**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)

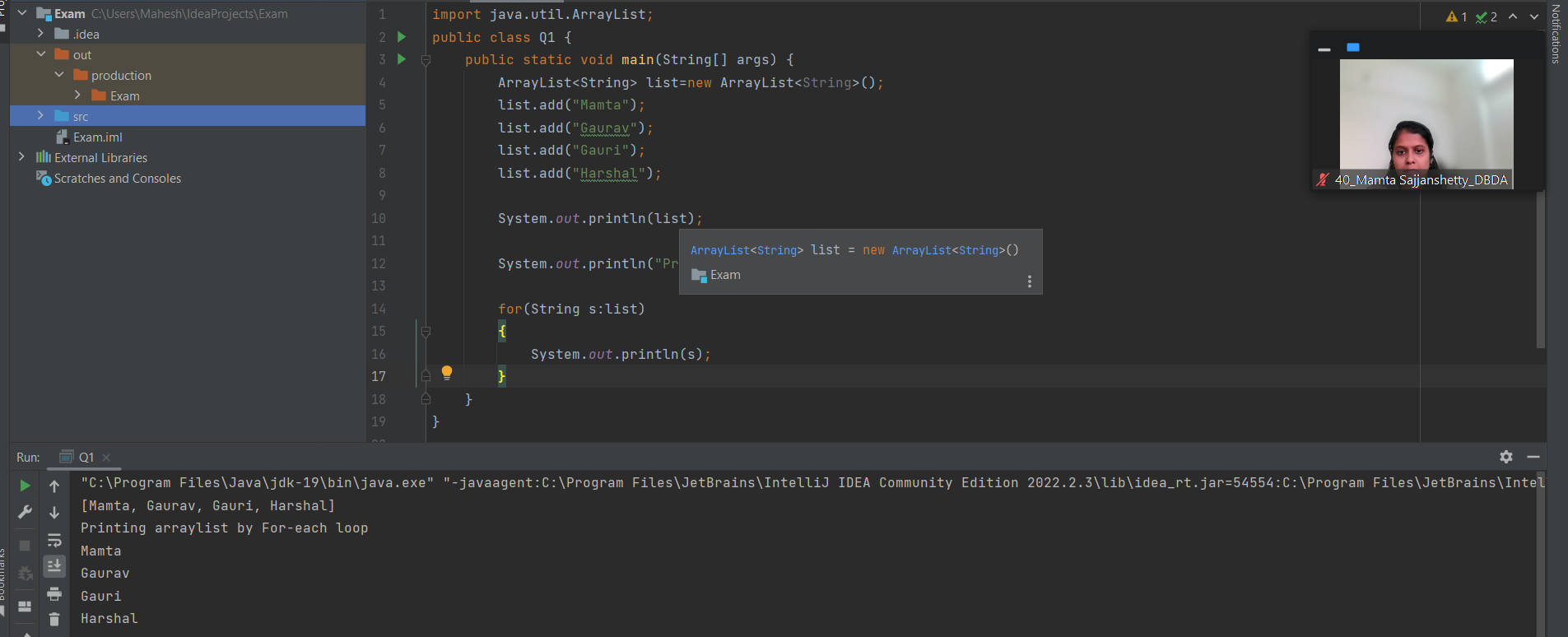
import java.util.ArrayList;  
public class Q1 {  
 public static void main(String[] args) {  
 ArrayList<String> list=new ArrayList<String>();  
 list.add("Mamta");  
 list.add("Gaurav");  
 list.add("Gauri");  
 list.add("Harshal");  
  
 System.*out*.println(list);  
  
 System.*out*.println("Printing arraylist by For-each loop");  
  
 for(String s:list)  
 {  
 System.*out*.println(s);  
 }  
 }  
}

Graphical user interface, text, application

Description automatically generated

––––––Text

Description automatically generated



**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)

class Shape  
{  
 void draw()  
 {  
 System.*out*.println("draw method of parent shape class ");  
 }  
  
 void erase()  
 {  
 System.*out*.println("erase method of parent shape class");  
 }  
  
}  
  
class circle extends Shape  
{  
 void draw()  
 {  
 System.*out*.println("draw method of child circle class ");  
 }  
  
 void erase()  
 {  
 System.*out*.println("erase method of child circle class ");  
 }  
  
}  
  
class triangle extends Shape  
{  
 void draw()  
 {  
 System.*out*.println("draw method of child triangle class ");  
 }  
  
 void erase()  
 {  
 System.*out*.println("erase method of child triangle class ");  
 }  
}  
  
class Square extends Shape  
{  
 void draw()  
 {  
 System.*out*.println("draw method of child Square class ");  
 }  
  
 void erase()  
 {  
 System.*out*.println("erase method of child Square class ");  
 }  
}  
  
  
  
public class Q3 {  
 public static void main(String[] args) {  
 System.*out*.println("Program started");  
 circle c=new circle();  
 triangle t=new triangle();  
 Square s=new Square();  
  
 System.*out*.println("calling draw() methods Overriden from parent class shape in all child classes ");  
 c.draw();  
 t.draw();  
 s.draw();  
  
 System.*out*.println("calling erase() methods Overriden from parent class shape in all child classes ");  
 c.erase();  
 t.erase();  
 s.erase();  
 System.*out*.println("Program completed");  
  
  
 }  
}

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

Text

Description automatically generated

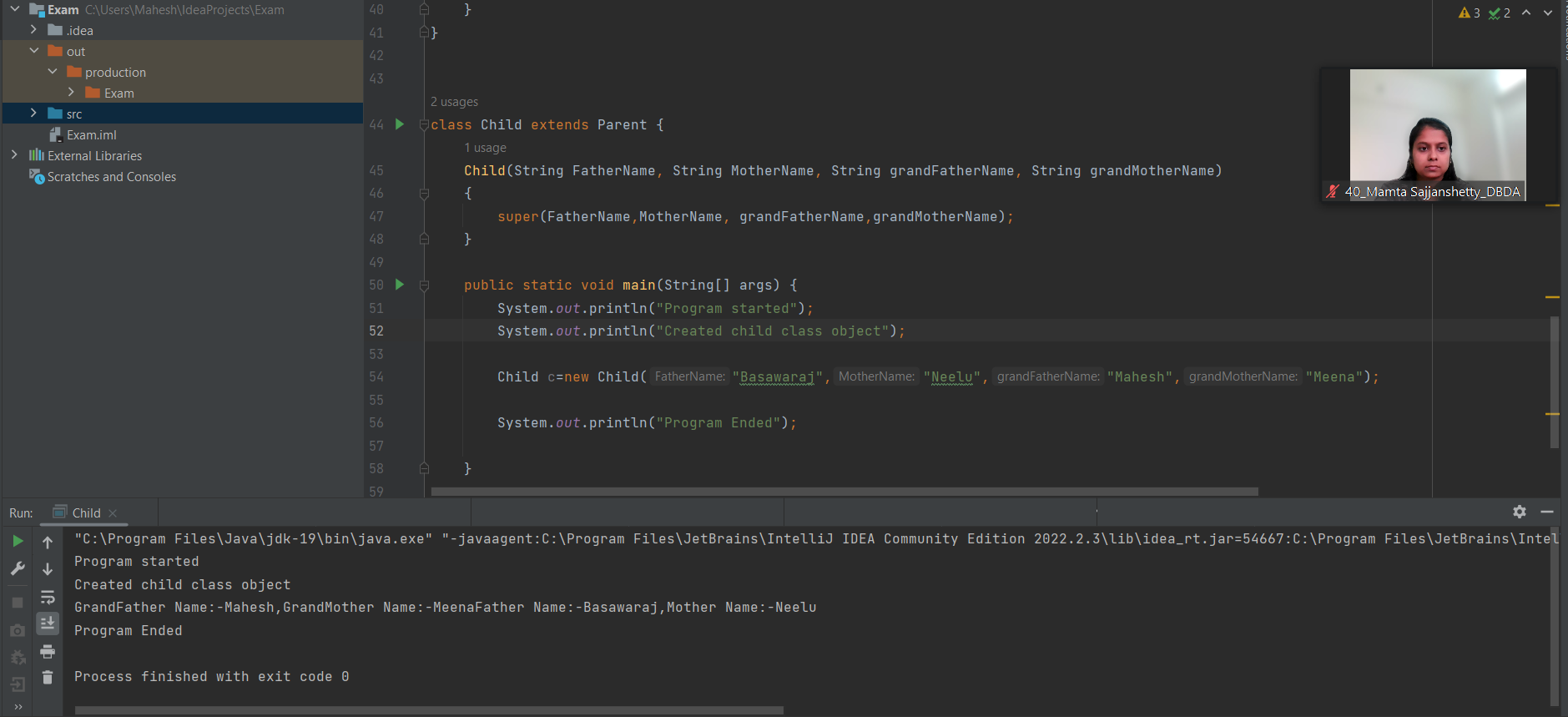
Text

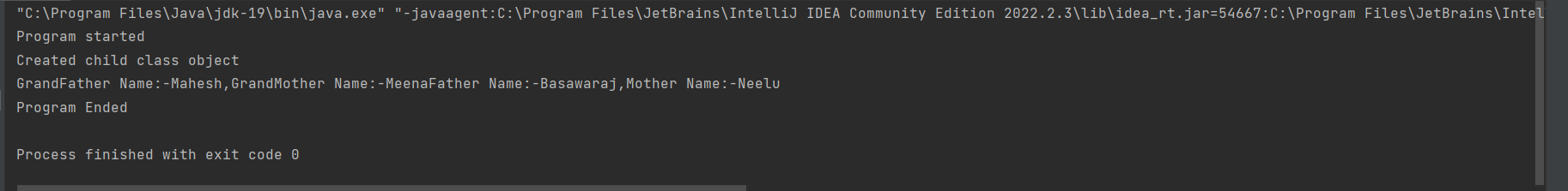
Description automatically generated

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



class GrandParent  
{  
 String grandFatherName;  
 String grandMotherName;  
 GrandParent()  
 {  
  
 }  
 GrandParent(String grandFatherName, String grandMotherName)  
 {  
 this.grandFatherName=grandFatherName;  
 this.grandMotherName=grandMotherName;  
 System.*out*.println("GrandFather Name:"+grandFatherName+"GrandMother Name:"+grandMotherName);  
 }  
  
}  
  
class Parent extends GrandParent  
{  
 String FatherName;  
 String MotherName;  
Parent()  
{  
  
}  
 Parent(String FatherName, String MotherName, String grandFatherName, String grandMotherName)  
 {  
 this.FatherName=FatherName;  
 this.MotherName=MotherName;  
 this.grandFatherName=grandFatherName;  
 this.grandMotherName=grandMotherName;  
 System.*out*.println("GrandFather Name:"+"-"+grandFatherName+","+"GrandMother Name:"+"-"+grandMotherName+"Father Name:"+"-"+FatherName+","+"Mother Name:"+"-"+MotherName);  
 }  
  
  
 Parent(String x, String y)  
 {  
 super(x,y);  
 }  
}  
  
  
class Child extends Parent {  
 Child(String FatherName, String MotherName, String grandFatherName, String grandMotherName)  
 {  
 super(FatherName,MotherName, grandFatherName,grandMotherName);  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("Program started");  
 System.*out*.println("Created child class object");  
  
 Child c=new Child("Basawaraj","Neelu","Mahesh","Meena");  
  
 System.*out*.println("Program Ended");  
  
 }  
  
}





**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.

import java.util.Scanner;  
  
public class BankAccount {  
 int accno;  
 double balance;  
  
 Scanner sc=new Scanner(System.*in*);  
 BankAccount (int acc,double bal)  
 {  
 this.accno=acc;  
 this.balance=bal;  
 }  
  
 void withdraw() throws Exception  
 {  
 double amount;  
  
 System.*out*.println("Enter the amount");  
 amount=sc.nextInt();  
 try {  
 if (balance >= amount) {  
 balance = balance - amount;  
 System.*out*.println("After withdrawal remaining balance is:" + balance);  
 } else {  
 throw new Exception();  
 }  
 }catch(NullPointerException e)  
 {  
 System.*out*.println("balance is:"+balance+"therefore withdrawal of amount:"+amount+"is not allowed");  
 }  
 }  
  
 void Deposit()  
 {  
 int amount;  
 System.*out*.println("Enter the amount to be deposited");  
 amount=sc.nextInt();  
 balance=balance+amount;  
  
 }  
  
 void show()  
 {  
 System.*out*.println("Account number:"+accno+"-----"+"having balance"+balance);  
 }  
  
  
 public static void main(String[] args) {  
 System.*out*.println("Program started");  
 BankAccount b1=new BankAccount(2342,6000.00);  
 try {  
 b1.withdraw();  
 }  
 catch(Exception e)  
 {  
 System.*out*.println("balance amount is less than the withdrawal amount. Please enter the valid amount that you wanted to withdraw");  
 }  
 b1.Deposit();  
 b1.show();  
 System.*out*.println("Program ended");  
 // sc=new Scanner(System.in);  
 }

}

Output without exception:

A picture containing text

Description automatically generated

Output with exception raised:

Text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence