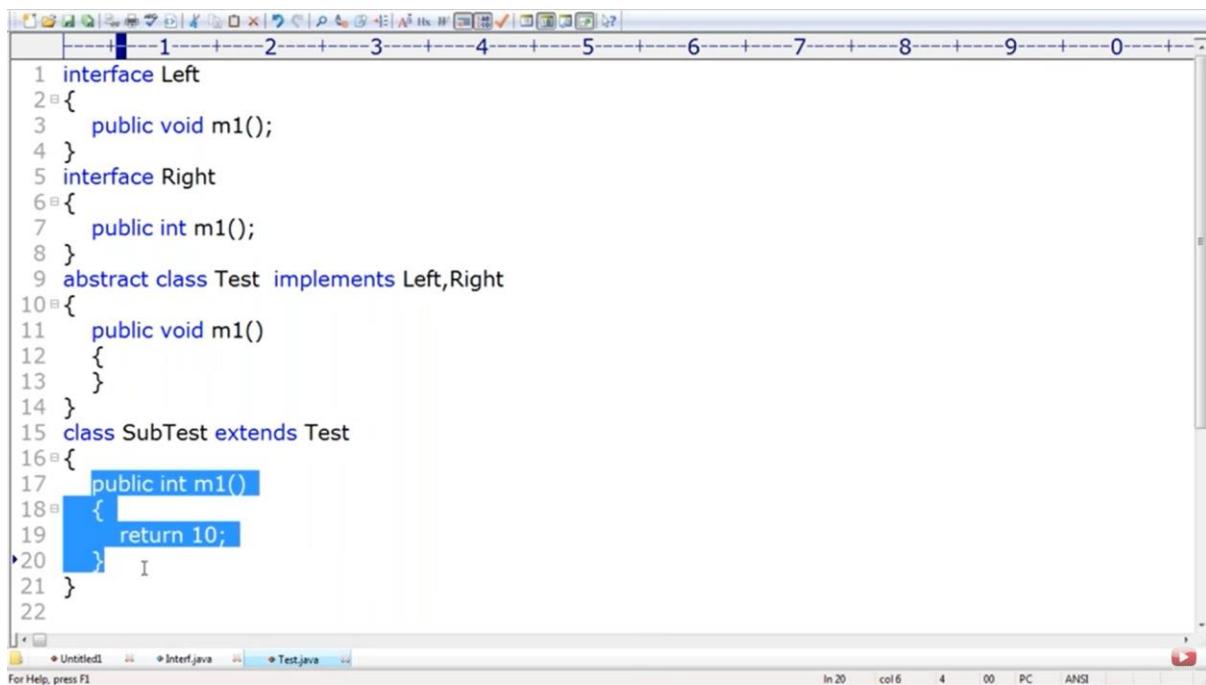
Compile time error , so If two interfaces have same method name and arguments and signature is different a class cant implement both the interface at the same time. Compiler would ask why are u declaring the method m1.Within a class two methods with same signature not allowed

New CE as below

```
C:\Windows\system32\cmd.exe - javac Test.java
D:\durgaclasses>javac Test.java
D:\durgaclasses>javac Test.java
Test.java:9: error: Test is not abstract and does not override abstract method m1() in Right
class Test implements Left,Right
^
Test.java:11: error: m1() in Test cannot implement m1() in Right
    public void m1()
        ^
        return type void is not compatible with int
1 errors

D:\durgaclasses>javac Test.java
D:\durgaclasses>javac Test.java
Test.java:14: error: method m1() is already defined in class Test
    public int m1()
        ^
1 errors
```

The below will also not work. U may think as overriding and use the second child class method



A screenshot of a Java code editor showing a syntax error. The code is as follows:

```
1 interface Left
2 {
3     public void m1();
4 }
5 interface Right
6 {
7     public int m1();
8 }
9 abstract class Test implements Left,Right
10 {
11     public void m1()
12     {
13     }
14 }
15 class SubTest extends Test
16 {
17     public int m1()
18     {
19         return 10;
20     }
21 }
```

The line `return 10;` is highlighted in blue, indicating a syntax error. The status bar at the bottom right shows "In 20 col 6 4 00 PC ANSI".

