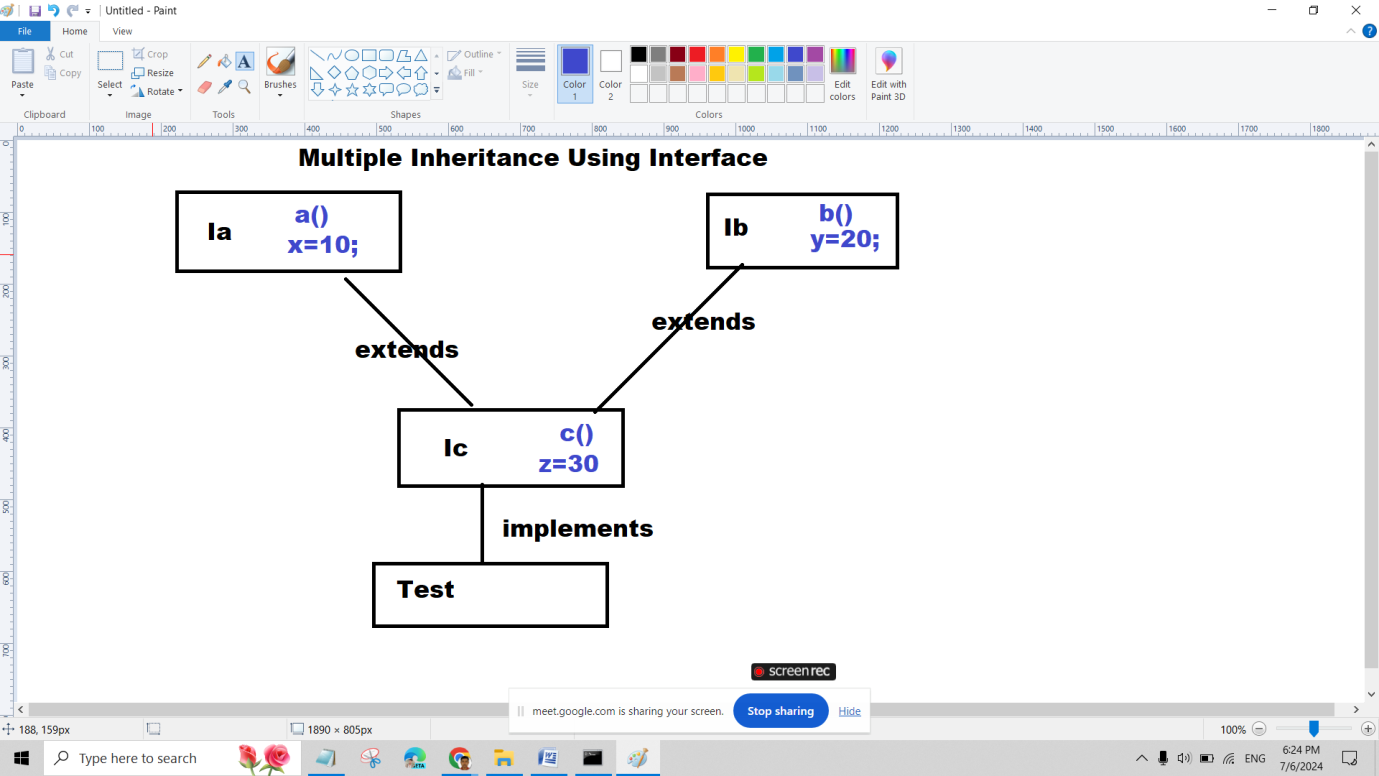
Q1.Explain interface in java programming?

Ans: we can also achieve abstraction in java programming with interface

1. Using interface we achieve 100% abstraction in java because by default every method in interface is public and abstract
2. No need to use abstract keyword in interface
3. Interface is a collection of method declaration and final variable
4. Interface keyword is used to declare interface
5. A class can implements any number of interfaces at a time
6. A interface can extends more than one interface at a time(Multiple inheritance)
7. We cannot create an instance / object of interface, but it can store the reference of its implemented class.

Apart from interface method a implemented class can have its own method

|  |
| --- |
| // Interface Ia with a constant integer x declared as 10 and an abstract method a()  interface Ia {  int x = 10; // Constant integer x  void a(); // Abstract method a()  }  // Interface Ib with a constant integer y declared as 20 and an abstract method b()  interface Ib {  int y = 20; // Constant integer y  void b(); // Abstract method b()  }  // Class Test implementing interfaces Ia and Ib  class Test implements Ia, Ib {  // Implementation of method a() from interface Ia  public void a() {  System.out.println("Ia Interface method : " + x);  }  // Implementation of method b() from interface Ib  public void b() {  System.out.println("Ib Interface method : " + y);  }  // Additional method defined in Test class  public void hello() {  System.out.println("This is Test class Method");  }  // Main method to demonstrate usage of interfaces and Test class  public static void main(String args[]) {  // Creating an object of type Ia, which references an instance of Test (polymorphism)  Ia obj = new Test();  obj.a(); // Calls method a() through interface reference  // Creating an object of type Test  Test t = new Test();  t.hello(); // Calls method hello() from Test class  t.a(); // Calls method a() from Test class  t.b(); // Calls method b() from Test class  }  } |



interface Ia{

int x=10;

void a();

}

interface Ib{

int y=20;

void b();

}

interface IC extends Ia,Ib{

int z=30;

void c();

}

class Test implements IC{

public void a(){

System.out.println("Ia INterface method : "+x);

}

public void b(){

System.out.println("Ib INterface method : "+y);

}

public void c(){

System.out.println("Ic INterface method : "+z);

}

public void hello(){

System.out.println("This is Test class Method");

}

public static void main(String args[]){

//Ia obj=new Ia();//C.E.

Ia obj=new Test();

obj.a();

//obj.hello();

Test t=new Test();

t.hello();

t.a();

t.b();

t.c();

//t.x=22;

}

}