Interface

- 100% abstraction
- Variables are public static final (by default)
- methods are public & abstract (by default)
- multiple inheritance is possible
- 1 interface & 1 class
- keywords - interface & implements
- interface must be inherited
- can't create the object of an interface.

Case 1

```
interface A
 {
    public static final int a = 10;
    public abstract void show();
 }
  class B implements A
  {
   void show()
   {
     Sopln("Value of a = "+a);
   }
    void display()
   {
     Sopln("I am the method of class B");
   }
  }
  class Main
  {
     public static void main(String as[])
      {
        B ob = new B();
         ob.show();
         ob.display();
      }
  }
```

Output :-

```
Value of a = 10
I am the method of class B
```

// An interface can extends another interface

Case 2

```
interface A
{
  int a = 10;
  void show();
}
interface B extends A
{
  void display();
}
class C implements B
{
  public void show()
  {
    Sopln(" Value of a = " + a);
  }
  public void display()
  {
    Sopln(" I am the method of interface B");
  }
}
class Main
{
  public static void main(String args[])
  {
    C ob = new C();
    ob.show();
    ob.display();
  }
}
```

Case 3 :-

```
interface A
{
 int a = 10;
 void show();
}
interface B
{
  void add(int a, int b);
  interface C extends A, B
```

```
{
  Sopln(" Value of a = " + a);
}
public void add(int a, int b)
{
 int c = a+b;
 Sopln(" Sum = " + c);
}
public void display()
{
 Sopln("I am the method of class B");
}
}
class Main
{
public static void main(String args[]);
{
  D ob = new D();
  ob.show();
  ob.display();
  ob.add(10, 12);
}
}
```

Default Method in Interface :-

```
interface Drawable
{
   void draw();
   default void msg()
      {
        Sopln("default method");
      }
}
   class Rectangle implements Drawable
    {
      public void draw()
        {
          Sopln("drawing rectangle");
        }
      }
        class Main
          {
             public static void main(String args[])
             Drawable d = new Rectangle();
             d.draw();
             d.msg();
          }
        }
```

/* Output

drawing rectangle
default method

// an interface can include static method.

```
interface Drawable
{
void draw();
static int cube(int x)
 {
   return x * x * x;
 }
}
class Rectangle implements Drawable
  {
     public void draw()
       {
        Sopln("drawing rectangle");
      }
   }
    class Main
     {
     public static void main(String args[])
       {
         Drawable d = new Rectangle();
          d.draw();
          Sopln(Drawable.cube(3));
       }
   }
```

/* Output

drawing rectangle
27