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:

```
public class demo{
public static void main(String args[])
{
Animal a=new Animal();
```

```
a.eat();
a.travel();
}
}
interface AnimalEat {
void eat();
}
interface AnimalTravel {
void travel();
}
class Animal implements AnimalEat , AnimalTravel {
 public void eat()
{
 System.out.println("Animal is eating...");
}
 public void travel()
{
 System.out.println("Animal is travelling...");
}
}
```

OUTPUT

```
C:\Users\yedu0\OneDrive\Documents\java>javac demo.java

C:\Users\yedu0\OneDrive\Documents\java>java demo.java
Animal is eating...
Animal is travelling...
```

Conclusion:

Comment on how interface are useful and implemented using java.

Interfaces in Java are essential for achieving abstraction and providing a way to achieve full abstraction in Java. They define a contract that specifies the capabilities a class must implement. Here are a few reasons why interfaces are useful and how they can be implemented in Java:

Abstraction: Interfaces allow you to create code that is more abstract and flexible. By defining a set of methods without specifying the implementation details, interfaces enable different classes to provide their own implementation. This promotes loose coupling between classes and allows for more modular and maintainable code.

Multiple Inheritance: Java does not support multiple inheritance of classes, but it does support the implementation of multiple interfaces. This means that a class can implement multiple interfaces, allowing it to inherit behaviors from multiple sources. This helps in achieving a higher level of flexibility and code reuse.

Standardization: Interfaces are often used to define a standard for a set of related classes. By implementing the same interface, different classes can adhere to the same contract, ensuring that they all provide the same set of functionalities. This makes it easier to work with different implementations interchangeably.