

Java

Interface

- Interface can have only abstract methods before java 8
- Default methods are introduced that can have a body
- The use of default method in interface is for example if the interface is implemented in three classes

ode:

```
interface Details{
    void personalDetails();

    default void generalDetails() {
        String company_name = "Aspire Systems";
        System.out.println("Company Name : "+company_name);
        System.out.println("-----");
    }

    static void permanentDetails() {
        System.out.println("Company Location : Chennai");
    }
}

public class Employee implements Details {

    @Override
    public void personalDetails() {
        String name ="Gautam";
        String role = "Testing";
        System.out.println("Personal Details");
        System.out.println("Name : "+name);
        System.out.println("Role : "+role);
    }

    @Override
    public void generalDetails() {
        String company_name = "Welcome to Aspire Systems";
        System.out.println(company_name);
        System.out.println("-----");
    }
}
```

```
}
```

```
public static void main(String args[]) {
```

```
Employee employee = new Employee();
```

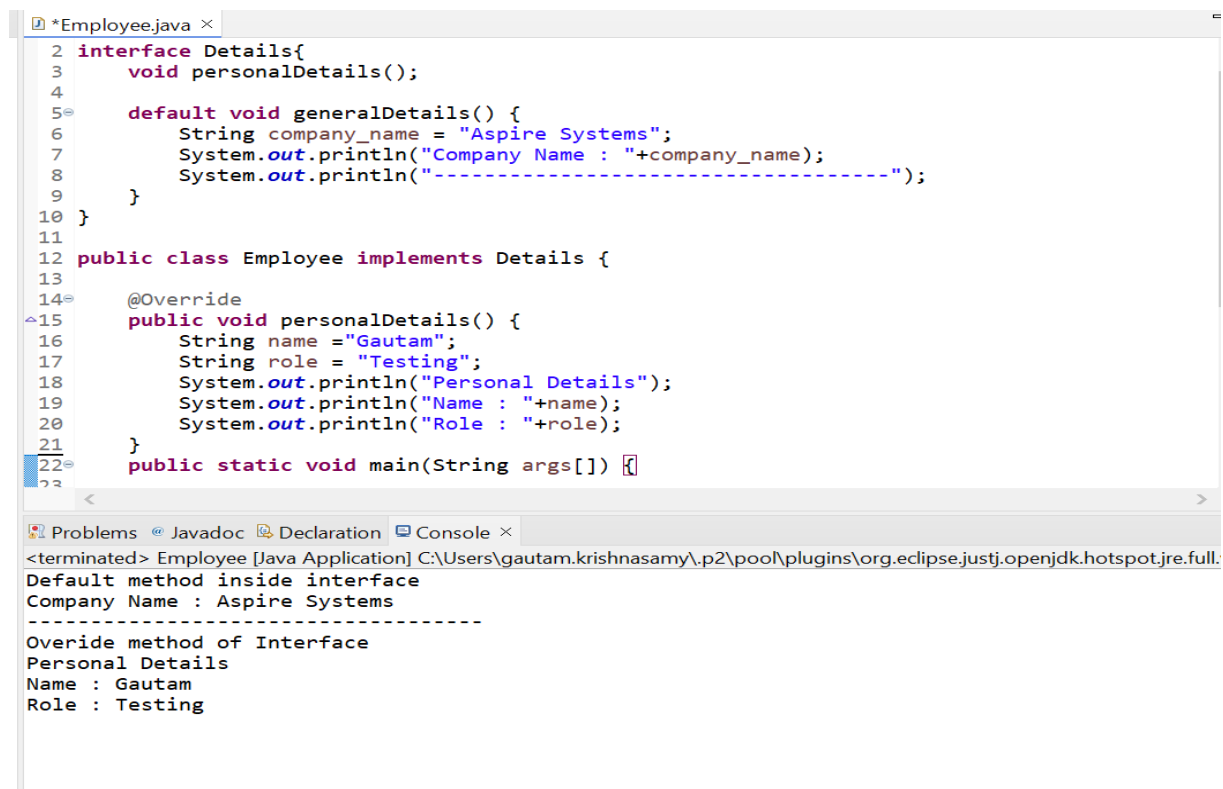
```
System.out.println("Default method inside interface");  
employee.generalDetails();
```

```
System.out.println("Override method of Interface");  
employee.personalDetails();
```

```
}
```

```
}
```

Output



The screenshot shows the Eclipse IDE with a Java project. The editor displays the source code for `*Employee.java`. The code defines an interface `Details` with a default method `generalDetails()` and a class `Employee` that implements `Details` and overrides `personalDetails()`. The `main` method calls both `generalDetails()` and `personalDetails()`.

```
1 interface Details{  
2     void personalDetails();  
3  
4  
5     default void generalDetails() {  
6         String company_name = "Aspire Systems";  
7         System.out.println("Company Name : "+company_name);  
8         System.out.println("-----");  
9     }  
10 }  
11  
12 public class Employee implements Details {  
13  
14     @Override  
15     public void personalDetails() {  
16         String name = "Gautam";  
17         String role = "Testing";  
18         System.out.println("Personal Details");  
19         System.out.println("Name : "+name);  
20         System.out.println("Role : "+role);  
21     }  
22     public static void main(String args[]) {  
23
```

The console output shows the execution results:

```
<terminated> Employee [Java Application] C:\Users\gautam.krishnasamy\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.  
Default method inside interface  
Company Name : Aspire Systems  
-----  
Override method of Interface  
Personal Details  
Name : Gautam  
Role : Testing
```

Overriding the Default Method

The default method in the interface can be override in the implemented class

```
@Override
public void generalDetails() {
    String company_name = "Welcome to Aspire Systems";
    System.out.println(company_name);
    System.out.println("-----");
}
```