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. (10 Marks)

import java.util.\*;

class array{

public static void main (String[] args){

List<String> list=new ArrayList();

list.add("alpha");

list.add("beta");

list.add("gamma");

list.add("sigma");

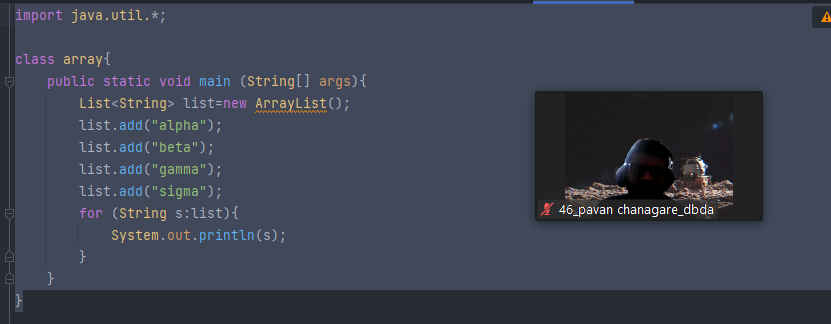
for (String s:list){

System.out.println(s);

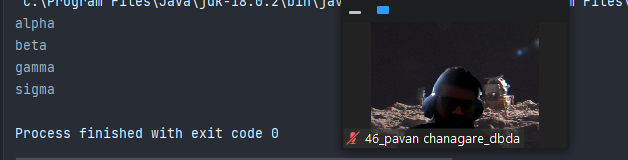
}

}

}



output:



***Develop a class BankAccount having following data members : (10 Marks)***

***int accnodouble balanceWrite appropriate constructors to initialize data membersDefine the following functions :withdraw : balance will reducedeposit : balance will increaseshow : display accno and balanceIf user tries to withdraw more than the balance, use exception handling code.Demonstrate the***

