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:	**Sakshi Morey**
Roll No:	**34**

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.

Theory:

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

1. Data Abstraction is the property by virtue of which only the essential details are displayed to the user.
2. The trivial or the non-essentials units are not displayed to the user.
3. Ex: A car is viewed as a car rather than its individual components.
4. Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details.
5. The properties and behaviours of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

❖ **There are two ways to achieve abstraction in java**

Abstract class (0 to 100%)
Interface (100%)

❖ **Abstract class in Java**

A class which is declared as abstract is known as an **abstract class**. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

❖ **Abstract Method in Java**

A method which is declared as abstract and does not have implementation is known as an abstract method.

Interface in Java

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve* abstraction.

There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

- Explain multiple inheritance and how is it performed in Java?

1. Multiple Inheritance is a feature of an object-oriented concept, where a class can inherit properties of more than one parent class.
2. The problem occurs when there exist methods with the same signature in both the super classes and subclass.
3. On calling the method, the compiler cannot determine which class method to be called and even on calling which class method gets the priority.

Algorithm :	1. START 2. Create interface Management. 3. Create another interface Department. 4. Create class HOD and write all the attributes 5. Then class HOD will extends Management and Department. 6. Create main class and call the objects. 7. END.
Program:	

```
/*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 interface2;
import java.util.*;

interface Management
{
    void selectCandidate();
}

interface Department
{
    void  allotSubject();
}
```

```java
class HOD implements Department, Management
{
    String Candidate;
    String Subject;
    void getdata()
    {
        Scanner t=new Scanner(System.in);
        System.out.println("Enter Candidate name:");
        Candidate=t.next();
        System.out.println("Enter Subject:");
        Subject=t.next();
    }
    public void selectCandidate()
    {
        System.out.println("Candidate Name : "+Candidate );
    }
    public void allotSubject()
    {
        System.out.println("Subject Alloted : "+Subject);
    }
}

class Main
{
    public static void main (String[] args)
    {
        HOD ob = new HOD();
        ob.getdata();
        ob.selectCandidate();
        ob.allotSubject();
    }
}
```

Input given:

Poorva

English

Output Screenshot: