

BITS PILANI, DUBAI CAMPUS
DUBAI INTERNATIONAL ACADEMIC CITY, DUBAI

FIRST SEMESTER 2025 – 2026

COURSE: CSF213/ECOM213/MACF212 (Object Oriented Programming)

COMPONENT: Lab 10

Week: 11

Aim:

To learn how to use interfaces, multiple inheritance, and object comparison in Java with the Comparable and Comparator interfaces, and to show how to sort and reuse class functions.

Objective:

To learn how to define and use interfaces, use interfaces to get multiple inheritance, and use the Comparable and Comparator interfaces to sort objects in a natural or custom order. This shows how to reuse code, abstract it, and manage objects in a flexible way.

Problem Statements:

1. Write a Java program that shows how to use interfaces to show the idea of multiple inheritance. Make two interfaces: WiFiEnabled and BluetoothEnabled. WiFiEnabled has a method called connectToWiFi(String networkName) and BluetoothEnabled has a method called connectToBluetooth(String deviceName). Then, make a class called SmartSpeaker that implements both interfaces and gives the method definitions that are needed. The class should also have a method called playMusic(String songName) that plays music on the smart speaker. In the main method, make a SmartSpeaker object and show how to connect to a Wi-Fi network, a Bluetooth device, and play a song. This problem shows how Java uses interfaces to let classes inherit from more than one class.

Expected Input:

```
WiFi Network Name: Home_Network
Bluetooth Device Name: John's Phone
Song Name: Shape of You
```

Expected Output:

```
Connected to WiFi network: Home_Network
Connected to Bluetooth device: John's Phone
Playing song: Shape of You
```

2. Write a Java program that sorts a group of objects by a certain property using the Comparable interface. Make a class called Book that has the title, author, and price as its properties. Add the Comparable interface to the Book class and change the compareTo() method so that it sorts books by their price in ascending order. In the main method, make an array or list of several Book objects with different prices. Then, use the Collections.sort() method to sort the list based on the natural order that the compareTo() method sets. Lastly, show the sorted list of books with all of their information. This problem shows how to use Comparable to set a default order for classes that users create in Java.

Expected Input:

```
Book 1: Title = "Java Basics", Author = "John Doe", Price = 350.0
Book 2: Title = "Data Structures", Author = "Smith", Price = 450.0
Book 3: Title = "Advanced Java", Author = "Jane", Price = 550.0
```

Expected Output:

```
Books sorted by price (ascending):  
Title: Java Basics Author: John Doe Price: 350.0  
Title: Data Structures Author: Smith Price: 450.0  
Title: Advanced Java Author: Jane Price: 550.0
```

3. Write a Java program that shows how to use the Comparator interface to sort objects in different ways. Make an Employee class that has the following properties: id, name, and age. Next, make two different comparator classes: one called NameComparator that sorts employees by name in alphabetical order, and another called AgeComparator that sorts employees by age in ascending order. Using the Collections.sort() method and the two comparator classes, sort the list of employees first by name and then by age in the main method. This program should make it clear how the Comparator interface lets you set up multiple ways to sort the same class without changing how it works.

Expected Input:

```
Employee 1: ID = 1, Name = Mike, Age = 28  
Employee 2: ID = 2, Name = Alex, Age = 30  
Employee 3: ID = 3, Name = John, Age = 25
```

Expected Output:

```
Sorting by Name:  
ID: 2 Name: Alex Age: 30  
ID: 3 Name: John Age: 25  
ID: 1 Name: Mike Age: 28
```

```
Sorting by Age:  
ID: 3 Name: John Age: 25  
ID: 1 Name: Mike Age: 28  
ID: 2 Name: Alex Age: 30
```