***concept of exception handling in main() function.***

***import java.util.Scanner;***

***public class bank {***

***public static void main(String[] args) {***

***Scanner s= new Scanner(System.in);***

***BankAccount p;***

***System.out.println("Enter account number");***

***int a= s.nextInt();***

***System.out.println("Enter balance");***

***int b =s.nextInt();***

***p=new BankAccount(a,b);***

***lp : while (true){***

***System.out.println("\n1.Withdraw money \n2.Deposit money \n3.Show account details\n0.Exit");***

***int c=s.nextInt();***

***switch (c){***

***case 0:***

***break lp;***

***case 1:***

***System.out.println("Enter the amount of money you want to Withdraw");***

***int w= s.nextInt();***

***p.withdraw(w);***

***break;***

***case 2:***

***System.out.println("Enter the amount of money you want to Deposit");***

***int d= s.nextInt();***

***p.deposit(d);***

***break;***

***case 3:***

***System.out.println("\*\*\*Showing Account Details\*\*\*");***

***p.show();***

***break;***

***}***

***}***

***}***

***}***

***class BankAccount{***

***int accno;***

***double balance;***

***int amt;***

***BankAccount(int a, double b){***

***this.accno=a;***

***this.balance=b;***

***}***

***void withdraw(int o) {***

***this.amt = o;***

***try {***

***if (balance < amt) {***

***throw new ArithmeticException("Your balance is low!");***

***}***

***} catch (ArithmeticException e) {***

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

***}***

***if (balance > amt) {***

***balance = balance - amt;***

***}***

***System.out.println("The updated balance is "+balance);***

***}***

***void deposit(int p){***

***this.amt=p;***

***balance=balance+amt;***

***System.out.println("The updated balance is "+balance);***

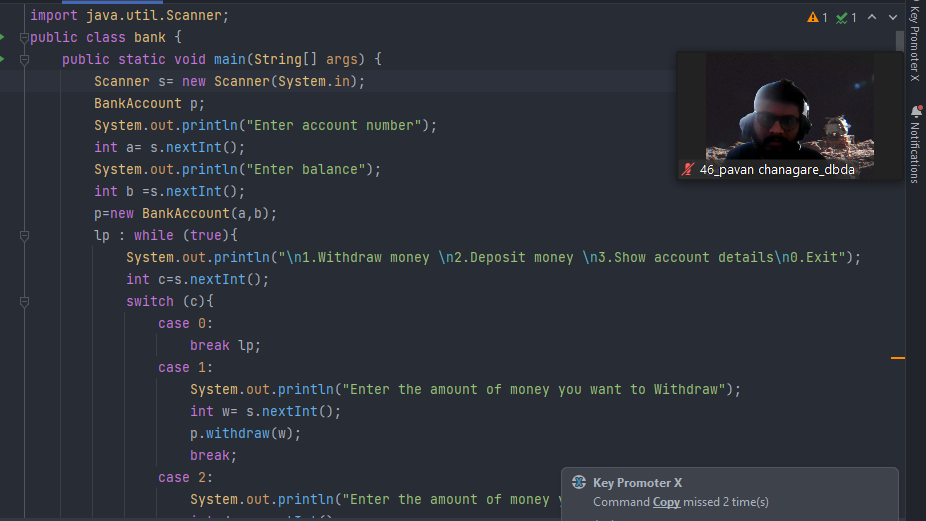
***}***

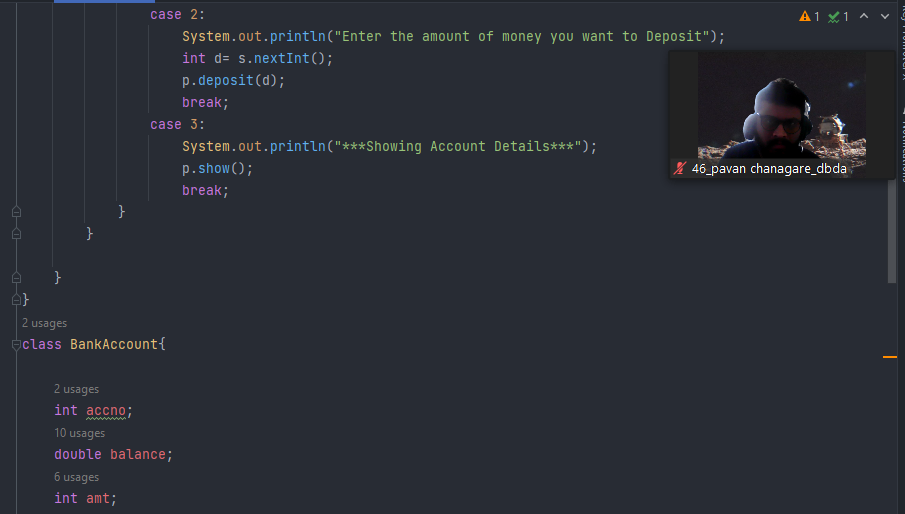
***void show(){***

***System.out.println("The account number is "+accno+" The balance is "+balance);***

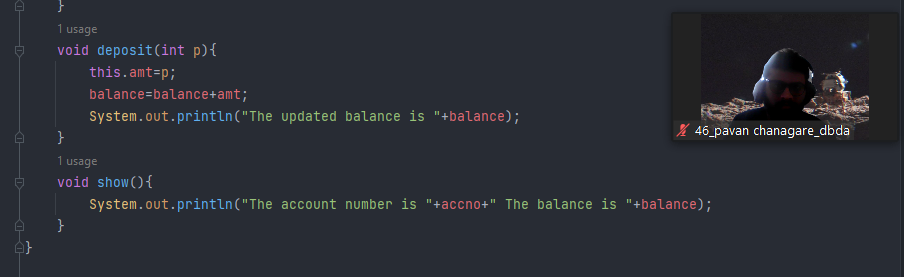
***}***

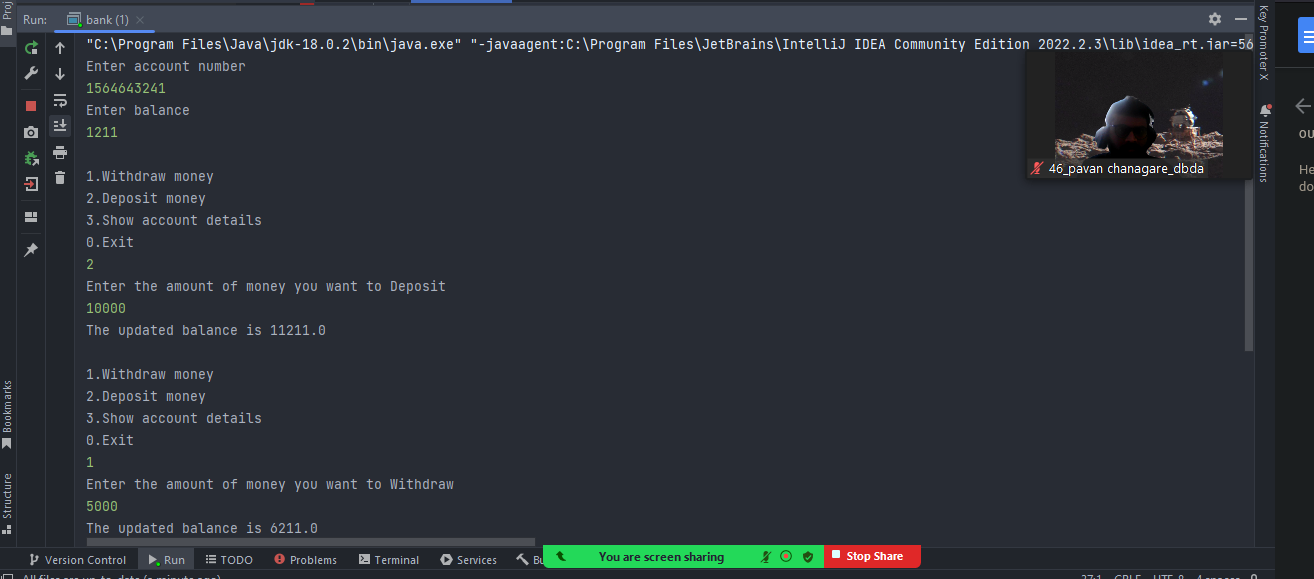
***}***

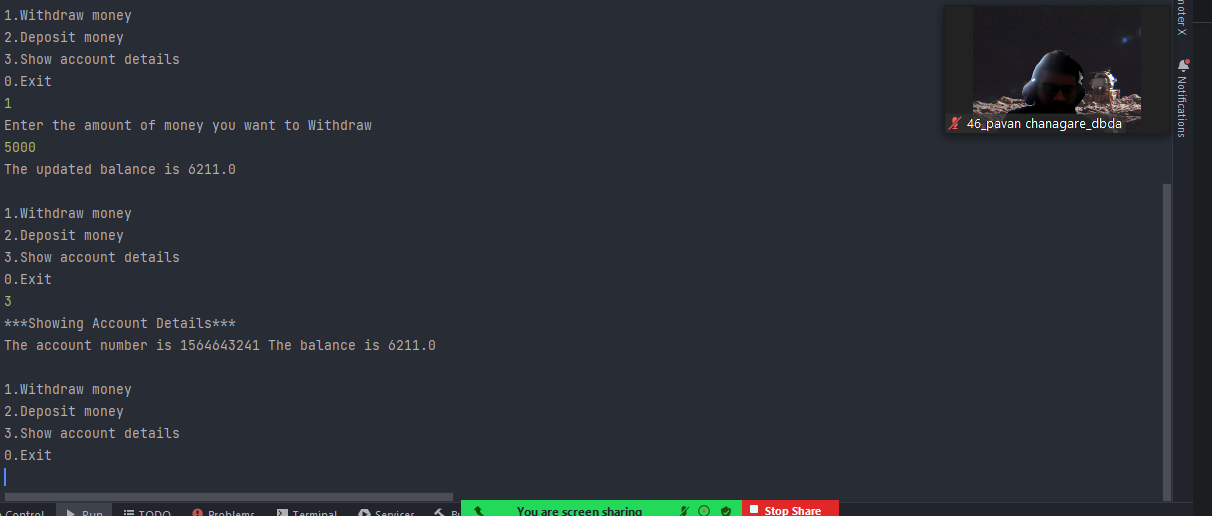
******

******

******

******

******

******

***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. (10 Marks)***

***import java.awt.