

Interfaces

=====

- Another way to achieve abstraction in Java, is with interfaces.
- Like a class, An interface can have methods and variables, but the methods declared in an interface are by default abstract (only method signature, no body).
- To achieve interface java provides a keyword called implements.
- It is used to achieve abstraction and Multiple Inheritance in JAVA.

Defining an Interface:-

- The interface keyword is used to declare an interface.
- All the methods in an interface, are declared with the empty body.
- All the fields are Public, static and final by default.

- A class that implements an interface must implement all the methods declared in the interface.

Syntax:-

```
interface <interface_name>
{
    // declare fields
    // declare methods that abstract by default
}
```

- An interface can contain any number of methods.
- An interface is written in a file with a .java extension.
- Each method in an interface is also abstract, so the ~~abstract~~ keyword is not needed.

Example:-

```

interface Animal
{
    public void eat ();
    public void sleep ();
}

```

or

```

// A simple interface
interface Player
{
    final int id = 10;
    public void move ();
}

```

* Implementing Interfaces:-

- A class uses the implements keyword to implement an interface.
- The implements keyword appears in the class declaration following the extends portion of declaration.

Example:- To implement an interface, we use keyword :- Implements

//_

```
interface Animal
```

```
{
```

```
    public void eat();
```

```
    public void sleep();
```

```
}
```

```
public class Main implements Animal
```

```
{
```

```
    public void eat()
```

```
    {
```

```
        System.out.println("Animal is eating something");
```

```
    }
```

```
    public void sleep()
```

```
    {
```

```
        System.out.println("Animal is sleeping");
```

```
    }
```

```
public static void main (String args [])
```

```
{
```

```
    Main obj = new Main();
```

```
    obj.eat();
```

```
    obj.sleep();
```

```
}
```

```
}
```

```
Output :-  
-----
```

```
Animal is eating something  
Animal is sleeping.
```