While overriding return types should be compatible

Parent class and child class has same signature then only overriding is allowed. So this also wrong

```
1 interface Left
2 {
3     public void m1();
4 }
5 interface Right
6 {
7     public int m1();
8 }
9 abstract class Test implements Left,Right
10 {
11     public void m1()
12     {
13     }
14 }
15 class SubTest extends Test
16 {
17     public int m1()
18     {
19         return 10;
20     }
21 }
22
```

```
Select C:\Windows\system32\cmd.exe
return type void is not compatible with int
3 errors

D:\durgaclasses>javac Test.java
Test.java:11: error: m1() in Test cannot implement m1() in Right
    public void m1()
               ^
      return type void is not compatible with int
Test.java:17: error: m1() in SubTest cannot implement m1() in Left
    public int m1()
               ^
      return type int is not compatible with void
2 errors

D:\durgaclasses>
```

If 2 interface with same signature but diff return types its not possible to implement both the interface simultaneously

The screenshot shows a Java code editor with the following code:

```
1 interface Left
2 {
3     public void m1();
4 }
5 interface Right
6 {
7     public int m1();
8 }
9 abstract class Test implements Left,Right
10 {
11     public void m1()
12     {
13     }
14 }
15 class SubTest extends Test
16 {
17     public int m1()
18     {
19         return 10;
20     }
21 }
```

Annotations in red boxes highlight the following parts of the code:

- A red box surrounds the declaration of the `Left` interface.
- A red box surrounds the declaration of the `Right` interface.
- A red box surrounds the implementation of `m1()` in the `Test` class.
- A red box surrounds the implementation of `m1()` in the `SubTest` class.
- A large red X is drawn over the entire code block, indicating that it is incorrect or non-functional.

To the right of the code editor is a "Questions" panel from a Q&A platform, showing a list of user posts and answers.

The screenshot shows a Java code editor with the following code:

```
34
35
36
37
38 Q. Is a java class can implements any no of interfaces simultaneously?
39
40
```

Annotation 38: A question is highlighted in blue: "Q. Is a java class can implements any no of interfaces simultaneously?"

Except if two interfaces the same method with same signature but diff return types than cant implement both the interfaces simultaneously

Variable naming conflict

A screenshot of a Java code editor window. The code defines two interfaces, Left and Right, each with its own variable x. A class Test implements both interfaces and prints the value of x. The code is as follows:

```
1 interface Left
2 {
3     int x = 777;
4 }
5 interface Right
6 {
7     int x = 888;
8 }
9 class Test implements Left,Right
10{
11     public static void main(String[] args)
12     {
13         System.out.println(x);
14     }
15 }
16
```

A screenshot of a terminal window showing the compilation of the Java code. It displays two errors: one about the return type of the main method and another about a reference to x being ambiguous. The terminal output is as follows:

```
Select C:\Windows\system32\cmd.exe
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 23-02-2018
public int m1()
^
return type int is not compatible with void
2 errors

D:\durgaclasses>javac Test.java
Test.java:13: error: reference to x is ambiguous
        System.out.println(x);
                           ^
both variable x in Left and variable x in Right match
1 error
```

A screenshot of a Java code editor window showing the same code as before, but with a fix. The class Test now prints the value of Left.x and Right.x separately. The code is as follows:

```
1 interface Left
2 {
3     int x = 777;
4 }
5 interface Right
6 {
7     int x = 888;
8 }
9 class Test implements Left,Right
10{
11     public static void main(String[] args)
12     {
13         //System.out.println(x);
14         System.out.println(Left.x);
15         System.out.println(Right.x);
16     }
17 }
```

The screenshot shows a Java code editor interface with the following details:

- Toolbar:** Standard Java editor toolbar with icons for file operations, search, and navigation.
- Status Bar:** Shows "In 17 col 6 19 00 PC ANSI".
- Code Area:** Displays the following Java code:

```
1 interface Left
2 {
3     public void m1();
4 }
5 interface Right
6 {
7     public int m1();
8 }
9 class Test implements Left,Right
10{
11     public void m1()
12     {
13     }
14     public int m1()
15     {
16         return 10;
17     }
18 }
```
- Tab Bar:** Shows three tabs: "Untitled1", "Interf.java", and "Test.java".

