Q1 : 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.\*;

public class Array

{

public static void main(String[] args) {

ArrayList<String> arr=new ArrayList<String>();

arr.add("shubham");

arr.add("bhopal");

arr.add("suman");

arr.add("avneesh");

arr.add("cdac");

arr.add("mumbai");

System.out.println("Elements of array list are:");

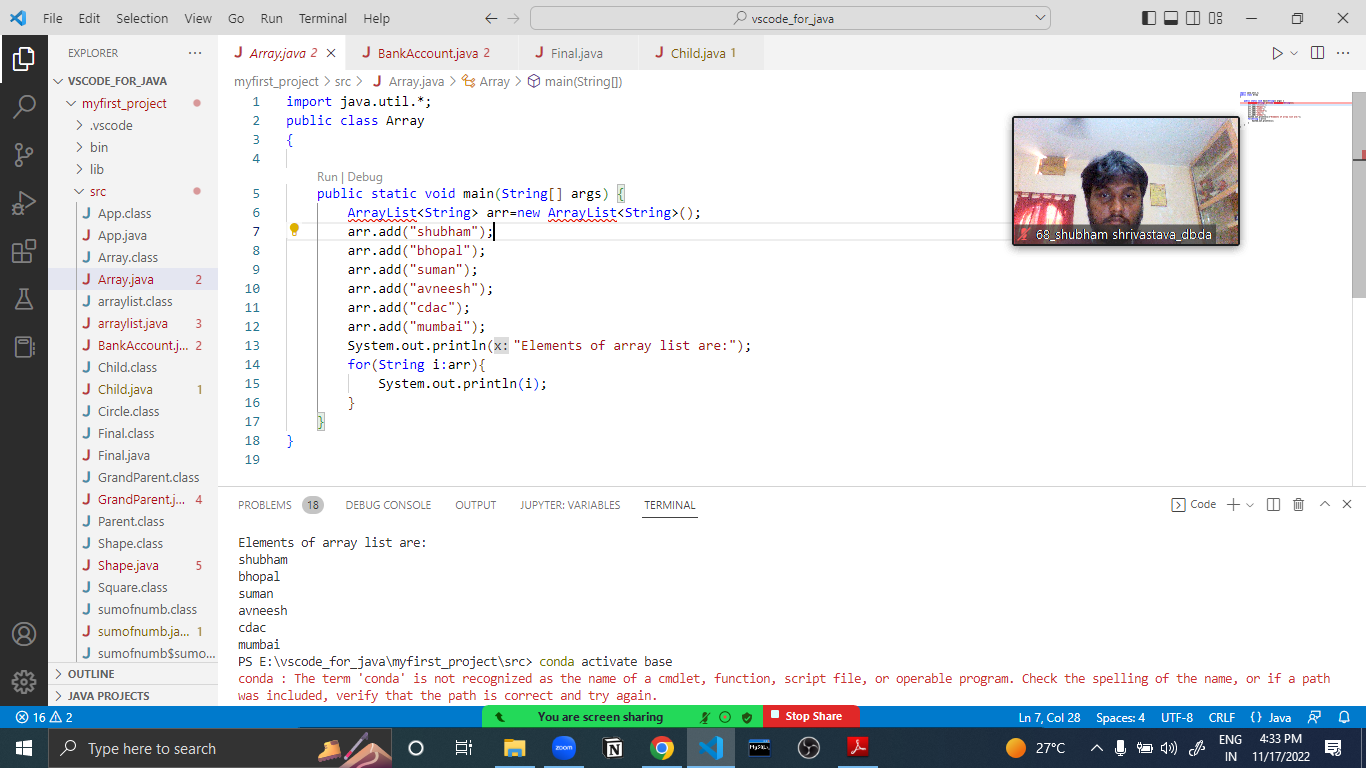
for(String i:arr){

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("Shape is drawing");

}

void erase(){

System.out.println("Shape is erasing");

}

}

class Circle extends Shape{

void draw(){

System.out.println("Circle is drawing");

}

void erase(){

System.out.println("Circle is erasing");

}

}

class Triangle extends Shape{

void draw(){

System.out.println("Triangle is drawing");

}

void erase(){

System.out.println("Triangle is erasing");

}

}

class Square extends Shape{

void draw(){

System.out.println("Square is drawing");

}

void erase(){

System.out.println("Square is erasing");

}

}

public class Final{

public static void main(String[] args){

Shape a = new Circle();

