Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**10**
Student Name:	**Shreyas Sanjay Nanaware**
Roll No :	**39**

Title:

1. Write a java program to implement Multiple Inheritance using Interfaces. Create an interface called Management with selectCandidate() method. Another interface called Department with allotSubject() method. Class called HOD will implements these two interfaces and define the methods and access them with valid objects.

Learning Objective:

Students will be able to implement multiple inheritance using Interface concepts

Learning Outcome:

- Understanding the abstraction concept and hiding of the unnecessary code using interfaces.

Course Outcome:

ECL304.4	1. Implement different programming applications using packaging.

Faculty: Ms. Deepali Kayande

Theory:

- **What is complete abstraction and how is it achieved in JAVA?**

Ans: 1. Abstraction is nothing but the process of hiding the implementation details ad showing only the functionality to the user.

2. There are two ways of achieving this abstraction and they are as follows:

3.First way is to use the abstract class in which we have an option of abstraction percentage to choose from ranging from 0 to all the way up to 100% , it basically means we can keep few thing visible and few thing hidden at the same time according to our convenience and required task.

4. For achieving complete abstraction in Java we have a concept known as 'Interface'.

5. Here we can keep all the things hidden from the end user making our data way more secure.

6. An interface is a fully abstract class. It includes a group of abstract methods (methods without a body). We have to just use the interface as the keyword before the name of the actual class , like for example the syntax is given as follows:

```
Interface Student{
int roll_no;
String name;
void  generate_result();
void display_result();
}
```

7. As we can see from the above example we have just declared the name of the method inside the interface and not the method body which makes it an by default abstract method.

8. Like abstract classes, we cannot create objects of interfaces.
To use an interface, other classes must implement it. We use the implements keyword to implement an interface.

- **Explain multiple abstraction and how is it performed in Java?**

Ans: 1.Interface is similar to Java classes. The difference is only that an interface contains empty methods (methods that do not have method implementation) and variables. In other words, it is a collection of abstract methods (the method that does not have a method body) and static constants.

2. In java we cannot perform multiple inheritance and to overcome this the interface concept comes into the picture.

3.When one class extends more than one classes then this is called multiple inheritance. For example: Class C extends class A and B then this type of inheritance is known as multiple inheritance. Java doesn't allow multiple inheritance.

4.A Java class can only extend one parent class. Multiple inheritance is not allowed. Interfaces are not classes, however, and an interface can extend more than one parent interface. The extends keyword is used once, and the parent interfaces are declared in a comma-separated list.

5.Like in the example given below we can see that first we have created an interface named Management and then another interface named Department and then a class named HOD which is implementing both these interfaces.

```
 interface Management{

}

interface Department{

}

class HOD implements Management,Department {

    }
```

6.This means that a class can implement multiple interfaces and an interface can also implement multiple interfaces. Except these two combinations nothing can be done in Java in the field of multiple abstraction as such.

1. **Write a java program to implement Multiple Inheritance using Interfaces. Create an interface called Management with selectCandidate() method. Another interface called Department with allotSubject() method. Class called HOD will implements these two interfaces and define the methods and access them with valid objects.**

Algorithm :

Step 1: Create an interface named Management with an abstract method selectCandidate()

Step 2: Create another interface named Department with an abstract method allotSubject()

Step 3: Create a class named HOD which is implementing both the interfaces

Step 4: Inside class HOD define the method body of both the abstract methods which helps us to take the user input for the name of the selected candidate and the alloted subject for the selected candidate

Step 5: The class HOD also has its own displaying method through which the output will be displayed

Step 6: In the main function create the object of the class HOD (child class) and using that object we will call the 3 methods for getting the desired output

Program:

```
/*Write a java program to implement Multiple Inheritance using Interfaces.
Create an interface called Management with selectCandidate() method. Another
interface
called Department with allotSubject() method. Class called HOD will
implements these
two interfaces and define the methods and access them with valid objects.
*/
```

```
package javaprogramming3;

import java.util.Scanner;

interface Management{  //creating an interface named Management
        public void selectCandidate();  //abstract method
}

interface Department{  //creating an interface named Department
        public void allotSubject();   //abstract method
}

class HOD implements Management,Department {   //class HOD is
implementing both the interfaces
        Scanner in= new Scanner(System.in);
        String Candidate_name= null;   //attributes of the class
        String Subject_name= null;

        public void selectCandidate() {   //method for taking input from the user
                System.out.println("Enter the name of Candidate :");
                Candidate_name=in.next();
        }

        public void allotSubject() {   //method for taking input from the user
                System.out.println("Enter the alloted subject for the candidate : ");
                Subject_name=in.next();
        }
void displaying_data() {    //method for displaying the output to the user
                System.out.println("\n" + Candidate_name +" has been alloted "+
Subject_name +" as the subject ");
        }
}

public class Lab_10 {
        public static void main(String[] args) {
        HOD xyz = new HOD();   //creating object of the child(HOD) class
        xyz.selectCandidate();  //accessing the methods using the created object
        xyz.allotSubject();
        xyz.displaying_data();
        }
```

}

Input Given:

Name of the candidate: Shreyas

Subject allotted: Mathematics

Output:

```
<terminated> Lab_10 [Java Application] C:\Users\Shreyas\.p2\pool\plugins\orc
Enter the name of Candidate :
Shreyas
Enter the alloted subject for the candidate :
Mathematics

Shreyas has been alloted Mathematics as the subject
```