1)Write a Java program to create a new array list, add some elements (string) and print out the collection by using for-each loop

import java.util.ArrayList;

class array1

{

public static void main(String args[])

{

ArrayList<Integer> list1 =new ArrayList< >();

list1.add(1);

list1.add(2);

list1.add(3);

list1.add(4);

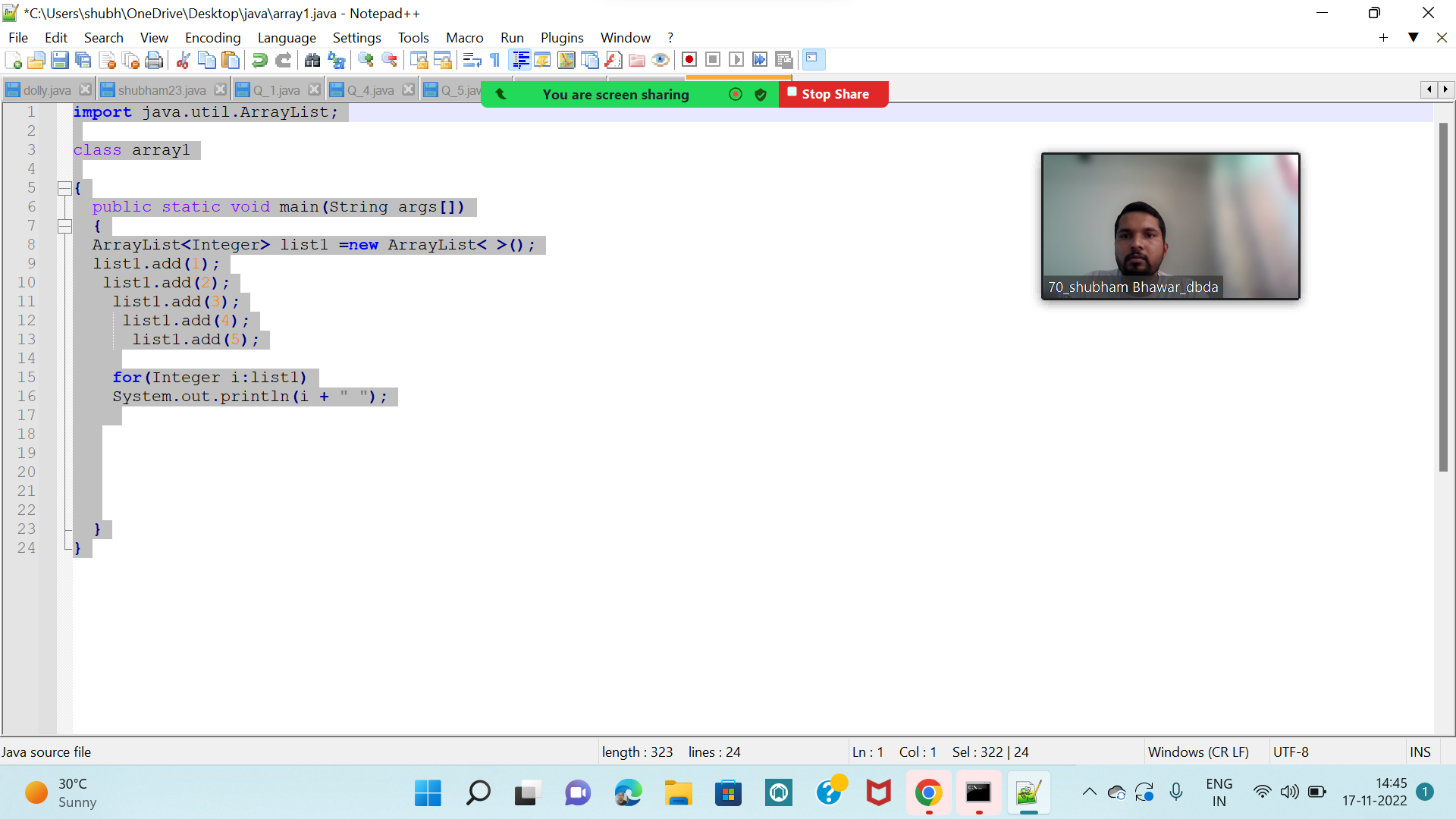
list1.add(5);