Shape b = new Triangle();

Shape c = new Square();

a.draw();

a.erase();

b.draw();

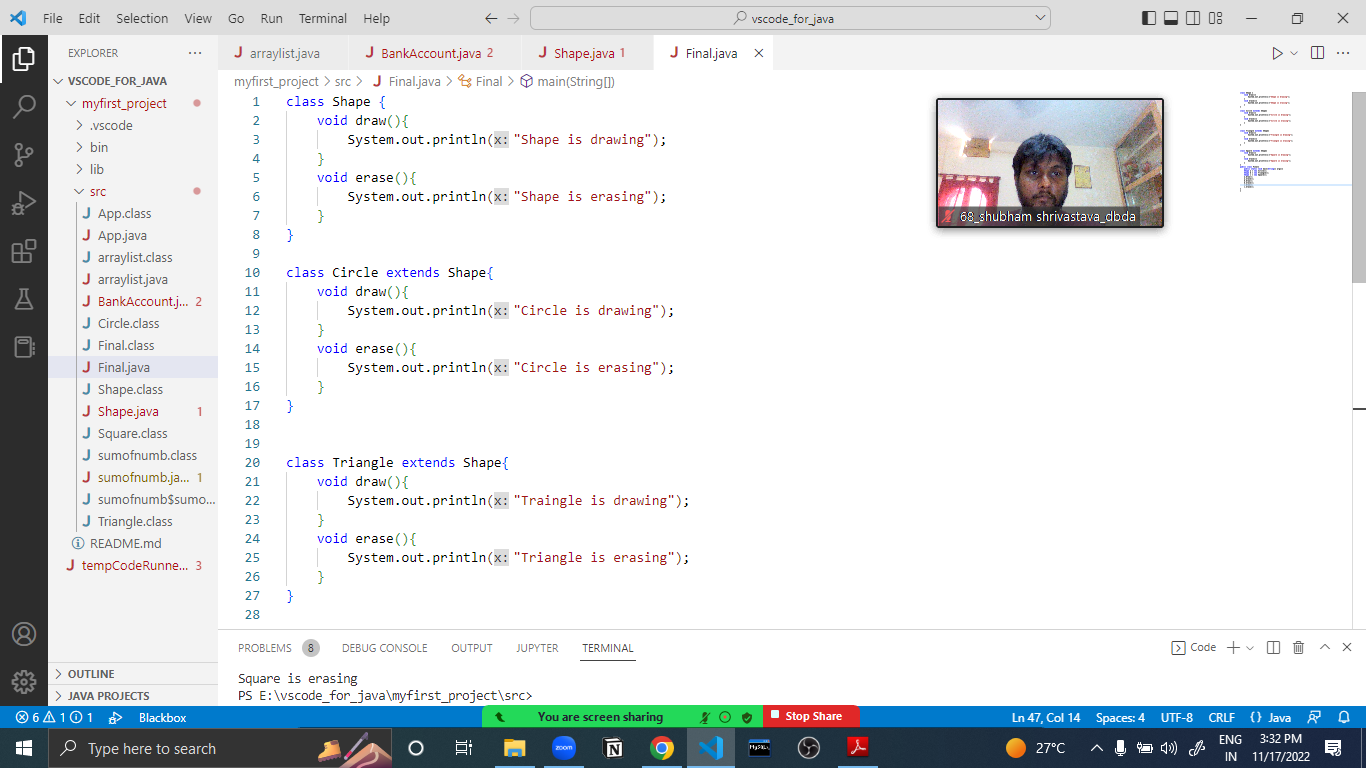
b.erase();

