Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

SAIRAM DIGITAL RESOURCES

YEAR II SEM III

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to EEE, CSE, EIE, ICE, IT)

UNIT NO 2

INHERITANCE AND INTERFACES

2.6 IMPLEMENTING INTERFACE AND EXTENDING INTERFACES

COMPUTER SCIENCE & ENGINEERING

IMPLEMENTING INTERFACE

- Implementing an Interface Once an interface has been defined, one or more classes can implement that interface.
- To implement an interface, **the 'implements' clause** is included in a class definition and then the methods defined by the interface are created.

Properties of java interface

• If a class implements more than one interface, the interfaces are separated with a comma.

• If a class implements two interfaces that declare the same method, then the same method will be used by clients of either interface.

• The methods that implement an interface must be declared public.

• The type signature of the implementing method must match exactly the type signature specified in the interface definition.

RULES AND SYNTAX FOR IMPLEMENTING INTERFACE

Rules

- A class can implement more than one interface at a time.
- A class can extend only one class, but can implement many interfaces.
- An interface can extend another interface, in a similar way as a class can extend another class.

Syntax:

```
class classname implements interface1, interface2,…interface n
{
    // Methods
}
```

PROGRAM FOR IMPLEMENTING INTERFACE

The following code implements an interface Drawable

```
interface Drawable
{
        void draw();
        int cube(int x){return x*x*x;}
}
class Rectangle implements Drawable
{
        public void draw()
        {
                System.out.println("drawing rectangle");
        }
        }
        public static void main(String args[])
        {
                Drawable d=new Rectangle();
               d.draw();
               System.out.println(Drawable.cube(3));
       }
```

OUTPUT: drawing rectangle

IMPLEMENTATION OF MULTIPLE INTERFACES

```
// Implementation of multiple interfaces java example
interface Flyable
{
        void fly();
}
interface Eatable
{
        void eat();
}
// Bird class will implement both interfaces
class Bird implements Flyable, Eatable
{
        public void fly()
        {
                System.out.println("Bird flying");
        }
        public void eat()
         {
                System.out.println("Bird eating");
        }
        // It can have more own methods.
}
public static void main(String[] args)
{
                        Bird b = new Bird();
                        b.eat();
                        b.fly();
} }
```

OUTPUT:
Bird flying
Bird eating

IMPLEMENTING MULTIPLE INTERFACE

In the following example, we have two interfaces : Motorbike and Cycle. Motorbike interface consists of the attribute speed. The method is totalDistance(). Cycle interface consists of the attribute distance() and the method speed(). Both these interfaces are implemented by the class TwoWheeler.

```
interface MotorBike
 {
        int speed=50;
        public void totalDistance();
}
interface Cycle
{
        int distance=150;
        public void speed()
 }
public class TwoWheeler implements MotorBike,Cycle
{
        int totalDistance;
        int avgSpeed;
        public void totalDistance()
         {
                totalDistance=speed*distance;
               System.out.println("Total Distance Travelled : "+totalDistance);
         }
```

IMPLEMENTING MULTIPLE INTERFACE(cont…)

```
public void speed()
{
        int avgSpeed=totalDistance/speed;
        System.out.println("Average Speed maintained : "+avgSpeed);
 }
public static void main(String args[])
 {
TwoWheeler t1=new TwoWheeler();
t1.totalDistance();
t1.speed();
}
 }
```

OUTPUT:

Total Distance Travelled : 7500
Average Speed maintained : 150

EXTENDING INTERFACES

- An interface can extend another interface in the same 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.

SYNTAX:

```
interface interface1{ }
interface interface2 extends interface1{}
interface interface3 extends interface1{}
```

EXTENDING INTERFACES

```
class Shape
{
        public void display()
        {
                System.out.println("Inside display");
        }
        }
class Rectangle extends Shape
{
        public void area()
        {
                System.out.println("Inside area");
        }
}
class Cube extends Rectangle
{
  public void volume()
{
     System.out.println("Inside volume");
} }
```

```
public class Tester
{
        public static void main(String[] arguments)
        {
                Cube cube = new Cube();
                cube.display();
                cube.area();
                cube.volume();
        }
        }
```

OUTPUT:

Inside display
Inside area
Inside Volume

VIDEO LINK

https://www.youtube.com/watch?v=HxuJpXS5CPI