The screenshot shows a Java code editor with the following code:

```
1 interface Left
2 {
3     public void m1();
4 }
5 interface Right
6 {
7     public int m1();
8 }
9 abstract class Test implements Left,Right
10 {
11     public void m1()
12     {
13     }
14 }
15 class SubTest extends Test
16 {
17     public int m1()
18     {
19         return 10;
20     }
21 }
```

The code defines two interfaces, `Left` and `Right`, and an abstract class `Test` that implements both. The `Test` class has an empty `m1()` method. A subclass `SubTest` extends `Test` and overrides the `m1()` method to return 10.

Marker Interface

The screenshot shows a Java code editor with the following code:

```
17 does not contain any methods and by implementing that interface if our objects will get some ability
18
19 JVM is responsible
20
21
22
23 Serializable(I)
24 Cloneable(I) I
25 RandomAccess(I)
26 SingleThreadModel
27
```

The code demonstrates various marker interfaces: `Serializable`, `Cloneable`, `RandomAccess`, and `SingleThreadModel`. The `Serializable` interface is shown with its parameter `I` highlighted.

The screenshot shows a Java code editor with two panes. The top pane displays a class hierarchy:

```
1 class Student
2 {
3 }
4 class Test
5 {
6     public static void main(String[] args) throws Exception
7     {
8         Student s = new Student();
9         Student s1= (Student)s.clone();
10    }
11 }
```

The bottom pane shows the command line output of running the compiler:

```
D:\durgaclasses>javac Test.java
Test.java:9: error: clone() has protected access in Object
          Student s1= (Student)s.clone();
                           ^
1 error
```

You can use clone which is a protected method in Object on the child class of that object only. So you can use it inside Student class not in Test

The screenshot shows a Java code editor with two panes. The top pane displays a class hierarchy:

```
1 class Test
2 {
3     public static void main(String[] args) throws Exception
4     {
5         Test t = new Test();
6         Test t1= (Test)t.clone();
7     }
8 }
```

The bottom pane shows the command line output of running the compiler and then executing the program:

```
1 Select On Windows System32cmd.exe
1 OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir On 26-02-2018
1 error
1
2 D:\durgaclasses>javac Test.java
3
4 D:\durgaclasses>java Test
Exception in thread "main" java.lang.CloneNotSupportedException: Test
        at java.lang.Object.clone(Native Method)
        at Test.main(Test.java:6)
```

The screenshot shows a Java code editor with two panes. The top pane displays a class hierarchy:

```
1 class Test implements Cloneable
2 {
3     public static void main(String[] args) throws Exception
4     {
5         Test t = new Test();
6         Test t1= (Test)t.clone();
7     }
8 }
```

We need to implement cloneable interface to use clone method. But the cloneable interface has no methods declared. By implementing Cloneable interface the object is going to get the ability to clone.

```

28
29 1.8 version 2 enhancements for interfaces:
30 -----
31
32 1. default methods
33 2. static methods
34
35

```

```

1 interface X
2 {
3     m1();
4     m2();
5 }
6 class Test1 implements X
7 {
8     m1(){}
9     m2(){}
10}
11 class Test2 implements X
12 {
13     m1(){}
14     m2(){}
15}
16 ..
17 class Test1000 implements X
18 {
19     m1(){}
20     m2(){}
21}
22

```

For Help, press F1 In 17 col 15 22 20 PC ANSI

Without affecting the implementation classes if you want to enhance the interface. After 1.7 you can use the below, if you want to override the below default u r welcome to do. Here default is not a modifier by a keyword

```

1 interface X
2 {
3     m1();
4     m2();
5     default m3()
6     {
7         dummy implementation
8     }
9 }
10
11 class Test1 implements X
12 {
13     m1(){}
14     m2(){}
15 }
16 class Test2 implements X
17 {
18     m1(){}
19     m2(){}
20 }
21 ..
22 class Test1000 implements X

```

X	Question	Asker
<input type="radio"/> yes		Amit Kumar
<input type="radio"/> can't change		Amit Kumar
<input type="radio"/> 1.7		Amit Kumar
<input type="radio"/> 3		Shubham Gupta
<input type="radio"/> yes sir		john acharya
<input type="radio"/> concrete		Shubham Gupta
<input type="radio"/> concrete		Govindu Rayapur
<input type="radio"/> concrete		Sai chand Khan
<input type="radio"/> concrete		john acharya
<input type="radio"/> what is use of dummy impl???		Mohammad Alim O...
<input type="radio"/> sir why have you done (Test) while creating clone?		Govindu Rayapur

