**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 Q\_1 {

public static void main(String args[])

{

ArrayList<String> al = new ArrayList<String>();*//created new array\_list*

al.add("Ram"); *//use add method to add elements in collection*

al.add("Sham");

al.add("Rina");

al.add("Rohan");

al.add("Sita");

System.*out*.println("The Created ArrayList : "+al);

System.*out*.println("Printing ArrayList using foreach loop : ");

for(String a : al) *//use of foreach loop*

{

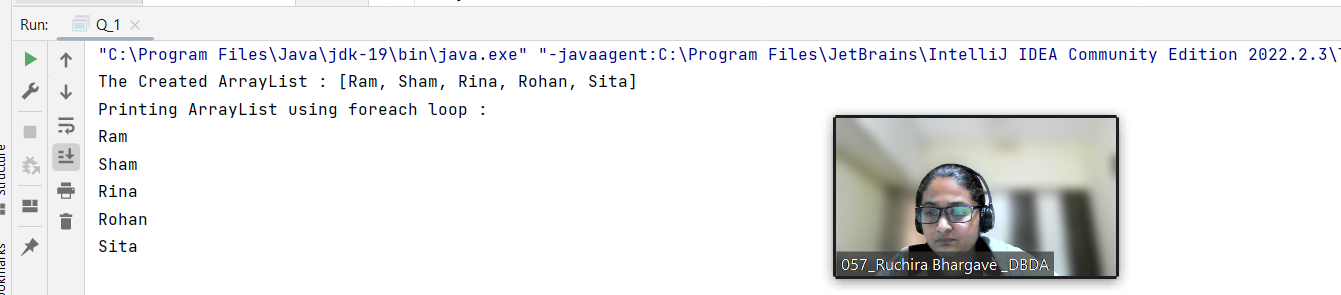
System.*out*.println(a);

}

}

}

Output:



**Q2 : Develop a class BankAccount having following data members :**

**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 acc no and balance**

**If the user tries to withdraw more than the balance, use exception handling code. Demonstrate the concept of exception handling in the main() function.**

import java.util.Scanner;

class BankAccount

{

int accno;

double balance;

public BankAccount(int accno,double balance)

{

this.accno=accno;

this.balance=balance;

}

public void withdraw(double amount)

{

if(balance < amount)

{

try

{

System.*out*.println("Invalid amount entered");

}

catch(Exception e)

{

System.*out*.println("Error : withdrawal amount should be less than the balance");

}

}

else

{

balance = balance - amount;

System.*out*.println("balance of account after withdraw is : " + balance);

}

}

public void deposite(double amount)

{

balance = balance + amount;

System.*out*.println("balance of account after depositing amount is : "+balance);

}

public void show\_account\_details()

{

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

System.*out*.println("balance of account : "+balance);

}

}

public class Q\_2

{

public static void main(String args[])

{

BankAccount ba = new BankAccount(123,10000);

ba.show\_account\_details();

ba.withdraw(12000);

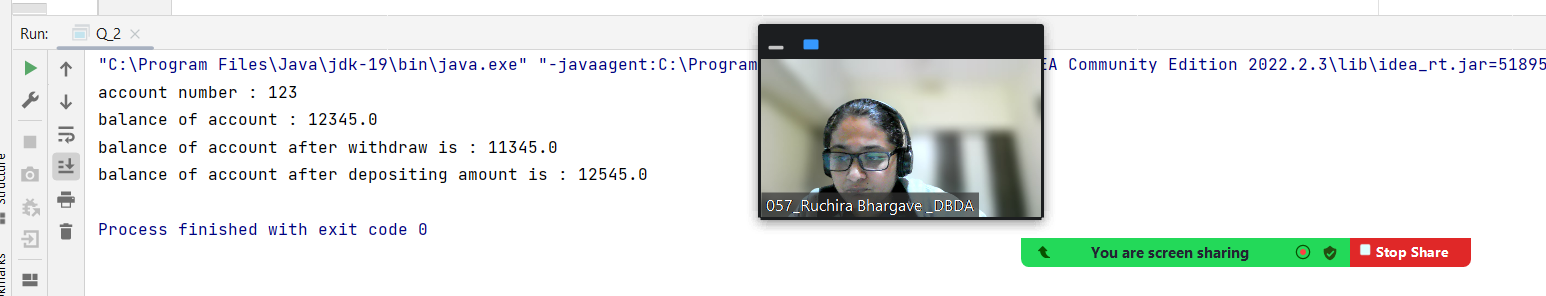
ba.deposite(12000);

}

}

**OutPut:**

1. **Without error occurred:**

****

1. **With the exception handling:**



**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("this is Draw of Shape");

}

void erase()

{

System.*out*.println("This is Erase of Shape");

}

}

class Circle extends Shape

{

void draw()

{

System.*out*.println("this is Draw of Circle");

}

void erase()

{

System.*out*.println("This is Erase of Circle");

}

}

class Triangle extends Shape

{

void draw()

{

System.*out*.println("this is Draw of Triangle");

}

void erase()

{

System.*out*.println("This is Erase of Triangle");

}

}

class Square extends Shape

{

void draw()

{

System.*out*.println("this is Draw of Square");

}

void erase()

{

System.*out*.println("This is Erase of Square");

}

}

public class Q\_3 {

public static void main(String args[])

{

Shape shape = new Shape();

Circle c = new Circle();

Triangle t = new Triangle();

Square s = new Square();

Shape c1 = new Circle();

shape.draw();

shape.erase();

c.draw();

c.erase();

t.draw();

t.erase();

s.draw();

s.erase();

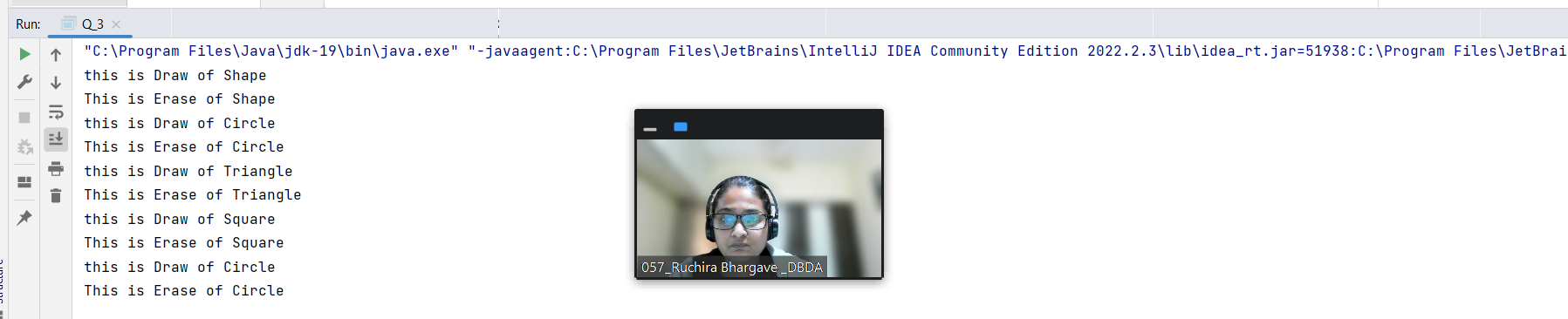
c1.draw();

c1.erase();