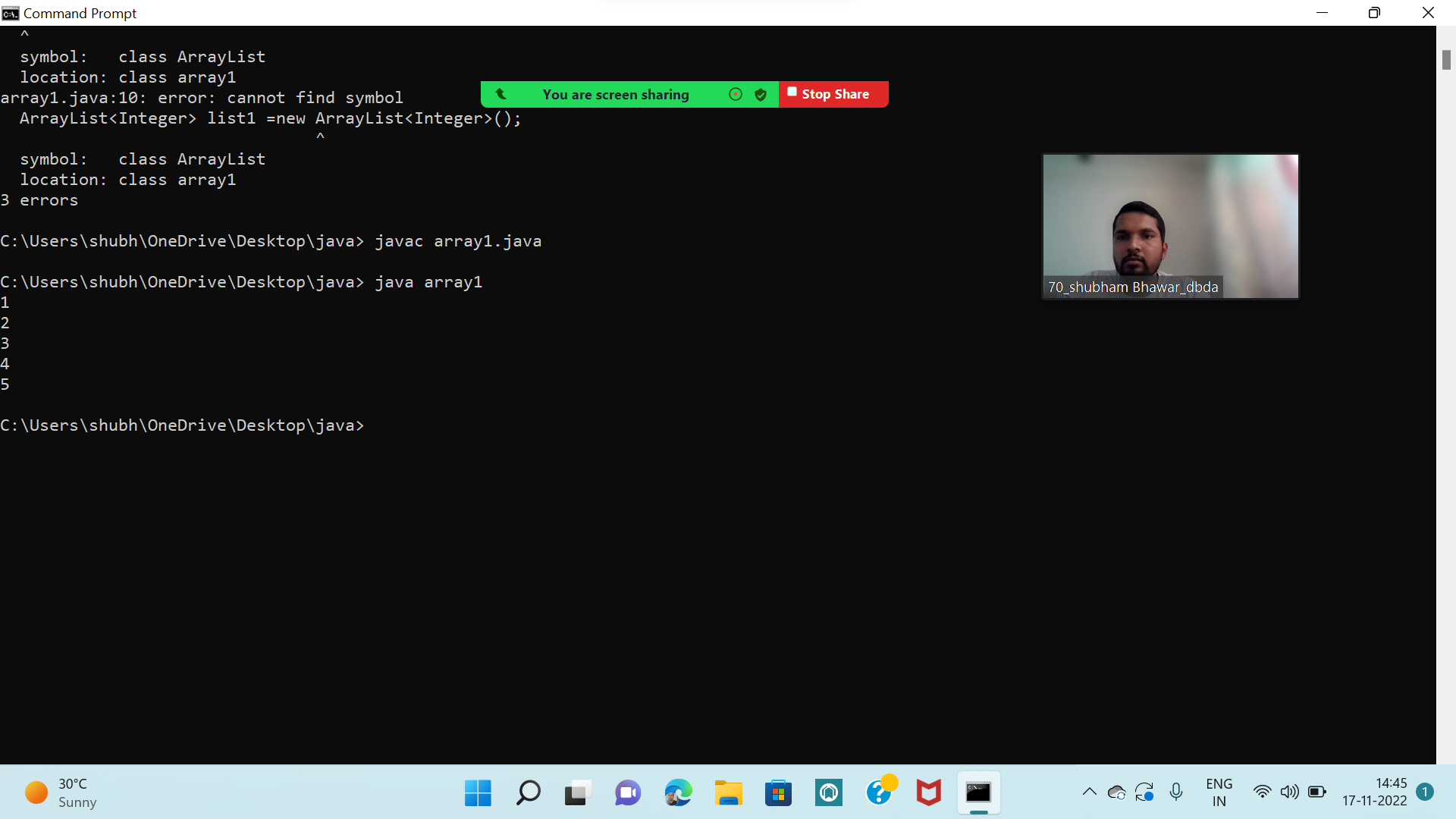
for(Integer i:list1)

System.out.println(i + " ");

}

}



.

