

```
1 Interfaces
2 -----
3 An interface is a construct which contains the collection of purely
  undefined methods or an interface is a collection of purely abstract
  methods.
4 1. Interface is also one of the type of class it contains only
  abstract methods.
5 2. For the interfaces also .class files will be generated.
6 3. Each and every interface by default abstract hence it is not
  possible to create an object.
7 4. Interfaces not alternative for abstract class it is extension for
  abstract classes.
8 5. 100 % pure abstract class is called interface.
9 6. The Interface contains only abstract methods means unimplemented
  methods.
10 7. Interfaces giving the information about the functionalities it are
  not giving the information about internal implementation.
11 8. To provide implementation for abstract methods we have to take
  separate class that class we can called it as implementation class
  for that interface.
12 9. Interface can be implemented by using implements keyword.
13 10. For the interfaces also the inheritance concept is applicable.
14 -----
15 Interfaces:
16 11. Interfaces are basically used to develop user defined data types.
17 12. With respect to interfaces we can achieve the concept of multiple
  inheritances.
18 13. With interfaces we can achieve the concept of polymorphism,
  dynamic binding and hence we can improve the performance of a JAVA
  program in turns of memory space and execution time.
19
20 Interface Variables:
21 -----
22 An interface can contain variables.
23 The main purpose of interface variables is to define requirement
  level constants.
24 Every interface variable is always public static and final whether we
  are declaring or not.
25 Ex:
26 interface IntFEx1
27 {{
28     int x=10;
29 }
30 class MyTest implements IntFEx1
31 {{
32     public static void main(String[] args)
33     {{
34         System.out.println("X="+x);
35     }
36 }
```

```
36 }
37
38
39 Syntax for defining an interface:
40 -----
41 interface <interface name>
42 {}{
43 Variable declaration;
44 Method declaration;
45 }
46 -----
47 Note1:
48
49 when ever we are implementing an interface compulsory for every
method of that interface we should provide implementation other wise
we have to declare class as abstract in that case child class is
responsible to provide implementation for remaining methods.
50 //Ex1
51 interface Intfacel//By defacult interface is abstract so 100%pure
abstract
52 {}{
53     //By default interface methods are public and abstract
54 void m1();
55 void m2();
56 void m3();
57 }
58 class Ex1_s15 implements Intfacel
59 {}{
60 public void m1()
61 {}{
62 System.out.println("In m1 method");
63
64 }
65 public void m2()
66 {}{
67 System.out.println("In m2 method");
68
69 }
70 public void m3()
71 {}{
72 System.out.println("In m3 method");
73
74 }
75 public static void main(String[] args)
76     {}{
77         Ex1_s15 ob1=new Ex1_s15();
78         ob1.m1();
79         ob1.m2();
80         ob1.m2();
```

```
81
82     Intfacel ob2=new Ex1_s15();
83     ob2.m1();
84     ob2.m2();
85     ob2.m3();
86
87
88
89
90
91     }
92 }
93 /*
94
95 D:\KLUUniversity\OOPCourse\S15Interface>javac Ex1_s15.java
96 Ex1_s15.java:21: error: m3() in Ex1_s15 cannot implement m3() in
    Intfacel
97 void m3()
98     ^
99     attempting to assign weaker access privileges; was public
100 Ex1_s15.java:16: error: m2() in Ex1_s15 cannot implement m2() in
    Intfacel
101 void m2()
102     ^
103     attempting to assign weaker access privileges; was public
104 Ex1_s15.java:11: error: m1() in Ex1_s15 cannot implement m1() in
    Intfacel
105 void m1()
106     ^
107     attempting to assign weaker access privileges; was public
108 3 errors
109
110 D:\KLUUniversity\OOPCourse\S15Interface>
111 */
112 -----
113 Ex 2:Extending an Interface
114 //Extending Interfaces
115 interface A
116 {}{
117 void showA();
118 }
119 interface B extends A
120 {}{
121 void showB();
122 }
123
124 class IntfaceDemo implements B
125 {}{
126 public void showA()
```

```
127  {}{
128  System.out.println("Overridden method of Interface-A");
129
130  }
131  public void showB()
132  {}{
133  System.out.println("Overridden method of Interface-B");
134
135  }
136  public static void main(String[] args)
137  {}{
138  InterfaceDemo d=new InterfaceDemo();
139  d.showA();
140  d.showB();
141  }
142  }
143  =====
144  //Multiple Inheritance using interface
145  interface Shape1
146  {}{
147  void area();
148  }
149  interface Shape2
150  {}{
151  void area();
152  }
153  interface Shape3
154  {}{
155  void areaTest();
156  }
157  class IntshapeTest implements Shape1,Shape2,Shape3
158  {}{
159      public void area()
160          {}{
161              System.out.println("we need to write logic Later");
162          }
163      public void areaTest()
164          {}{
165          System.out.println("Method is Not Overriding because
166              overridden and overriding method are not same");
167          }
168      public static void main(String[] args)
169          {}{
170          IntshapeTest t=new IntshapeTest();
171          t.area();
172          t.areaTest();
173      }
174  }
```

175 }

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.