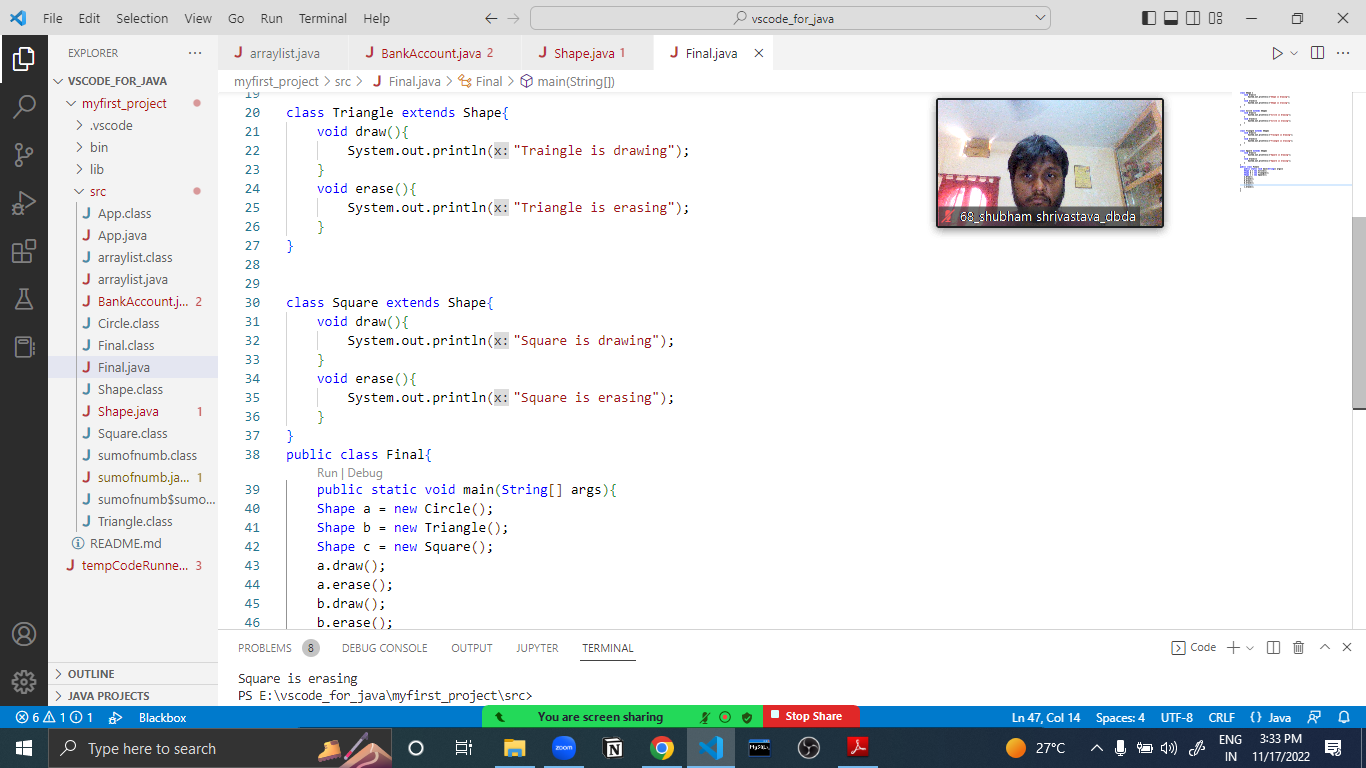
c.draw();

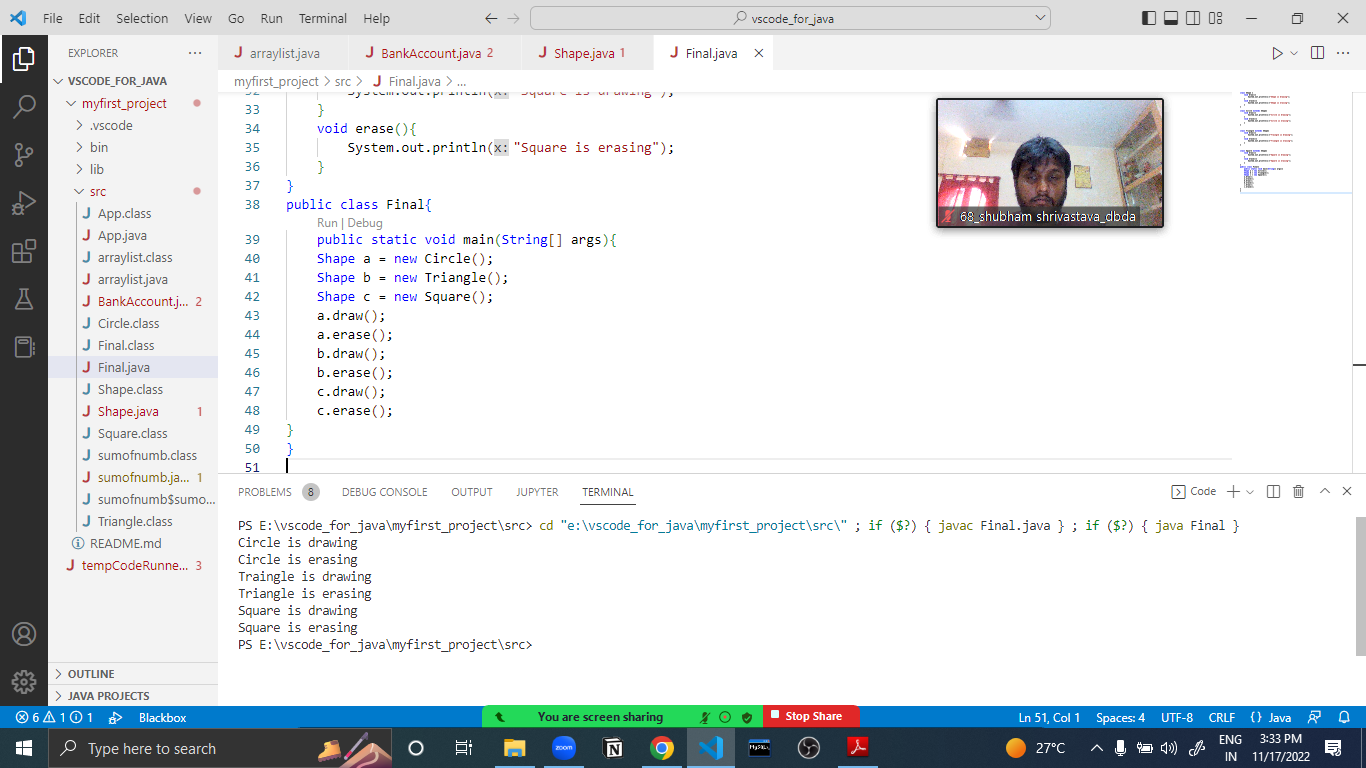
c.erase();

}

}







Q4 : Constructor chaining………….

class GrandParent{

String grandFatherName , grandMotherName;

GrandParent(String a,String b){

grandFatherName = a;

grandMotherName = b;

System.out.println("Grand father name is :"+grandFatherName);

System.out.println("Grand father name is :"+grandMotherName);

}

}

class Parent extends GrandParent{

String FatherName , MotherName;

Parent(String c, String d, String a, String b){

super(a,b);

FatherName = c;

MotherName = d;

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

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

}

}

class Child extends Parent{

Child(String c, String d, String a, String b){

super(c,d,a,b);

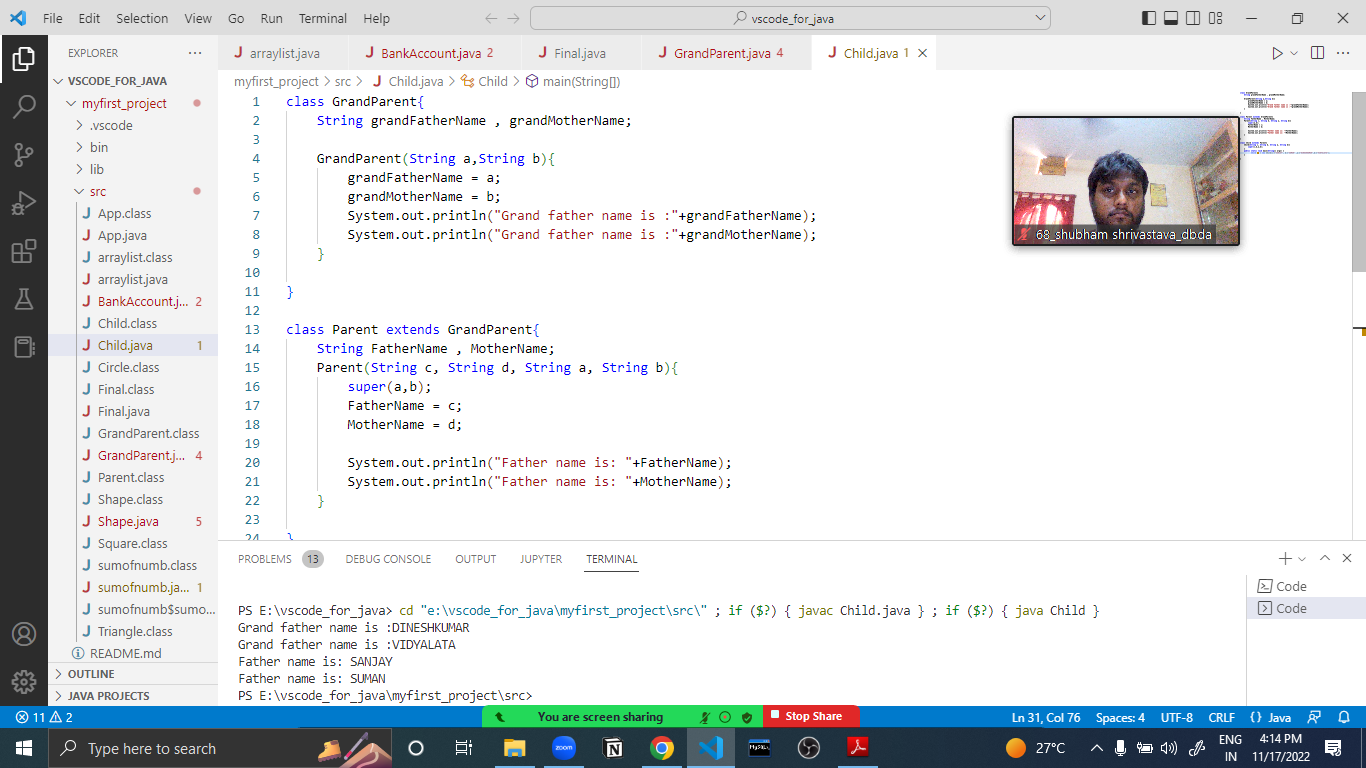
}

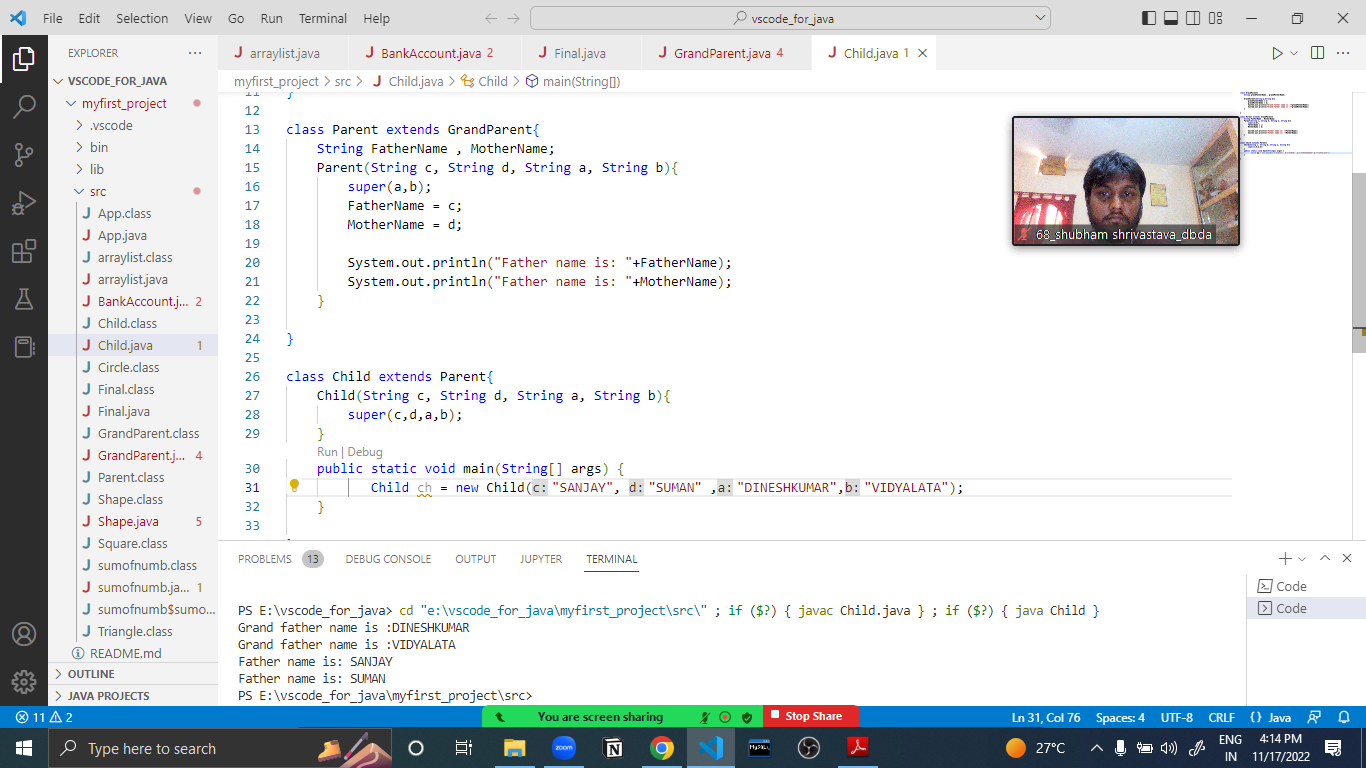
public static void main(String[] args) {

