

Experiment No 7

Write a Java program which implements interface.

Objectives:- To learn the use of interface in java.

Software Used:- Eclipse

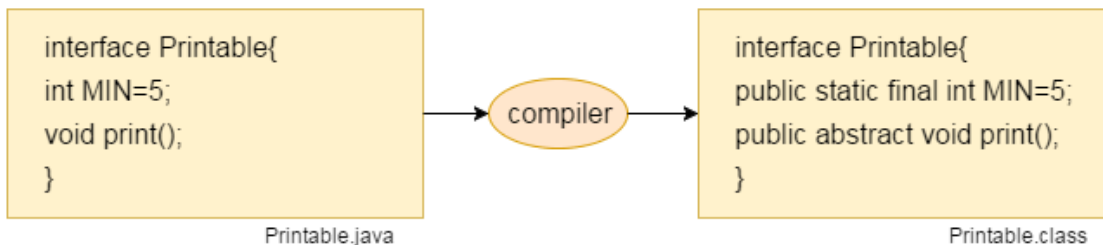
Theory:-

Interface:-

A Java interface is a collection of 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 is used to achieve abstraction and multiple inheritance in Java.

Interface fields are public, static and final by default, and the methods are public and abstract .



An interface is declared by using the interface keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and 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:

1. **interface** <interface_name>
2. {
- 3.
4. // declare constant fields
5. // declare methods that abstract
6. // by default.
- }

An interface which is declared inside another interface or class is called nested interface.

Here are the key points to remember about interfaces:

- 1) We can't instantiate an interface in java. That means we cannot create the object of an interface
- 2) Interface provides full abstraction as none of its methods have body. On the other hand abstract class provides partial abstraction as it can have abstract and concrete(methods with body) methods both.
- 3) implements keyword is used by classes to implement an interface.
- 4) While providing implementation in class of any method of an interface, it needs to be mentioned as public.
- 5) Class that implements any interface must implement all the methods of that interface, else the class should be declared abstract.
- 6) Interface cannot be declared as private, protected or transient.
- 7) All the interface methods are by default abstract and public.
- 8) Variables declared in interface are public, static and final by default.

```
interface Try
{
    int a=10;
    public int a=10;
    public static final int a=10;
    final int a=10;
    static int a=0;
}
```

All of the above statements are identical.

- 9) Interface variables must be initialized at the time of declaration otherwise compiler will throw an error.

```
interface Try
{
    int x;    //Compile-time error
}
```

Above code will throw a compile time error as the value of the variable x is not initialized at the time of declaration.

- 10) Inside any implementation class, you cannot change the variables declared in interface because by default, they are public, static and final. Here we are implementing the interface "Try" which

has a variable x. When we tried to set the value for variable x we got compilation error as the variable x is public static final by default and final variables can not be re-initialized.

```
class Sample implements Try{
    public static void main(String args[])
    {
        x=20; //compile time error
    }
}
```

11) An interface can extend any interface but cannot implement it. Class implements interface and interface extends interface.

12) A class can implement any number of interfaces.

13) If there are two or more same methods in two interfaces and a class implements both interfaces, implementation of the method once is enough.

```
interface A
{
    public void aaa();
}
interface B
{
    public void aaa();
}
class Central implements A,B
{
    public void aaa()
    {
        //Any Code here
    }
    public static void main(String args[])
    {
        //Statements
    }
}
```

Program:

Output:

Conclusion:-

Questions-

- 1) Can an interface extend a class?**
- 2) Can you achieve multiple inheritance through classes in Java?**
- 3) How can you achieve runtime polymorphism using interfaces?**
- 4) Can an interface have a constructor? Explain**