

*Accredited by NBA - AICTE and Accredited by NAAC - UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai*

Department of Computer Applications

Course Code: 23CAT606

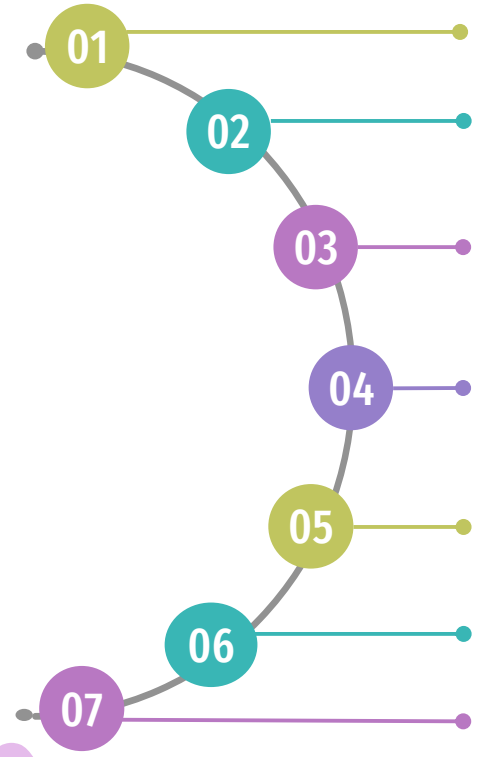
Course Name: Java Programming

Unit I: Java Fundamentals

Topic 5: Interface



Recall



Package

Types

Built in Package

User defined Package

Accessing the Package

Example

Assessment

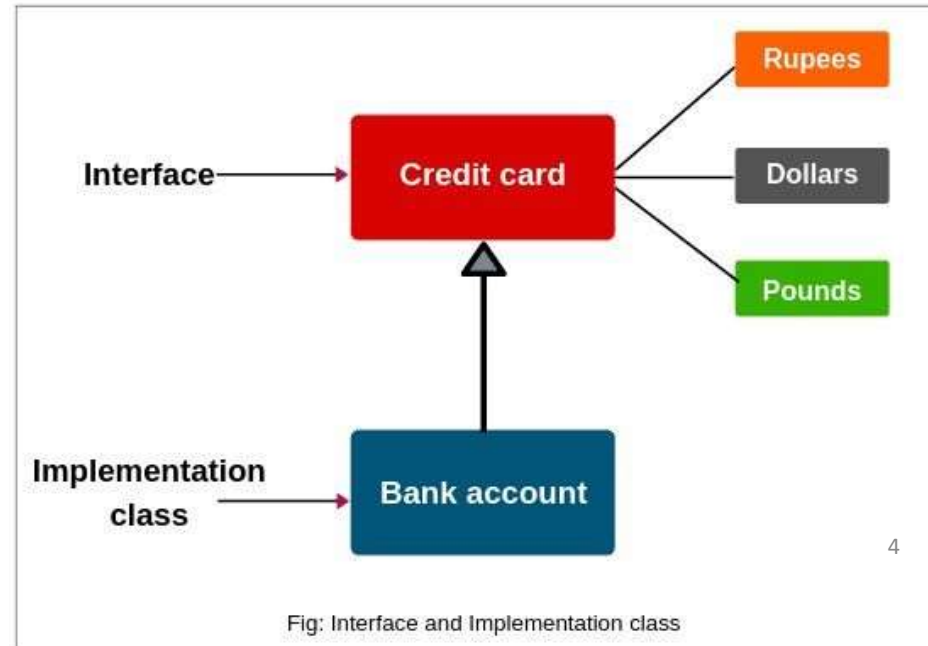


INTERFACE





Use of Interface in Java Application with Realtime Examples



4



INTRODUCTION TO INTERFACE

- An **interface in java** is a blueprint of a class.
- It has static constants and abstract methods.
- The interface in java is a **mechanism to achieve abstraction.**
- There can be only abstract methods in the java interface not method body.
- It cannot be instantiated just like abstract class.
- An interface is not extended by a class; it is implemented by a class.
- An interface can extend multiple interfaces.