Q3 : Write a program to create a class named shape. In this class we have three sub classes circle, triangle and square, each class has two member function named draw () and erase (). Create these using Runtime Polymorphism concepts.

class shape {

void draw(){

System.out.println("drawing shape");

}

void erase(){

System.out.println("erasing shape");

}

}

class circle extends shape{

@Override

void draw(){

System.out.println(" i am drawing circle");

}

@Override

void erase(){

System.out.println(" i am erasing circle");

}

}

class tringle extends shape{

@Override

void draw(){

System.out.println(" i am drawing tringle");

}

@Override

void erase(){

System.out.println(" i am erasing tringle");

}

}

class square extends shape{

@Override

void draw(){

System.out.println(" i am drawing square");

}

@Override

void erase(){

System.out.println(" i ameasing square");

}

}

public class shape12{

public static void main(String args[])

{

shape circle1 = new circle();

shape tringle1= new tringle();

shape square1 = new square();

circle1.draw();

circle1.erase();

tringle1.draw();

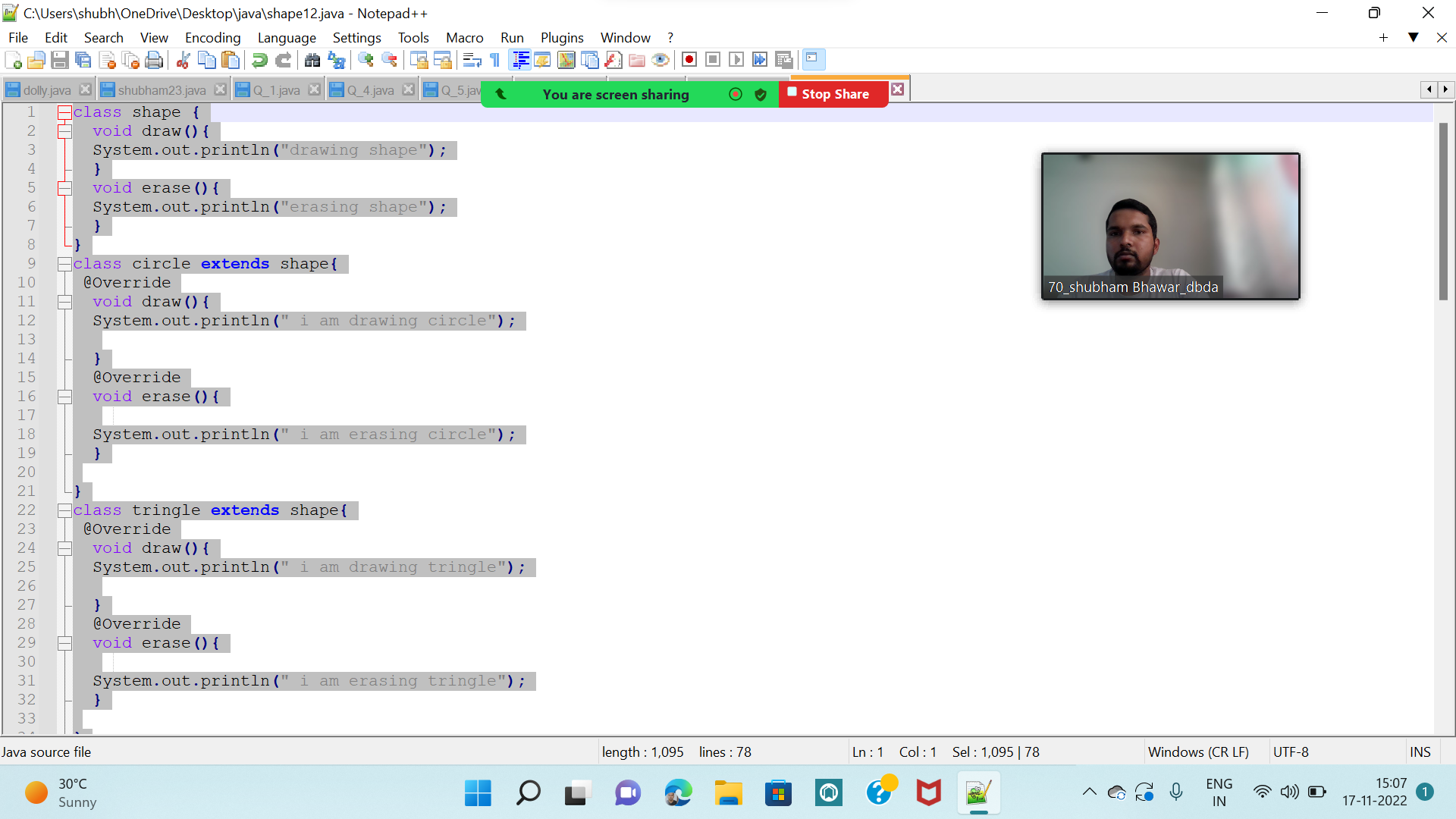
tringle1.erase();

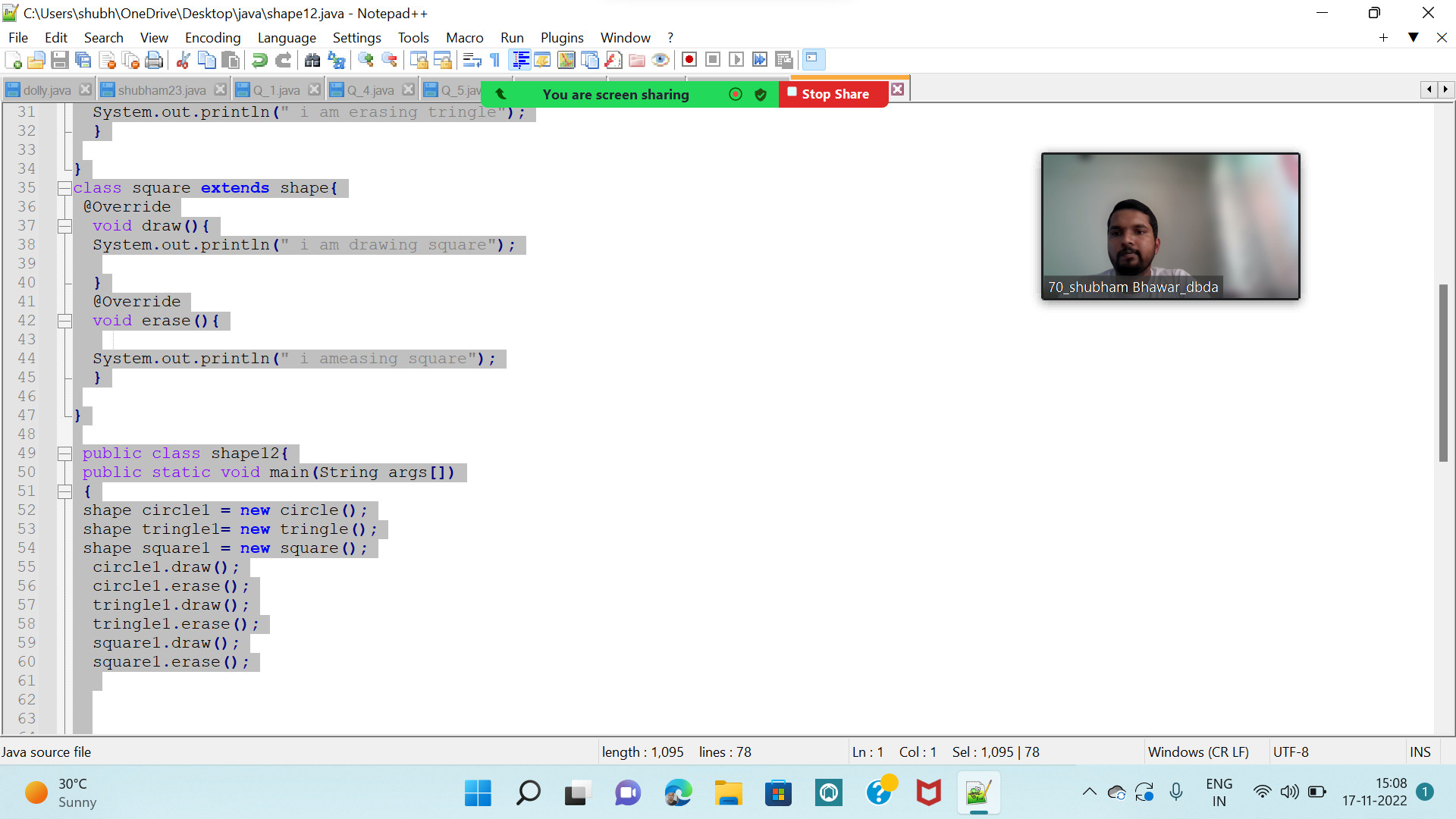
square1.draw();