\*;***

***public class shapes {***

***public static void main(String[] args) {***

***shape c = new circle();***

***c.draw();***

***c.erase();***

***shape t = new triangle();***

***t.draw();***

***t.erase();***

***shape s = new square();***

***s.draw();***

***s.erase();***

***}***

***}***

***abstract class shape{***

***abstract void draw();***

***abstract void erase();***

***}***

***class circle extends shape{***

***@Override***

***void draw() {***

***System.out.println("Drawing a Cricle");***

***}***

***@Override***

***void erase() {***

***System.out.println("Erasing a Circle");***

***}***

***}***

***class triangle extends shape{***

***@Override***

***void draw() {***

***System.out.println("Drawing a Triangle");***

***}***

***@Override***

***void erase() {***

***System.out.println("Erasing a Triangle");***

***}***

***}***

***class square extends shape{***

***@Override***

***void draw() {***

***System.out.println("Drawing a Square");***

***}***

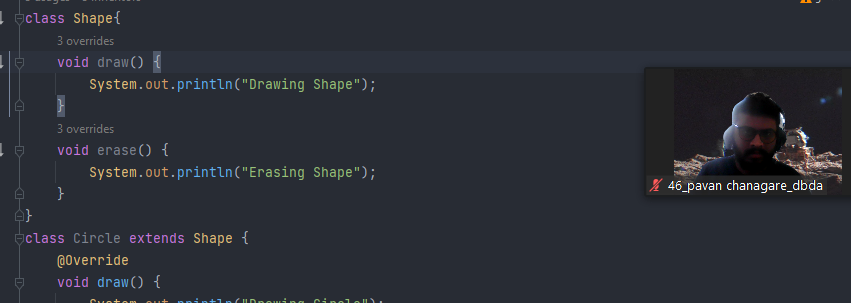
***@Override***

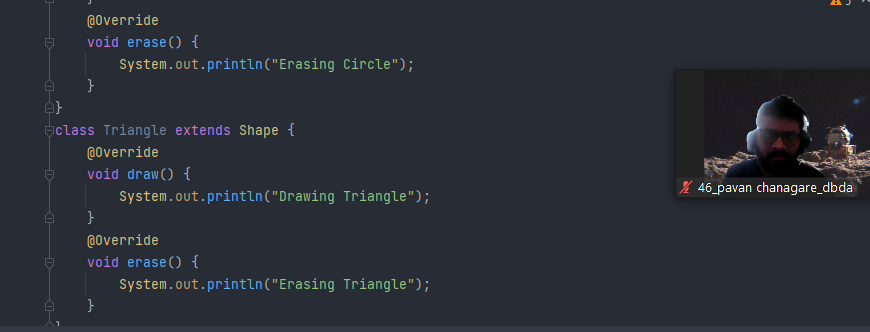
***void erase() {***

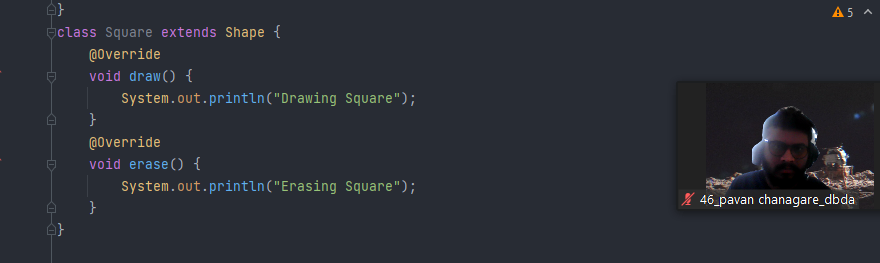
***System.out.println("Erasing a Square");***

