Java module Exam

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.Arrays;

class StringArray

{

public static void main(String args[])

{

String[] SubNames = new String[]{"linux", "cloud computing", "python", "java", "big data"};

System.*out*.println(" String Array list");

for(int i=0; i < SubNames.length; i++)

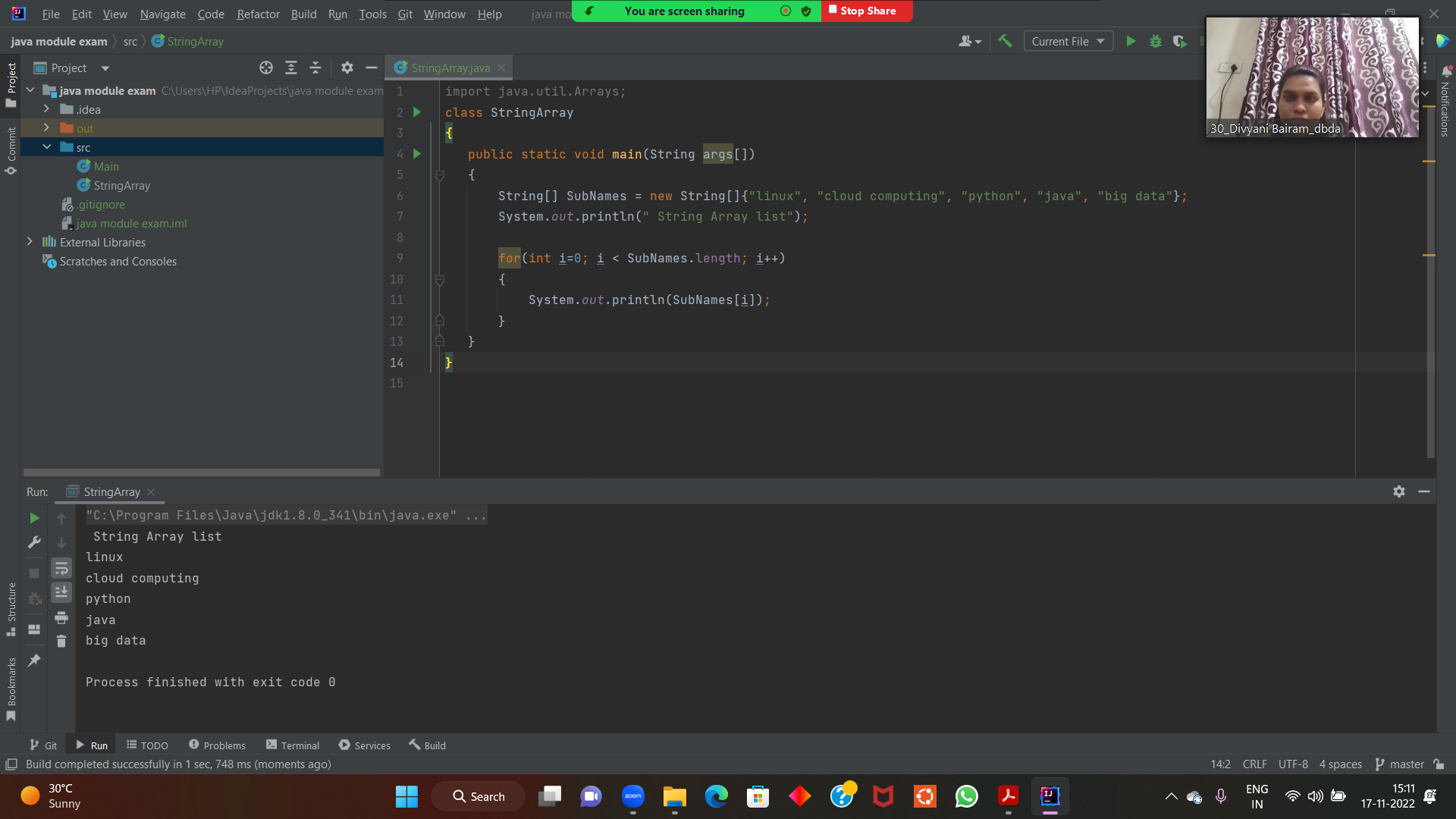
{

System.*out*.println(SubNames[i]);

}

}

}



Q2 : Develop a class BankAccount having following data members : (10 Marks)

int accno

double balance

import java.util.

public class Bank {

private String accno;

private String name;

private Double balance;

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

void openAccount()

{

System.*out*.println("Enter Account Number :-");

accno= KB.next();

System.*out*.println("Enter Name :-");

name= KB.next();

System.*out*.println("Enter Balance :-");

balance= KB.nextDouble();

}

void showAccount()

{

System.*out*.println(accno + "," + name + "," + balance);

}

void deposit()

{

Double amt;

System.*out*.println("Enter Amount u want to deposit :- ");

amt = KB.nextDouble();

balance = balance + amt;

}

void withdrawal() {

Double amt;

System.*out*.println("Enter amount u want to withdraw :-");

amt = KB.nextDouble();

if (balance >= amt) {

balance = balance - amt;

} else {

System.*out*.println("less balance.. Transcation failed..");

}

}

}

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("Drawing Shape");

}

void erase(){

System.*out*.println("Erasing Shapes");