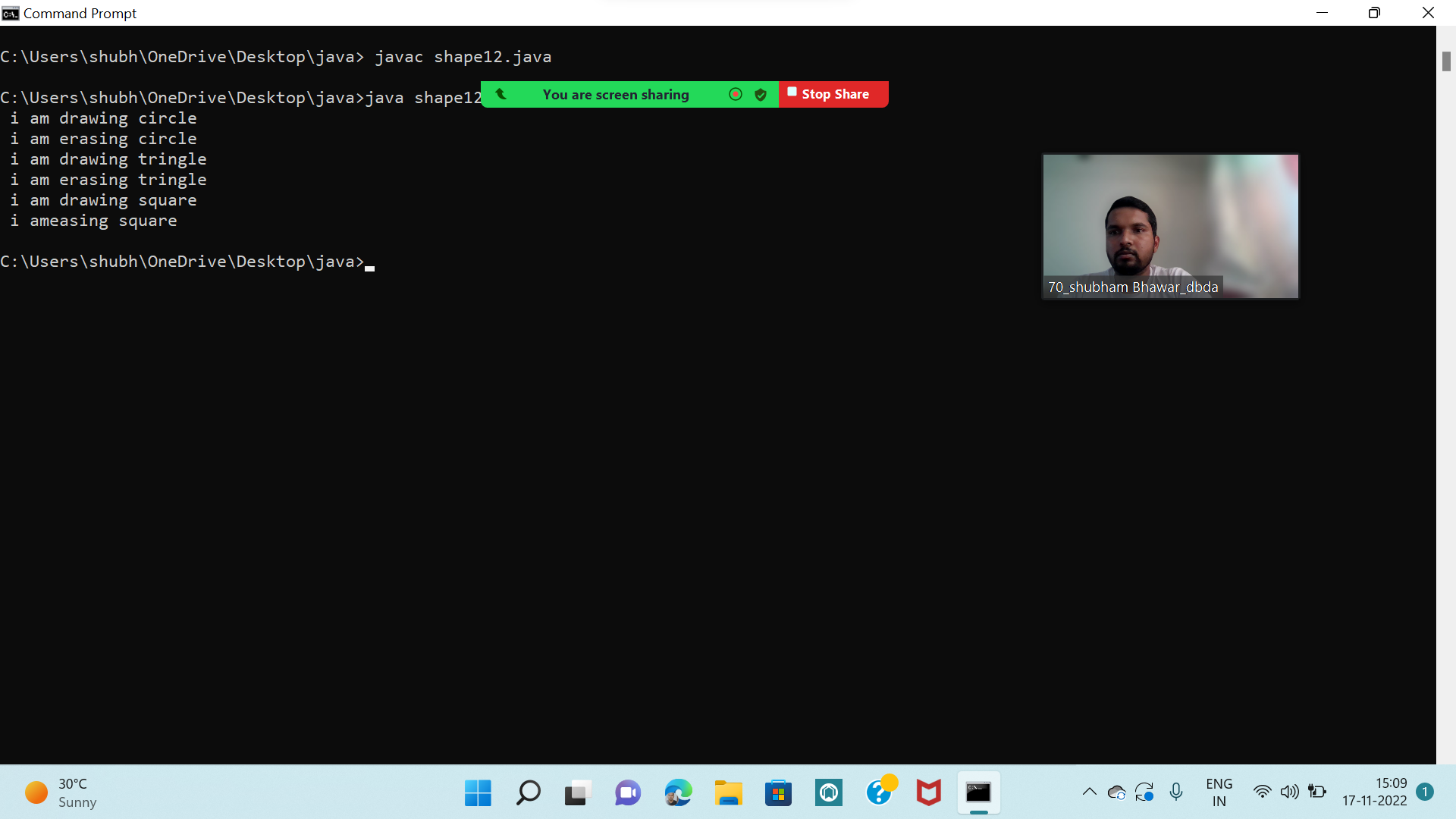
square1.erase();

}

}







Q4 : Constructor chaining (10 Marks)

class grandparent

{

String grandfathername ;

String grandmothername ;

grandparent( String p, String q )

{

grandfathername=p;

grandmothername=q;

System.out.println("grandfather name is :"+grandfather);

System.out.println("grandfather name is :"+grandmothername);

}

}

class parent extends grandparent

{

String fathername ;

String mothername ;

parent (String r, String s,String p,String q)

{

super(p,q);

fathername=r;

mothername=s;

System.out.println("mothername is :"+mothername);

System.out.println("fathername is:"+fathername);

}

}

class child extends grandparent{

{

child(String p,String q,String r,String s){

super(p,q,r,s);

}

public static void main(String[] args)