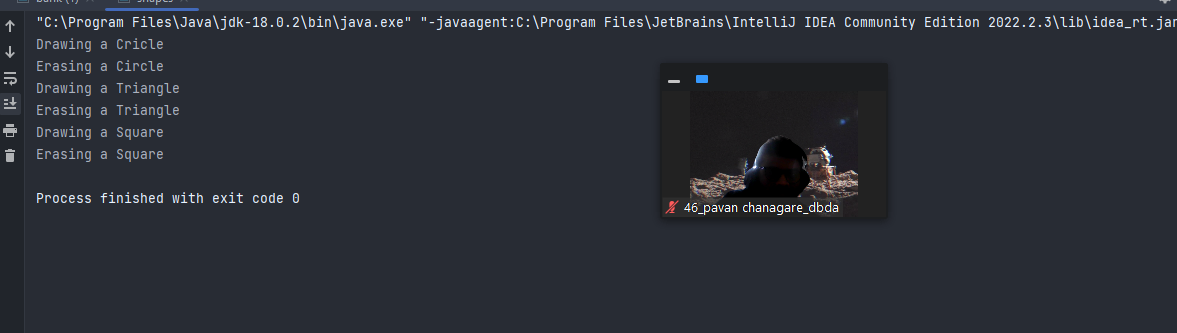
***}***

***}***









***Q4 : Constructor chaining (10 Marks)***

***public class four {***

***public static void main(String[] args) {***

***child c = new child("Cake","lava","Choco","ChocoLava");***

***}***

***}***

***class GrandParent{***

***String grandFatherName;***

***String grandMotherName;***

***public GrandParent(String grandFatherName, String grandMotherName) {***

***this.grandFatherName = grandFatherName;***

***this.grandMotherName = grandMotherName;***

***System.out.println("Grand Father name is "+grandFatherName+" and Grand Mother name is "+grandMotherName);***

***}***

***}***

***class parent extends GrandParent{***

***String fatherName;***

***String motherName;***

***public parent(String grandFatherName, String grandMotherName, String