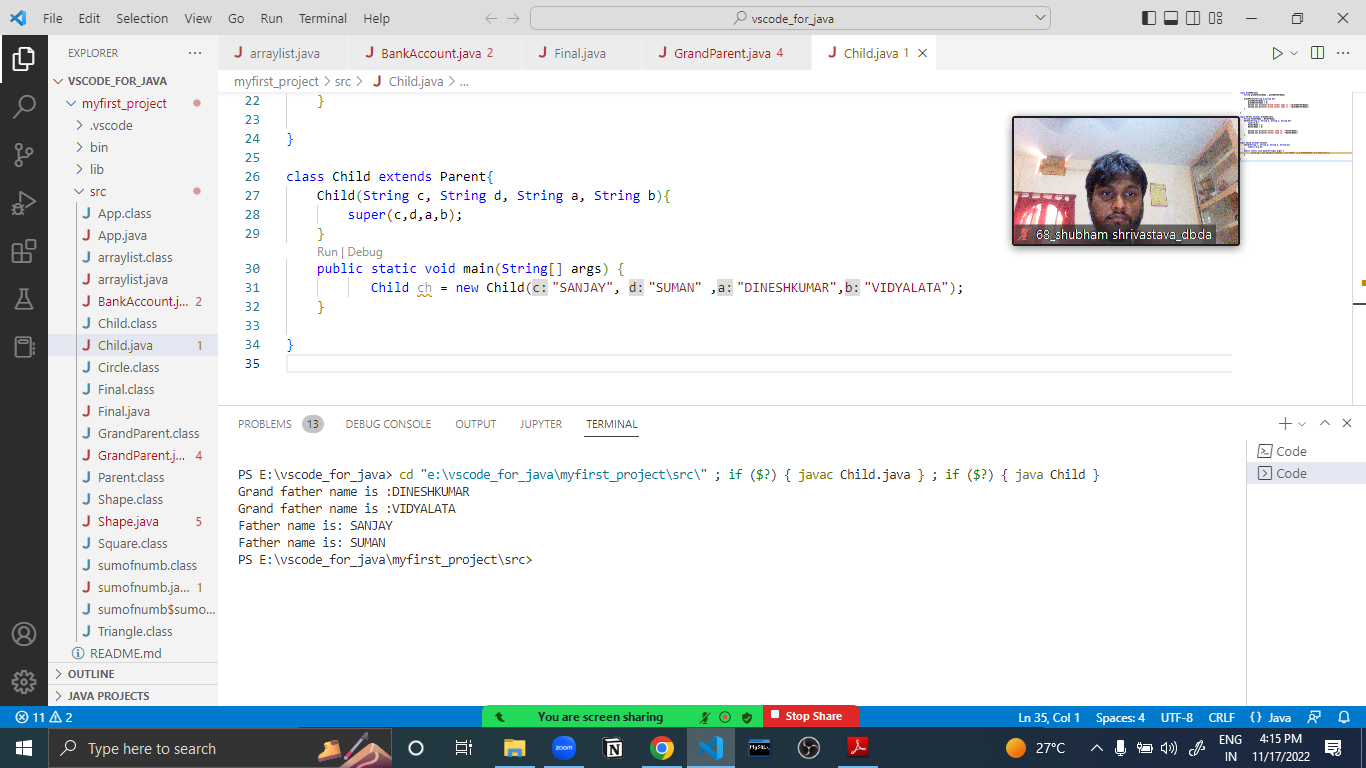
Child ch = new Child("SANJAY","SUMAN,"DINESHKUMAR","VIDYALATA");

}

}







Q2:

-Develop a class BankAccount having following data members

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.getMessage());

}

}

else {

balance = balance - c;

System.out.println("Current balance after withdraw" + c + " is : "+ balance);

}

}

void deposit(int d){

balance = balance + d;

System.out.println("Current balance after withdraw" + d + " is : "+ balance);

}

void show(){

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

System.out.println("available balance is:"+balance);

}

public static void main(String[] args){

BankAccount bank = new BankAccount("100","10000");

bank.withdraw("1000");

bank.deposit("1000");

bank.show();

}