}

}

class Circle extends Shape{

@Override

void draw() {

System.*out*.println("Drawing Circle");

}

@Override

void erase(){

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

}

}

class Triangle extends Shape{

@Override

void draw() {

System.*out*.println("drawing triangle");

}

@Override

void erase() {

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

}

}

class Square extends Shape{

@Override

void draw() {

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

}

@Override

void erase() {

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

}

}

class Solution {

public static void main(String args[]) {

Shape c = new Circle();

Shape t = new Triangle();

Shape s = new Square();

c.draw();

c.erase();

t.draw();

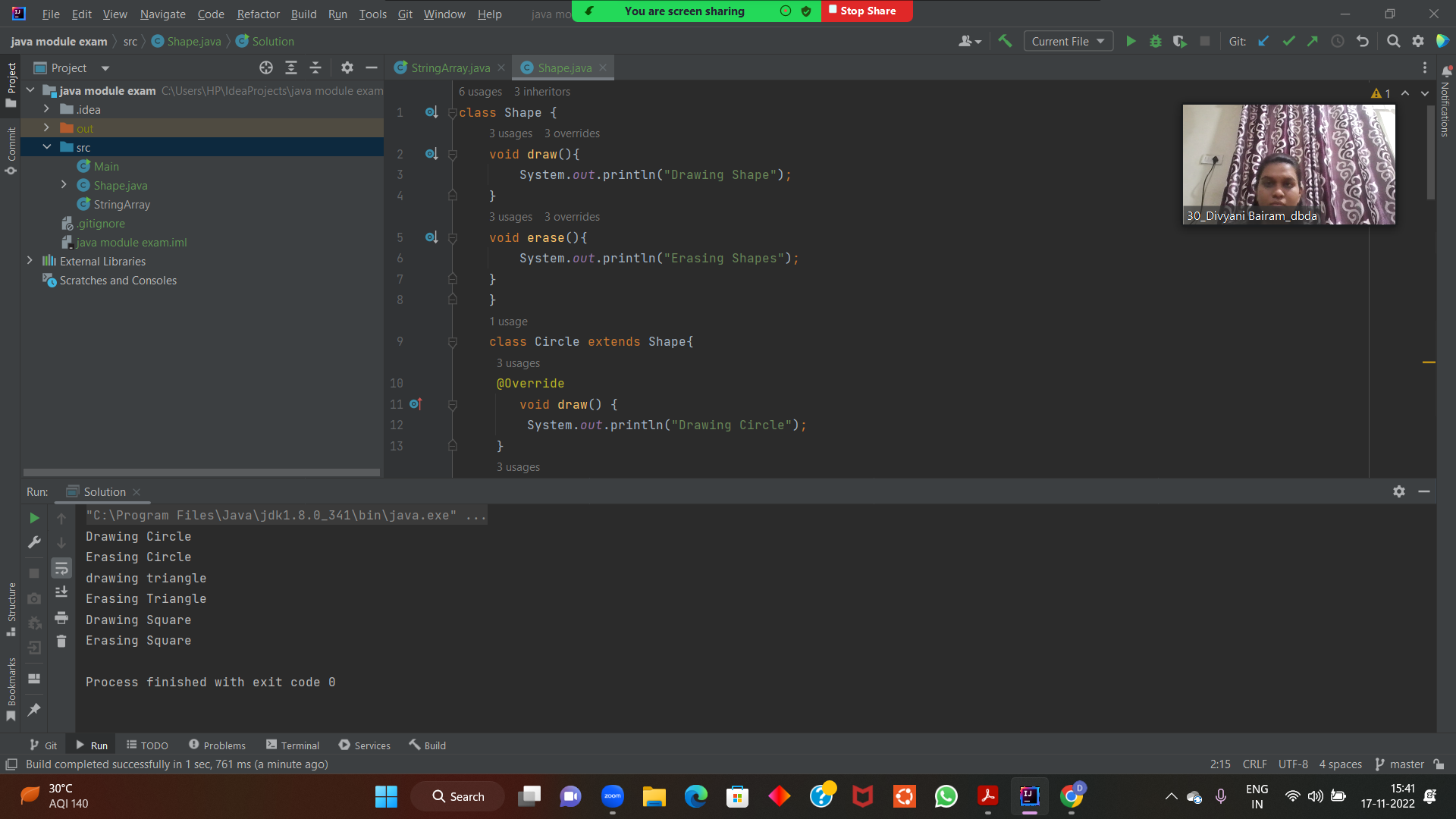
t.erase();

s.draw();

s.erase();

}

}



Q4 : Constructor chaining (10 Marks)

class grandParent{

public String grandFathername,grandMothername;

public grandParent(String a, String b) {

grandFathername = a;

grandMothername = b;

System.*out*.println("Grandfather's name : " + grandFathername + " " + " Grandmother's name :" + grandMothername);

}

}

class parent extends grandParent{

public String fatherName,motherName;

public parent(String a,String b,String c, String d) {

this(c, d);

fatherName = a;

motherName = b;

System.*out*.println("Father's Name :" + fatherName + " " + " Mother's name : " + motherName);

}

public parent(String a,String b) {

super(a,b);

}

}

class child extends parent{

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

super(a, b, c, d);

}

public static void main(String args[]) {

child c1 = new

child("father", "mother", "grandfather", "grandmother");

}

}