fatherName, String motherName) {***

***super(grandFatherName, grandMotherName);***

***this.fatherName = fatherName;***

***this.motherName = motherName;***

***System.out.println("Father name is "+fatherName+" and Mother name is "+motherName);***

***}***

***public parent(String grandFatherName, String grandMotherName) {***

***super(grandFatherName, grandMotherName);***

***}***

***}***

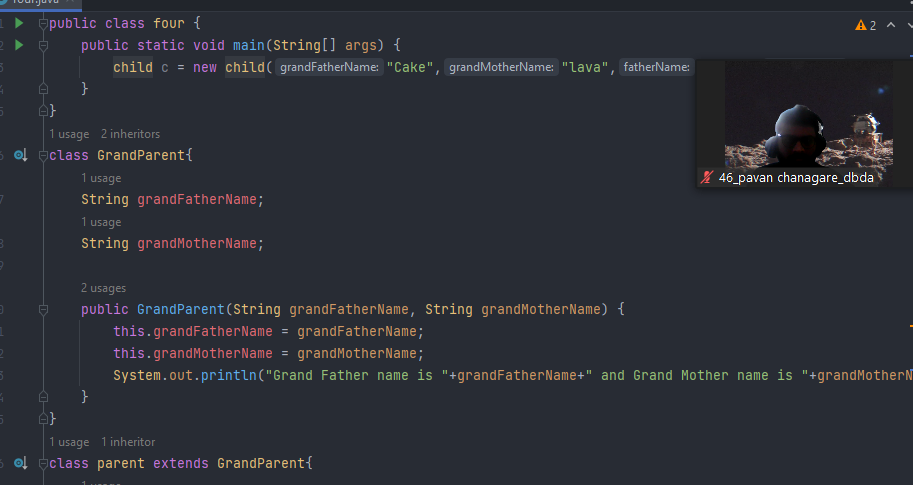
***class child extends parent{***

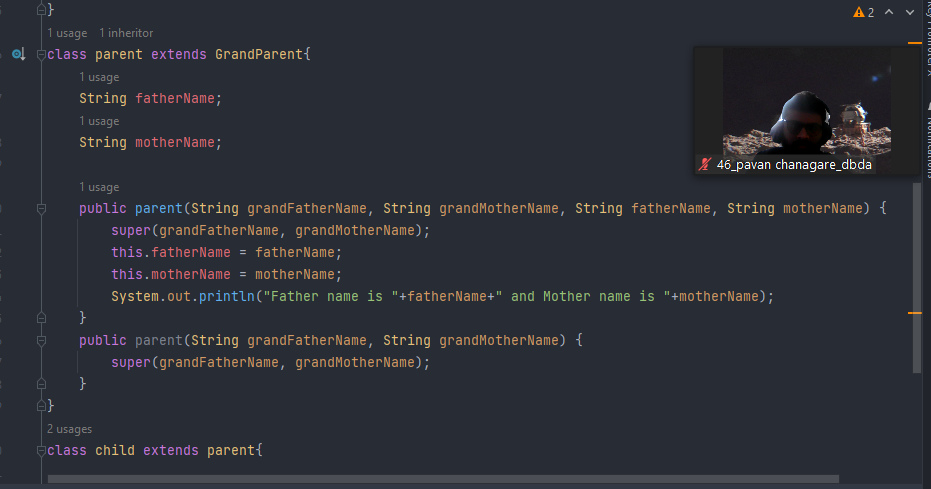
***public child(String grandFatherName, String grandMotherName, String fatherName, String motherName) {***

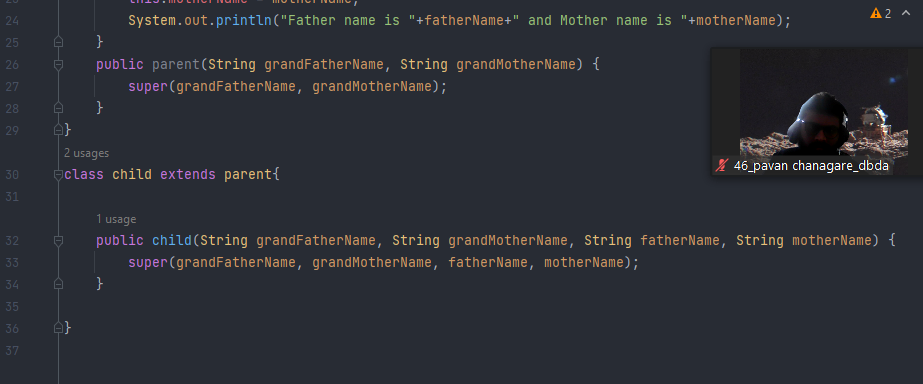
***super(grandFatherName, grandMotherName, fatherName, motherName);***

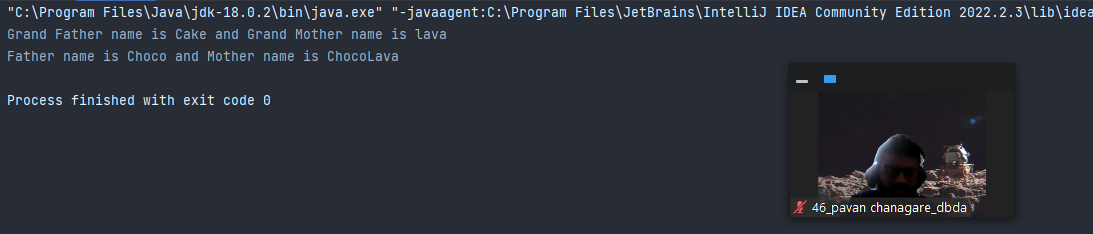
***}***

***}***

******

******

******

******