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: oink-oink");
}
public void sleep(){
System.out.println("Sleeping: zzzzzzzz...");
}
}
```

Output:

```
C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>javac MultInherit.java

C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>java MultInherit.java
The Pig says: oink-oink
Sleeping: zzzzzzzz...
```

Conclusion:

Comment on how interface are useful and implemented using java.

- nterface are useful and implemented using Java in the following ways:
- Interface are used to achieve abstraction and multiple inheritance in Java. Abstraction means hiding the implementation details and showing only the functionality. Multiple inheritance means a class can inherit from more than one interface.
- Interface are also used to achieve loose coupling, which means reducing the dependency between the classes. By using interface, we can decouple the classes from the implementation details and make them more flexible and adaptable.
- Interface are declared using the keyword `interface` followed by the interface name and the interface body. The interface body can contain only constants, abstract methods, default methods, static methods, and nested types. All the members of an interface are public by default.
- A class can implement an interface using the keyword `implements` followed by the interface name. A class that implements an interface must provide the implementation for all the abstract methods of the interface. A class can implement multiple interfaces separated by commas.
- An interface can extend another interface using the keyword `extends` followed by the interface name. An interface that extends another interface inherits all the members of the parent interface. An interface can extend multiple interfaces separated by commas.