Name: Gaurav Kishor Patil

Roll no:54 DIV: 2

Batch: C

Experiment No. 10
Multiple Inheritance.
Date of Performance:29/09/23
Date of Submission:05/09/23

Aim:- To implement the concept of Multiple Inheritance.

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.

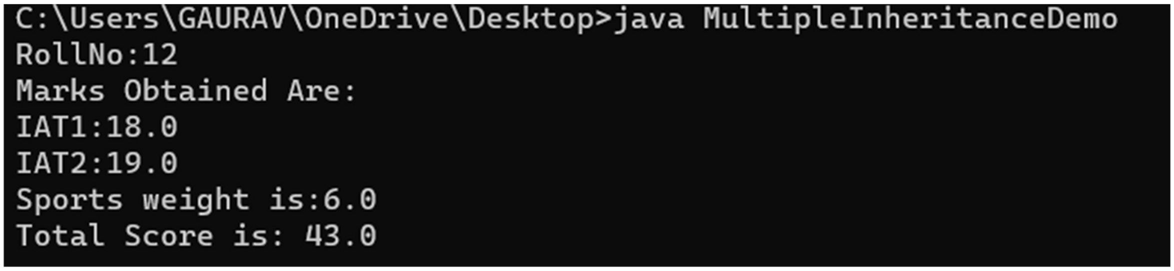
Theory:-

- 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 Student
{
int rollNumber;
void getNumber(int n)
{
rollNumber=n;
}
void putNumber()
{
System.out.println("RollNo:" +rollNumber);
}
}

class Test extends Student{
float IAT1, IAT2;

void getMarks (float m1, float m2){
IAT1=m1;
IAT2=m2;
}
void putMarks()
{
System.out.println("Marks Obtained Are: ");
System.out.println("IAT1:" +IAT1);
System.out.println("IAT2:" +IAT2);
}
}

interface Sports {
    float sportswt = 6.0F;

    void putWt();
}

class Result extends Test implements Sports {
    float total;

    public void putWt() {
        System.out.println("Sports weight is:" + sportswt);
    }

    void display() {
        total = IAT1 + IAT2 + sportswt;
        putNumber();
        putMarks();
        putWt();
        System.out.println("Total Score is: " + total);
    }
}


class MultipleInheritanceDemo{
public static void main(String[] args)

{

Result r1=new Result();
r1.getNumber(12);
r1.getMarks (18.0F, 19.0F);
r1.display();
}

}
```



```
C:\Users\GAURAV\OneDrive\Desktop>java MultipleInheritanceDemo
RollNo:12
Marks Obtained Are:
IAT1:18.0
IAT2:19.0
Sports weight is:6.0
Total Score is: 43.0
```

Conclusion:-

Multiple inheritance in Java is a concept that allows an object to inherit properties from more than one parent class. However, Java does not support multiple inheritance directly through classes to avoid ambiguity and complexity, especially in cases where multiple parent classes have methods with the same name. Instead, Java provides a workaround by using interfaces. An interface is a reference type in Java, similar to a class, that can contain only constants, method signatures, default methods, static methods, and nested types. A class can implement multiple interfaces, thereby achieving the effect of multiple inheritance in a clear and controlled way. This approach maintains the simplicity and robustness of the Java programming language.