



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution



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

COURSE NAME : 19CST102 & Object Oriented Programming

I YEAR/ II SEMESTER

UNIT – III INHERITANCE AND POLYMORPHISM

Topic: Interface

P.Poonkodi

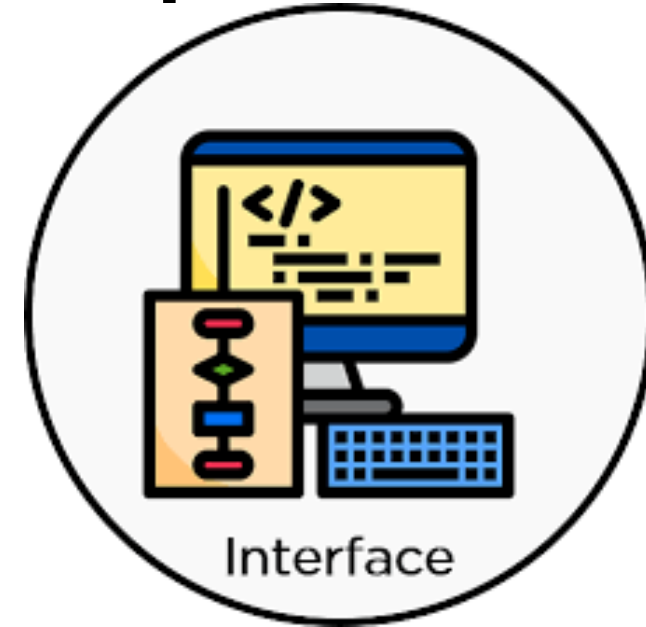
Assistant Professor

Department of Computer Science and Engineering





Real time example





Introduction



- blueprint of a class
- It has static constants and abstract methods
- mechanism to achieve abstraction through multiple inheritance
- only abstract methods in the Java interface, not method body
- **represents the IS-A relationship**





Interface Declaration



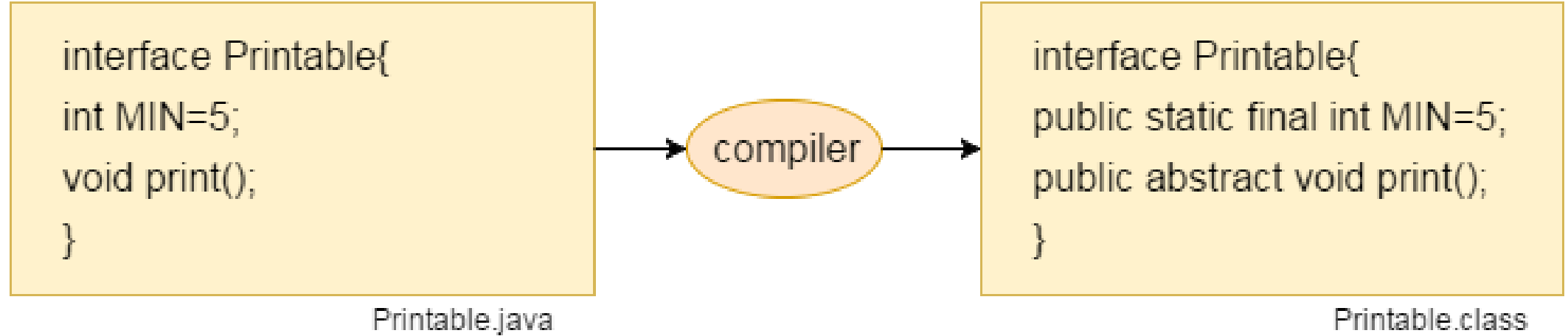
- An interface is declared by using the **interface** keyword
- provides total abstraction
 - methods in an interface are declared with the **empty body**, and all the fields are **public, static** and **final** by default
- Syntax

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





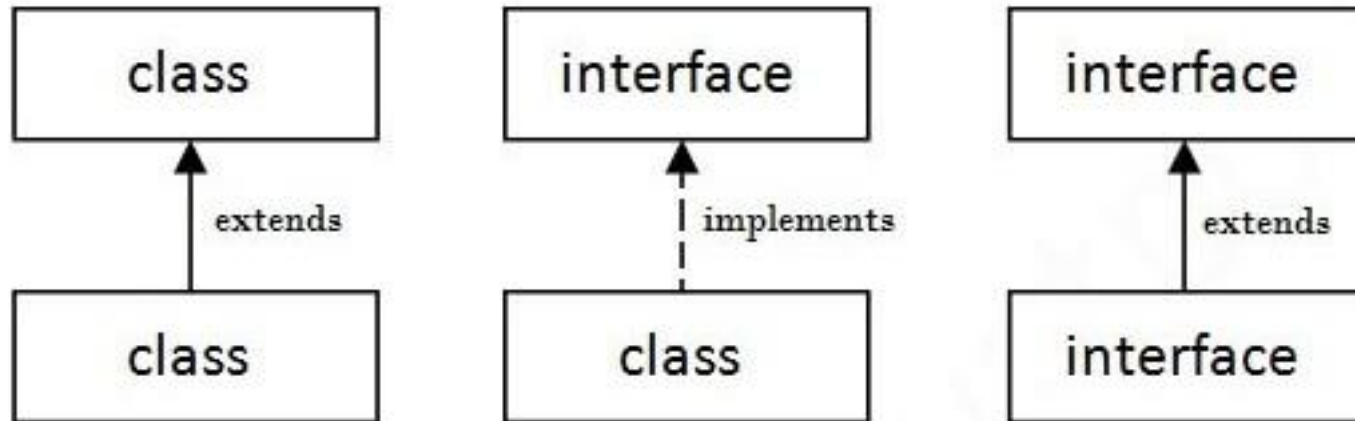
Internal addition by the compiler





Relationship between classes and interfaces

- a class extends another class
- an interface extends another interface
- but a **class implements an interface**





Example



```
interface printable
{
    void print();
}
class A implements printable
{
    public void print(){System.out.println("Hello");
}
public static void main(String args[])
{
    A obj = new A();
    obj.print();
}
}
```





Interface Example: Drawable

```
interface Drawable
{
    void draw();
}
class Rectangle implements Drawable
{
    public void draw()
    {
        System.out.println("drawing rectangle");
    }
}
class Circle implements Drawable
{
    public void draw()
    {
        System.out.println("drawing circle");
    }
}
```





Interface Example: Drawable

```
class TestInterface1
{
public static void main(String args[])
{
    Drawable d=new Circle();
    //In real scenario, object is provided by method e.g. getDrawable()
    d.draw();
}
}
```

Output

drawing circle





Example

```
interface A
{
void a();
void b();
void c();
void d();
}
abstract class B implements A
{
    public void c()
    {
        System.out.println("I am C");
    }
}
class M extends B
{
public void a()
{
    System.out.println("I am a");
}
}
```

```
public void b()
{
System.out.println("I am b");
}
public void d()
{
System.out.println("I am d");
}
}

class Test5
{
    public static void main(String args[])
    {
        A a=new M();
        a.a();
        a.b();
        a.c();
        a.d();
    }
}
```





References



- Java : the complete Reference (Eleventh Edition), Herbert Schildt, 2018.