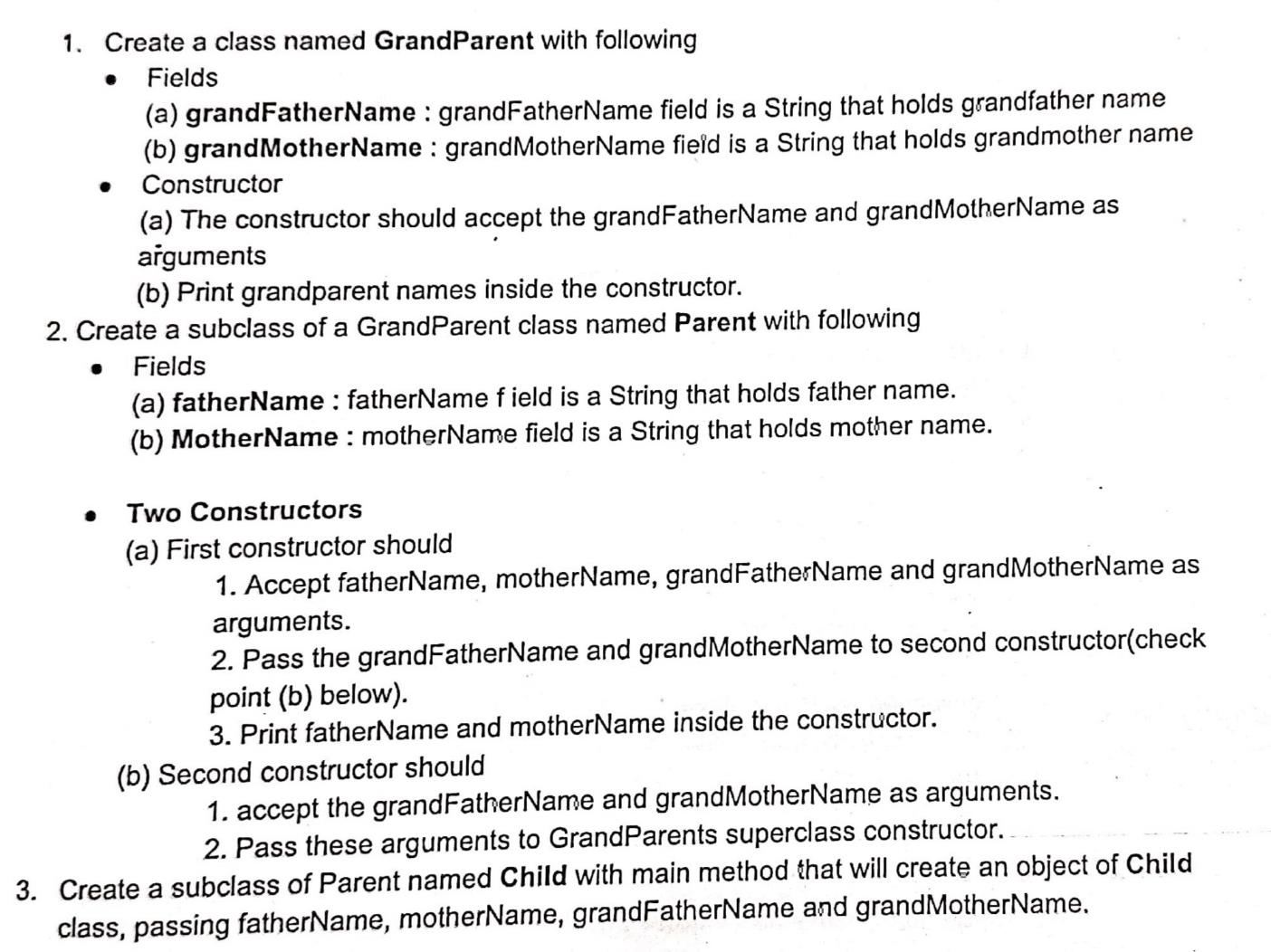
}

}

**Output**

****

**Q.4 Constructor Chaining:**



*//class GrandParent*

class GrandParent1

{

String grandFatherName;

String grandMotherName;

*//Constructor of class GrandParent*

public GrandParent1(String grandFatherName, String grandMotherName)

{

this.grandFatherName=grandFatherName;

this.grandMotherName=grandMotherName;

System.*out*.println("Name of Grand Father is : "+grandFatherName);

System.*out*.println("Name of Grand Mother is : "+grandMotherName);

}

}

*//subclass Parent of class GrandParent*

class Parent1 extends GrandParent1

{

String FatherName;

String MotherName;

*//constructor 1*

public Parent1(String FatherName,String MotherName,String grandFatherName, String grandMotherName)

{

super(grandFatherName, grandMotherName);

this.FatherName=FatherName;

this.MotherName=MotherName;

System.*out*.println("Name of Father is : "+FatherName);

System.*out*.println("Name of Mother is : "+MotherName);

}

*//constructor 2 : constructor of Parent class*

public Parent1(String grandFatherName, String grandMotherName) {

super(grandFatherName, grandMotherName);

}

}

public class Child1 extends Parent1

{

public Child1(String FatherName, String MotherName, String grandFatherName, String grandMotherName) {

super(FatherName, MotherName, grandFatherName, grandMotherName);

}

public static void main(String args[])

{

Child1 c = new Child1("Ram","Seeta","Sham","Neeta");

*//System.out.println(c.FatherName+ " "+c.MotherName+ " "+c.grandFatherName+ " "+c.grandMotherName);*

}

}

**OutPut:**

