|  |
| --- |
| 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](https://www.geeksforgeeks.org/java-and-multiple-inheritance/) with classes. In java, we can achieve multiple inheritance only through [Interfaces](http://quiz.geeksforgeeks.org/interfaces-in-java/).
* 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:**

interface Interface1 {

void method1();

}

interface Interface2 {

void method2();

}

class MyClass implements Interface1, Interface2 {

@Override

public void method1() {

System.out.println("Implementing method1");

}

@Override

public void method2() {

System.out.println("Implementing method2");

}

void myClassMethod() {

System.out.println("Additional method in MyClass");

}

}

public class multiple\_inheritance {

public static void main(String[] args) {

MyClass obj = new MyClass();

obj.method1();

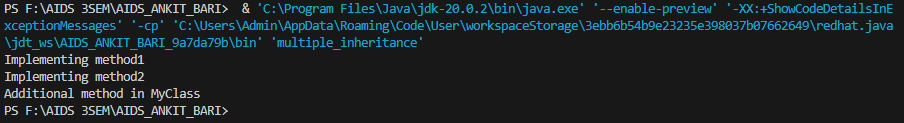
obj.method2();

obj.myClassMethod();

}

}

**OUTPUT:**



## Conclusion:

Multiple inheritance in Java is achieved through interfaces, allowing a class to inherit from more than one interface. This is useful for achieving a higher level of abstraction and code reusability. Each interface can declare a set of methods, and a class implementing those interfaces must provide concrete implementations for all the methods declared in the interfaces. In conclusion, Java's approach to multiple inheritance through interfaces provides a flexible and clean way to design and organize code while avoiding some of the challenges associated with multiple inheritance in other programming languages.