INTERFACE

Prepared by

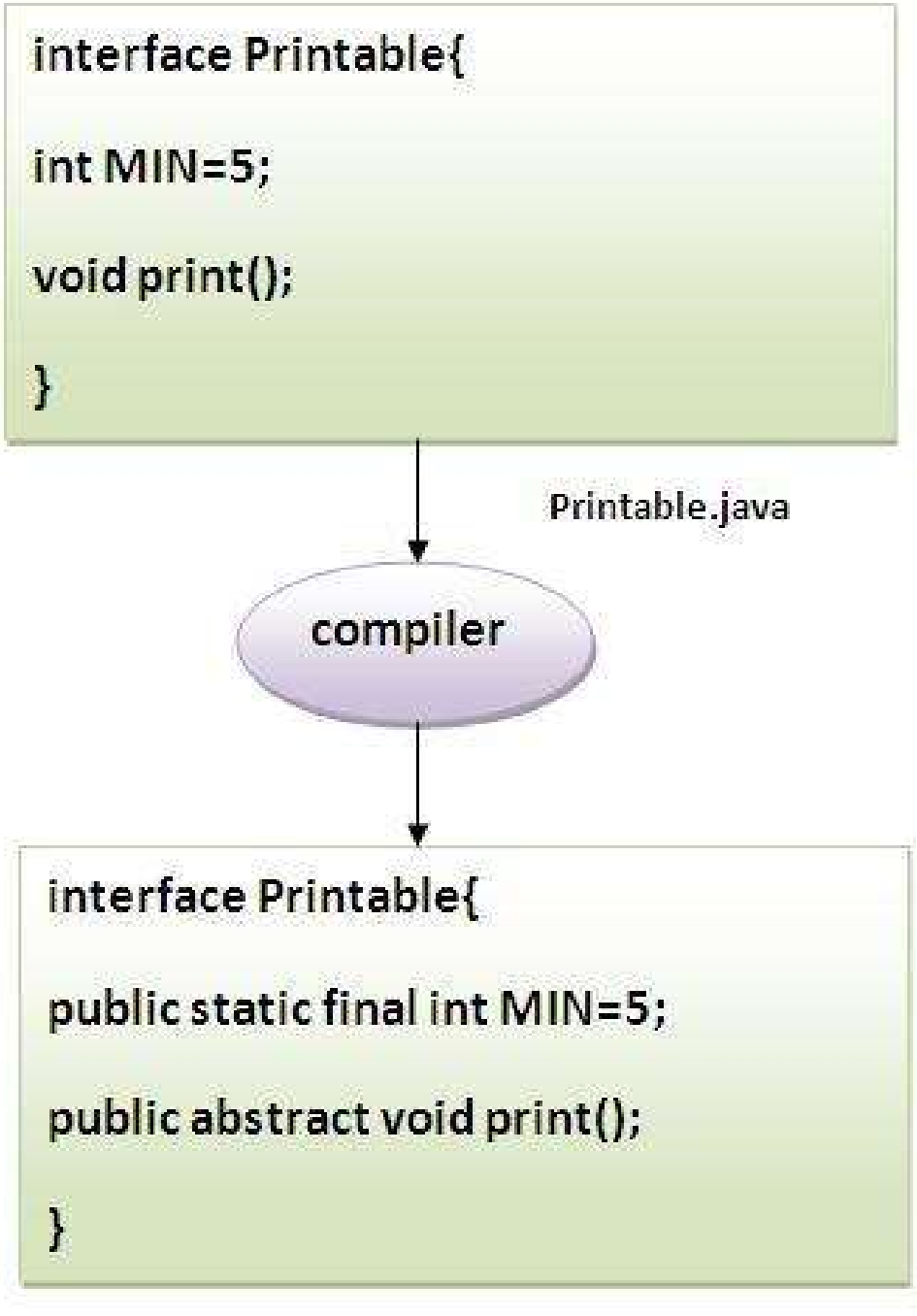
Dr. Rajesh Kumar Ojha
Asst. Prof., CSE, Silicon University

- An **interface in java** is a blueprint of a class. It has static constants and abstract methods only.
- The interface in java is **a mechanism to achieve fully abstraction**. There can be only abstract methods in the java interface not method body. It is used to achieve fully abstraction and multiple inheritance in Java.
- Java Interface also **represents IS-A relationship**.
- It cannot be instantiated just like abstract class.

