Experiment No. 8
Implement a program on multiple inheritance with interface.
Date of Performance:
Date of Submission:

Aim: Implement a program on multiple inheritance with interface.

Objective: Implement multiple inheritance in a program to perform addition, multiplication and transpose operations on a matrix. Create an interface to hold prototypes of these methods and create a class input to read input. Inherit a new class from this interface and class. In main class create object of this child class and invoke required methods.

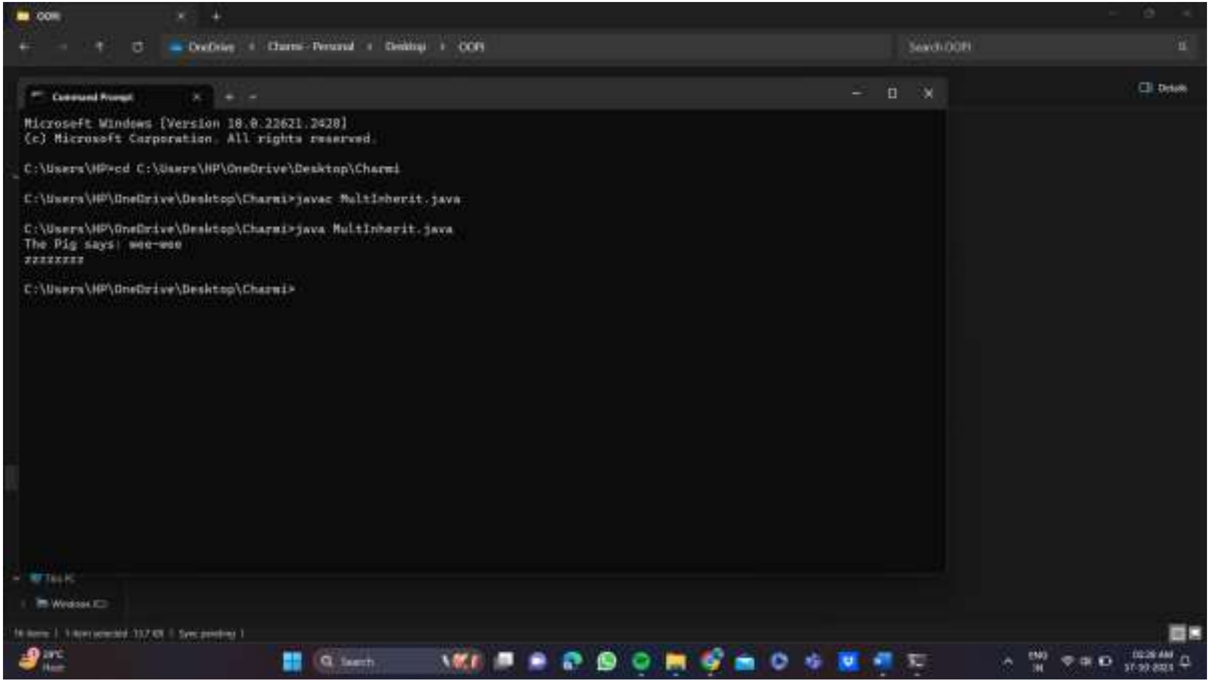
Theory:

- In Multiple inheritance, one class can have more than one superclass and inherit features from all parent classes. Java does not support multiple inheritance with classes. In java, we can achieve multiple inheritance only through Interfaces.
- An interface contains variables and methods like a class but the methods in an interface are abstract by default unlike a class. If a class implements multiple interfaces, or an interface extends multiple interfaces, it is known as multiple inheritance.
- However, Java supports multiple interface inheritance where an interface extends more than one super interfaces.
- A class implements an interface, but one interface extends another interface. Multiple inheritance by interface occurs if a class implements multiple interfaces or also if an interface itself extends multiple interfaces.
- The following is the syntax used to extend multiple interfaces in Java:

```
access_specifier interface subinterfaceName extends superinterface1, superinterface2, …… {
// Body
}
```

Code:

```
class MultInherit{
public static void main(String args[])
{
Pig a=new Pig();
a.animalsound();
a.sleep();
}
}
interface Animal{
public void animalsound();
public void sleep();
}
class Pig implements Animal{
public void animalsound(){
System.out.println("The Pig says: wee-wee");
}
public void sleep(){
System.out.println("zzzzzzzz");
}
}
```



```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\Users\HP\OneDrive\Desktop\Charmi

C:\Users\HP\OneDrive\Desktop\Charmi>javac MultInherit.java

C:\Users\HP\OneDrive\Desktop\Charmi>java MultInherit.java
The Pig says: wee-wee
zzzzzzzz

C:\Users\HP\OneDrive\Desktop\Charmi>
```

Conclusion:

In Java, interfaces are a fundamental concept that allows you to define a contract specifying a set of methods that implementing classes must adhere to. Interfaces allow you to define a contract or a set of methods without specifying the implementation. This promotes abstraction, enabling you to focus on what a class should do rather than how it should do it.

An interface is declared using the `interface` keyword, followed by the interface name and the method signatures. The methods in an interface are declared without an implementation, and the classes that implement the interface must provide an implementation for all the methods declared in the interface.

Interfaces are useful for creating loosely coupled systems, where different components can interact with each other through well-defined interfaces. They also enable polymorphism, where objects of different classes can be treated as instances of the same interface.

In summary, interfaces are a powerful tool in Java that allows you to define a contract specifying a set of methods that implementing classes must adhere to. They promote abstraction and enable polymorphism, making them useful for creating loosely coupled systems.