For Help, press F1 In 8 col 6 Send Privately Send to All

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 26-02-2018  
1 interface Interf  
2 {  
3     default void m1()  
4     {  
5         System.out.println("Default Method");  
6     }  
7 }  
8 class Test implements Interf  
9 {  
10    public static void main(String[] args)  
11    {  
12        Test t = new Test();  
13        t.m1();  
14    }  
15 }  
16
```

overriding

```
1 interface Interf  
2 {  
3     default void m1() +  
4     {  
5         System.out.println("Default Method");  
6     }  
7 }  
8 class Test implements Interf  
9 {  
10    public void m1()  
11    {  
12        System.out.println("My own implementation");  
13    }  
14    public static void main(String[] args)  
15    {  
16        Test t = new Test();  
17        t.m1();  
18    }  
19 }  
20
```

Questions

X	Question	Asker
1	nooo sir	Govindu Rayapur
2	ohhhhhh	Shubham Gupta
3	ok	Shubham Gupta
4	sir we can take as it is , default key avail to interface only??	Govindu Rayapur
5	clear.	Shubham Gupta
6	s	Amit Kumar
7	got it..	Amit Kumar
8	s	Mohammad Alim O...
9	switch	Shubham Gupta
10	switch	Sai chand Kilar

sir why have you done (Test) while creating clone?

```
1 interface Collection  
2 {  
3     ...  
4     default stream(){}  
5 }  
6 AL,LL,V,S,HS,TS... 100 classes
```

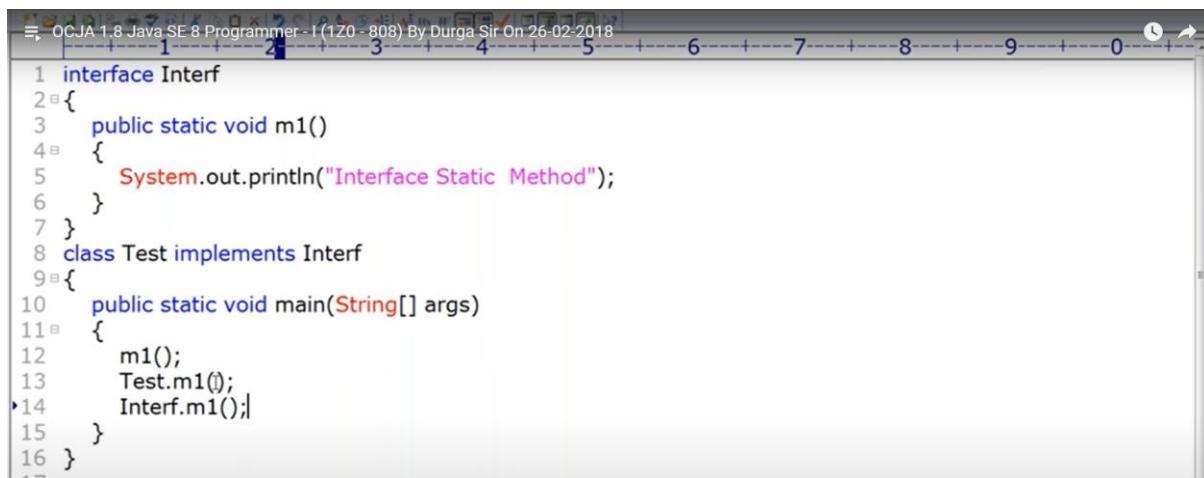
We can have static methods in a class, classes are heavy weight and object creation is costly. Since static method needs no object creation for simplicity purpose we can use it in interface. Available after 1.7



```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 26-02-2018  
1 static methods  
2  
3 class and interface  
4  
5 public static void m1()  
6 {  
7 }  
8
```

Static method present inside interface by default not available for implementing classes.

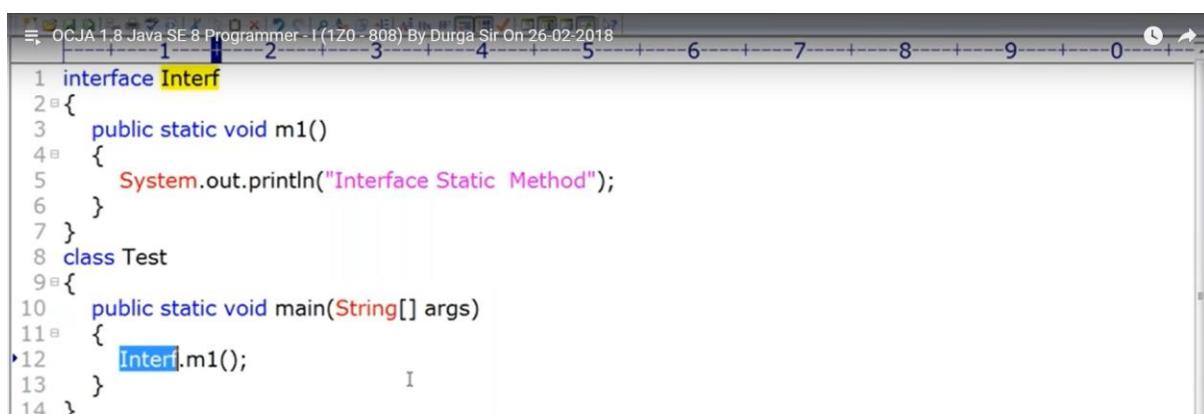
Should be called using Interf.m1(); So line 12 and 13 are wrong



```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 26-02-2018  
1 interface Interf  
2 {  
3     public static void m1()  
4     {  
5         System.out.println("Interface Static Method");  
6     }  
7 }  
8 class Test implements Interf  
9 {  
10    public static void main(String[] args)  
11    {  
12        m1();  
13        Test.m1();  
14        Interf.m1();  
15    }  
16 }
```

Interface static methods is available for every person even if you don't implement it

Default method needs to be implemented and object has to be created



```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir On 26-02-2018  
1 interface Interf  
2 {  
3     public static void m1()  
4     {  
5         System.out.println("Interface Static Method");  
6     }  
7 }  
8 class Test  
9 {  
10    public static void main(String[] args)  
11    {  
12        Interf.m1();  
13    }  
14 }
```

U can have a main method also in Interf and can run the interface

```
1 interface Interf
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Interface Main method");
6     }
7 }
```

After 1.7 from 1.8

```
D:\durgaclasses>javac Test.java
D:\durgaclasses>javac Interf.java
D:\durgaclasses>java Interf
Interface Main method
```

```
32 1. default methods
33 2. static methods
34
35
36
37 sir why we can't use default & static both in a single method means why this one is invalid sir default static
void methodOne() {}
38
39
40
41 default methods==>implementation classes
42 static methods ==>general utility methods need not be to implementation
43
```

```
41 default methods==>implementation classes
42 static methods ==>general utility methods need not be to implementation
43
44 sir one more doubt then abstract class is more or less similar to interface now? abstract class also have 0
or more abstract methods like interface
45
46 interface ==>
47 abstract class==>int x =10
48             constructor
49             instance block
50             static block
51             public concrete instance methods
```

After 1.8

```
56
57 1. default methods
58 2. static methods
59
60 1.9 version:
61 -----
62 private methods concepts |
```

For code reusability

A screenshot of a Java IDE showing the code for an interface named 'Interf'. The code includes three methods: m1(), m2(), and m3(), each containing a call to a private static method cm(). A tooltip '90 lines of code' is visible over the cm() method. The IDE interface shows tabs for Untitled1, Test.java, Untitled3.java, Untitled4.java, and Interf.java.

```
1 interface Interf
2 {
3     public static void m1()
4     {
5         10
6         cm();
7     }
8     public static void m2()
9     {
10        10
11        cm();
12    }
13     public static void m3()
14    {
15        10;
16        cm();
17    }
18     private static void cm()
19    {
20        90 lines of code
21    }
22 }
```

A screenshot of a text editor showing a question from a Java course. The question asks why a constructor is valid inside an abstract class, despite object creation being impossible for abstract classes. The text is part of a larger document with other numbered questions.

65
66
67 sir why constructor is valid inside abstract class as object creation is not possible for abstract class means
no constructor calling is possible
68
69

Abstract class can also have instance variable, when u create child class object parent class constructor will be called . Also use super()