Why use Java interface?

There are mainly three reasons to use interface. They are given below.

- It is used to achieve fully abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.


interface Printable{
int MIN=5;
void print();
}
Printable.java
compiler
interface Printable{
public static final int MIN=5;
public abstract void print();
}
Printable.class

Syntax:

```
interface interface_name();
```

Example:

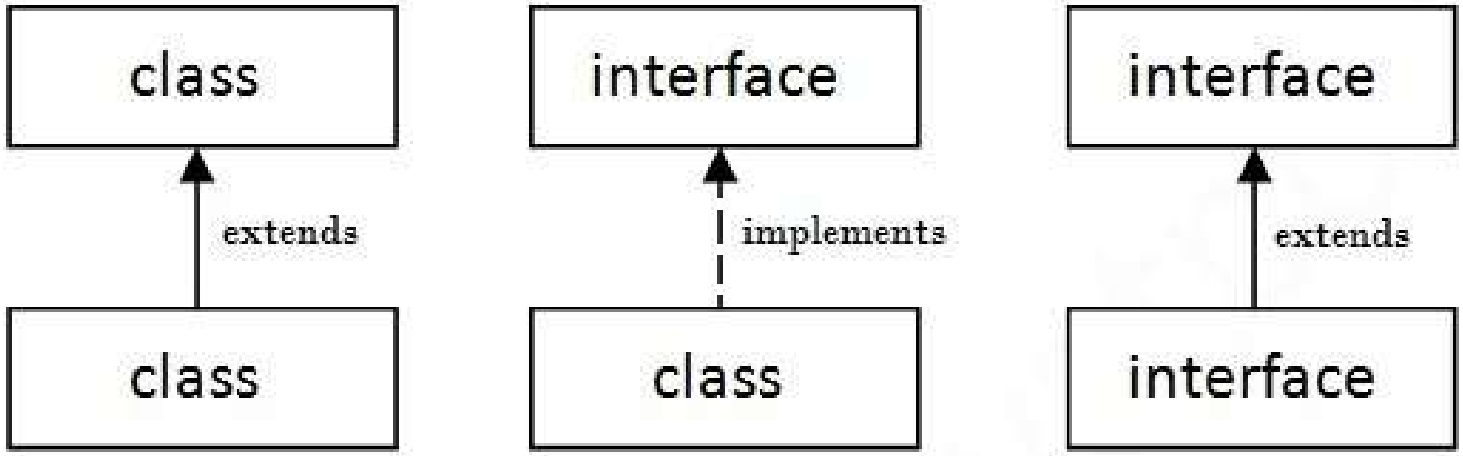
```
interface Moveable
{
   int AVERAGE-SPEED=40;
   void move();
}
```

Rules for using Interface

1. Methods inside Interface must not be static, final, native or strictfp.
2. All variables declared inside interface are implicitly public static final variables(constants).
3. All methods declared inside Java Interfaces are implicitly public and abstract, even if you don't use public or abstract keyword.
4. Interface can extend one or more other interface.
5. Interface cannot implement a class.
6. Interface can be nested inside another interface.

Relationship between classes and interfaces

As shown in the figure given below, a class extends another class, an interface extends another interface but a **class implements an interface**.



Example of Interfaces

Example

```
/* File name : Animal.java */
interface Animal {
public void eat();
 public void travel();
}
```

```
/* File name : MammalInt.java */
public class MammalInt implements
Animal{
 public void eat(){
System.out.println("Mammal eats");
}
public void travel(){
System.out.println("Mammal
travels");
 } public int noOfLegs(){ return 0; }
public static void main(String args[]){
MammalInt m = new MammalInt();
m.eat(); m.travel(); } }
```

O/P: Mammal eats

Mammal travels

Example

```
interface printable{
void print();
}

class A6 implements printable{
public void print(){System.out.println("Hello");}

public static void main(String args[]){
A6 obj = new A6();
obj.print();
 }
}
```

O/P: Hello

Extending Interfaces:

An interface can extend another interface, similarly to the way that a class can extend another class. The **extends** keyword is used to extend an interface, and the child interface inherits the methods of the parent interface.