DECLARING INTERFACE

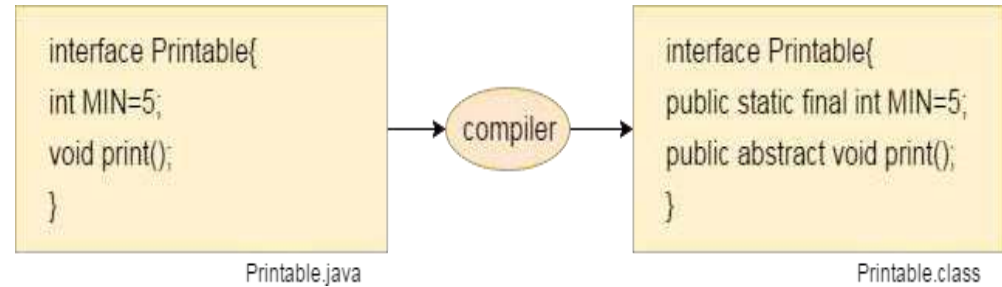
- The **interface** keyword is used to declare an interface.
- Syntax:

```
public interface NameOfInterface
{
    // Any number of final, static fields
    // Any number of abstract method declarations
}
```



Interface

- An interface is implicitly abstract.
- So not need to use the **abstract** keyword while declaring an interface.
- Each Fields in an interface is also implicitly **static** and **final**, so the static and final keyword is not needed (Refer below diagram).
- Methods in an interface are also implicitly public.





Interface example

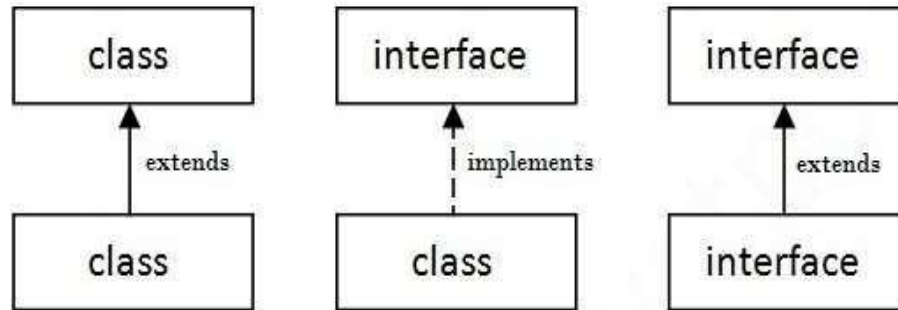
Example:

```
Interface ItemConstants           //interface declared
{
    int code = 1001;              // Variable declared in interface
    string name = "Fan";
    void display();               // Method declared in interface
}
```




Extending interface

- An interface can extend another interface like class.
- The **extends** keyword is used to extend an interface.
- A **class implements an interface**.



```
interface Name2 extends Name1
{
    Body of Name2
}
```



Implementing interface

- Interfaces are used as “Super classes” whose properties are inherited by classes.
- syntax

```
Class className implements interfacename  
{  
    Body of classname  
}
```

1
0



Implementing Interface

Example:

```
interface Drawable           // Interface declared
{
void draw();
}
class Rectangle implements Drawable //implementing
{
public void draw()
{
System.out.println("drawing rectangle");
}}
1
1
```

OUTPUT: drawing rectangle



Implementing Interface

```
interface Printable
{
void print();
}
interface Showable
{
void show();
}
class A7 implements Printable, Showable

{
public void print()
{
System.out.println("Hello");
}

public void show()
{
```

```
System.out.println("Welcome");
}

public static void main(String args[])
{
A7 obj = new A7(); obj.print();

obj.show();
} }
```

OUTPUT:

```
Hello
Welcome1
2
```



Reference

1. Herbert Schildt “ The Complete Reference Java 2, 8th edition , Tata McGraw Hill, 2011
2. Ralph Bravaco, Shai Simonson, “Java Programming: From the Ground up Tata McGraw Hill, 2012
3. <https://www.scientecheasy.com/2020/06/packages-in-java.html/#2> Predefined Packages in Java Built-in Packages
4. <https://www.scientecheasy.com/2019/06/java-interface-use.html/>



Summary

①

②

③

