

Interfaces

Direct Challenges

1. Create an interface Drawable with method draw().

```
interface Drawable {  
    void draw();  
}  
  
class Circle implements Drawable {  
    public void draw() {  
        System.out.println("Drawing a Circle");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Circle c = new Circle();  
        c.draw();  
    }  
}
```

Output:

Drawing a Circle

2. Implement the interface in class Circle.

```
interface Drawable {  
    void draw();  
}  
  
class Circle implements Drawable {  
    public void draw() {  
        System.out.println("Drawing a Circle");  
    }  
}
```

```
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Circle c = new Circle();  
        c.draw();  
    }  
}
```

Output:

Drawing a Circle

3. Create another interface Colorable and implement both interfaces in a single class.

```
interface Drawable {  
    void draw();  
}
```

```
interface Colorable {  
    void fillColor();  
}
```

```
class Square implements Drawable, Colorable {  
    public void draw() {  
        System.out.println("Drawing a Square");  
    }  
    public void fillColor() {  
        System.out.println("Filling Square with Blue color");  
    }  
}
```

```
public class Main {
```

```

    public static void main(String[] args) {
        Square s = new Square();
        s.draw();
        s.fillColor();
    }
}

```

Output:

Drawing a Square

Filling Square with Blue color

Scenario-Based Challenges

1. Design an interface Payable and implement it in classes Invoice and Employee.

```

interface Payable {
    void pay();
}

```

```

class Invoice implements Payable {
    public void pay() {
        System.out.println("Paying invoice...");
    }
}

```

```

class Employee implements Payable {
    public void pay() {
        System.out.println("Paying employee salary...");
    }
}

```

```

public class Main {
    public static void main(String[] args) {
        Payable p1 = new Invoice();
        Payable p2 = new Employee();
    }
}

```

```
        p1.pay();
        p2.pay();
    }
}
```

Output:

Paying invoice...

Paying employee salary...

2. Create an interface Database with connect() method and implement for MySQL and Oracle.

```
interface Database {
    void connect();
}
```

```
class MySQL implements Database {
    public void connect() {
        System.out.println("Connecting to MySQL database...");
    }
}
```

```
class Oracle implements Database {
    public void connect() {
        System.out.println("Connecting to Oracle database...");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Database db1 = new MySQL();
        Database db2 = new Oracle();
        db1.connect();
        db2.connect();
    }
}
```

```
}
```

```
}
```

Output:

Connecting to MySQL database...

Connecting to Oracle database...