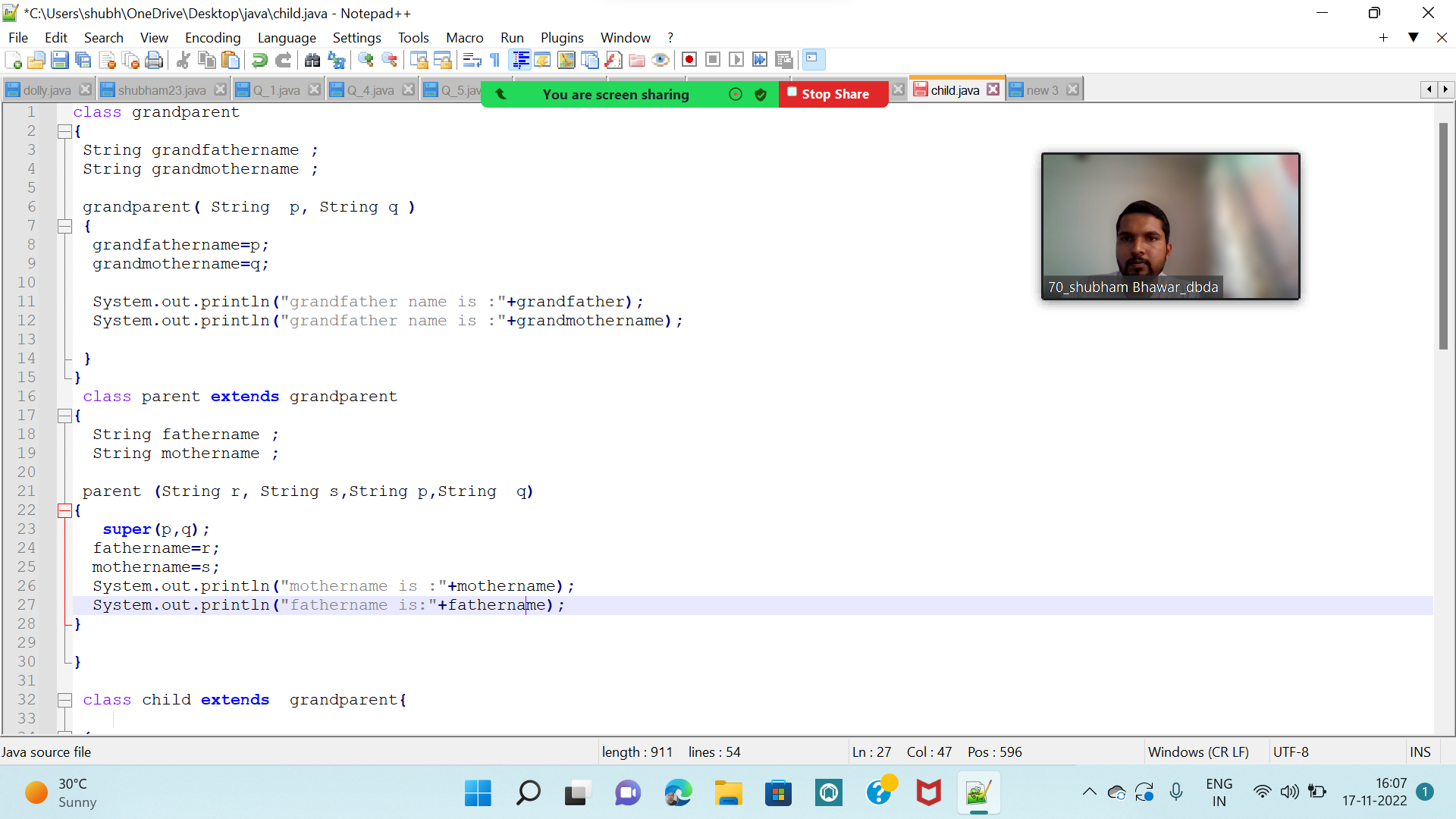
{

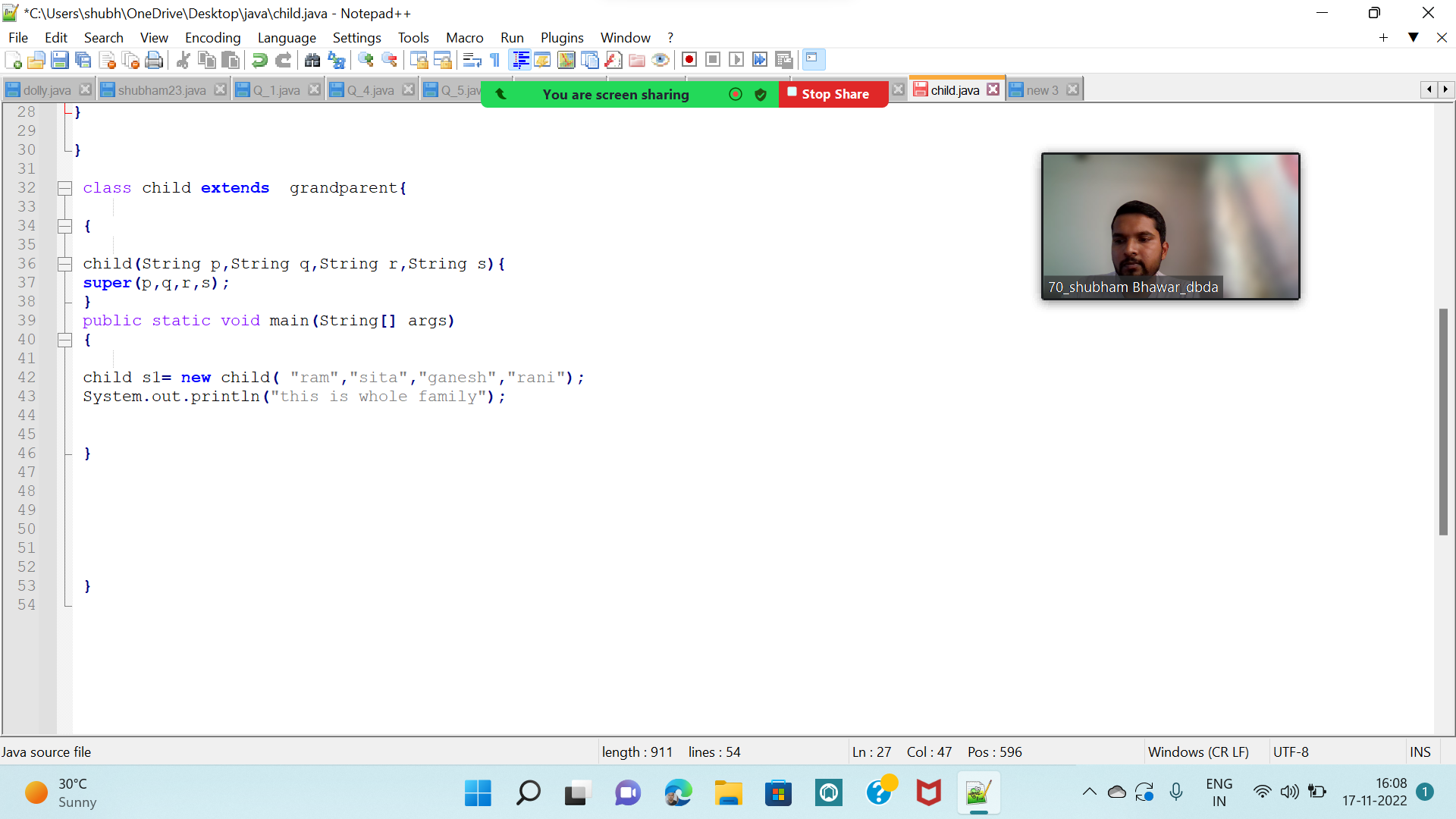
child s1= new child( "ram","sita","ganesh","rani");

System.out.println("this is whole family");

}

}





Q2 : Develop a class BankAccount having following data members : (10 Marks) int accno double balance Write appropriate constructors to initialize data members Define the following functions : withdraw : balance will reduce deposit : balance will increase show : display accno and balance If user tries to withdraw more than the balance, use exception handling code. Demonstrate the concept of exception handling in main() function.

class balance extends Exception

{

balance(String s){

super(s);

}

}

public class bankaccount{

int accno;

double balance;

bankaccount(int a,double b){

accno=a;

balance=b;

}

void withdraw(int c);{

if(c>balance){

try{

throw new balance("insufficient amount");

}

catch(balance e){

System.out.println(e.getmeassage());

}

}

else{

balance=balance -c ;

System.out.println("current balance after withdrwal"+c-a);

}

void deposit(int d);{

balance= balance + d;

System.out.println();

}

void show();

{

System.out.println("account number " +accno);

System.out.println("avialable balance"+balance);

}

public static void main(String args[]){

bankaccount bank=new bankaccount(101,20000);

bank.withdraw(100);

bank.deposit(1000);

bank.show();

}

}

