- **Name of Teacher:** Miss Nita N. Bargale
- **Class:** B.Sc. Computer Science (Entire)- III
- **Course Title:** Java programming

interface

Java suports multilevel inheritance but not multiple inheritance
You can not define a class as
class A extends B extends C
{
Statements;
}
For this java provides an alternative approach is **interface.**

What is an interface in Java?

An interface can have methods and variables just like the class but the methods declared in interface are by default abstract (only method signature, no body). Also, the variables declared in an interface are public, static & final by default.

- Syntax

access specifire interface <interface_name>

{

   type variable1= value;

   reture_type method_name1(para_list1);

  reture_type method_name1(para_list1);

  default reture_type method_name1(para_list1)

{

Statements;

}

}

- Here access specifier is either public or not used.when no access specifier is included,then the default access results and the interface is only availeble to other members of the package in which it is declared .when it is declared as public, then interface can be used by any other package
- interface keyword is used to define interface.
- Interface_name is the name of inteface
- Since Java 8, interface can have default and static methods

e.g-

```
interface Test
{
Void add(String address);
int  sum (int a,int b);
int  min=10;
}
```

- All the interface methods are by default **abstract and public**.

- Variables declared in interface are **public, static and final** by default.

- We can't instantiate an interface in java. That means we cannot create the object of an interface

- implements keyword is used by classes to implement an interface.

- The method that implements an interface must be declared public and type signature must be same as declared in the interface.

- If a class implements multiple interfaces, or an interface extends multiple interfaces, it is known as multiple inheritance
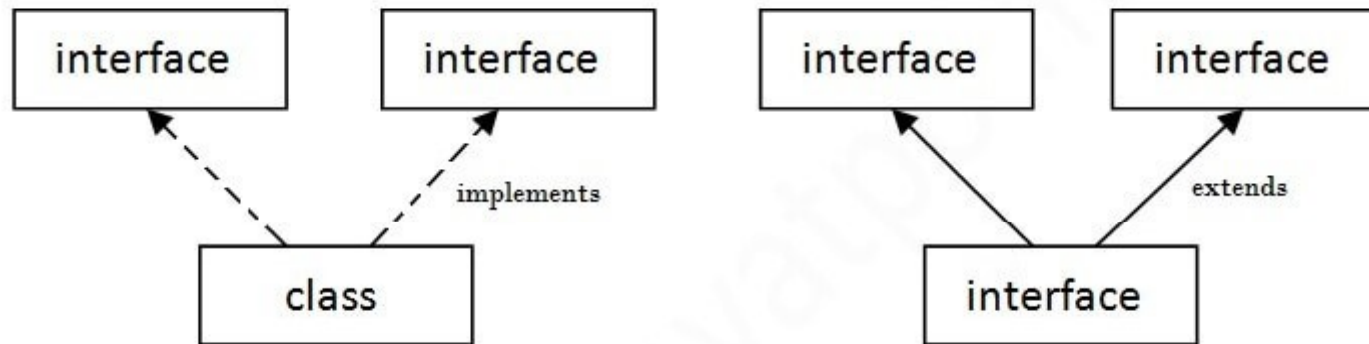
Multiple Inheritance in Java

- Implementing interface
- Interfaces are used as super class whose properties are inherited by classes . General form
- class class_nm extends class_nm implements interface1,interface2

  {

  statements;

  }

class A extends B implements Test,Test1

  {

  statements;

  }

```java
interface solution
{
public void Hello();
public void Welcome();
}
public class Test implements solution
{
    public void Hello()
 {
        System.out.println("Hello world");
    }
    public void Welcome()
 {
        System.out.println("Welcome to BCS");
}
public static void main(String[] args)
 {
    Test T= new Test()
     T.Hello();
     T.Welcome();
     }
}
```

- public interface maths {
- public void add();
- public void sub();
- public void mul();
- public void div();
- }

```java
import java.util.Scanner;
public class student1 implements maths {
        public void add() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values to perform
    addition");
        int a=kb.nextInt();
        int b=kb.nextInt();
        int s=a+b;
        System.out.println("Sum of "+a+" and "+b+" is "+s);
    }

        public void sub() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values to perform
    substraction");
        int a=kb.nextInt();
        int b=kb.nextInt();
        int s=a-b;
        System.out.println("Difference of "+a+" and "+b+" is "+s);
    }
```

```java
        public void mul() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values multiplication");
        int a=kb.nextInt();
        int b=kb.nextInt();
        int s=a*b;
        System.out.println("Product of "+a+" and "+b+" is "+s);
    }

         public void div() {
        Scanner kb = new Scanner(System.in);
        System.out.println("Enter any two integer values division");
        int a=kb.nextInt();
        int b=kb.nextInt();
        int s=a/b;
        System.out.println("Quotient of "+a+" and "+b+" is "+s);
    }
   public static void main(String[] args) {
     student1  s  =  new student1();
     s.add();
     s.sub();
     s.mul();
     s.div();
     }
}
```

- **Implementing Multiple Interfaces**
  In Java, a class can also implement multiple interfaces.
   For example,
  interface A
   {
  // members of A
  }
  interface B
  {
  // members of B
   }
   class C implements A, B
   {
  // abstract members of A
  // abstract members of B
  }

- **Extending an Interface**

  Similar to classes, interfaces can extend other interfaces. The extends keyword is used for extending interfaces.

  For example,

  interface Line

  {

  // members of Line interface

  }

  interface Polygon extends Line

  {

  // members of Polygon interface

  // members of Line interface

  }

- Write a program to calculate area of circle and square. for that create interface area with methods public void Square(),public void Circle().implement these methods. In class shape