Ex:-

```
interface NewsPaper {
news();
}
interface Magazine extends NewsPaper {
colorful();
}
```

Example

```
 public interface Sports {
public void setHomeTeam(String name);
public void setVisitingTeam(String name);
 }
//Filename: Football.java
public interface Football extends Sports {
public void homeTeamScored(int points);
public void visitingTeamScored(int points);
public void endOfQuarter(int quarter);
}
```

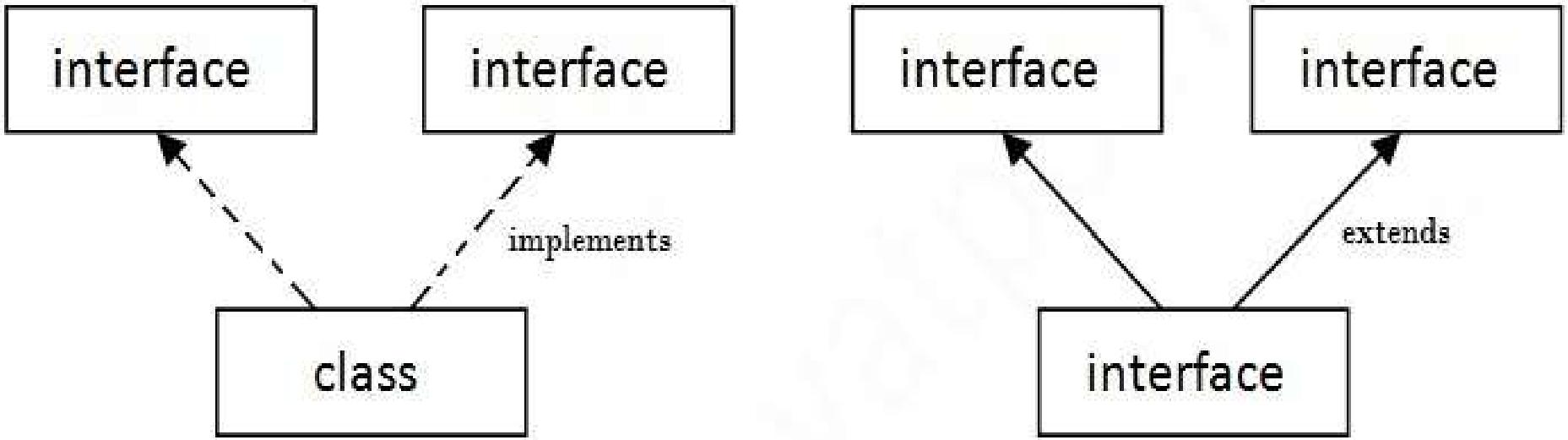
Example Contd..

```
//Filename: Hockey.java
public interface Hockey extends Sports {
public void homeGoalScored();
public void visitingGoalScored();
public void endOfPeriod(int period);
public void overtimePeriod(int ot);
}
```

The Hockey interface has four methods, but it inherits two from Sports; thus, a class that implements Hockey needs to implement all six methods. Similarly, a class that implements Football needs to define the three methods from Football and the two methods from Sports.

Multiple inheritance in Java by interface

- If a class implements multiple interfaces, or an interface extends multiple interfaces i.e. known as multiple inheritance.



Multiple Inheritance in Java

Example

```
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();
 }
}
```

Interface inheritance

```
interface Printable{
void print();
}
interface Showable extends Printable{
void show();
}
class Testinterface2 implements Showable{

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

public static void main(String args[]){
Testinterface2 obj = new Testinterface2();
obj.print();
obj.show();
 }
}
```

Thank you