**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 Question1

{

public static void main(String args [])

{

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

list.add("Red");

list.add("Yellow");

list.add("Green");

list.add("Orange");

list.add("White");

System.out.println("ArrayList Element are:"+list);

System.out.println("Now iterating ArrayList Element one by one using for loop And Printing Them");

for(int i=0;i<list.size();i++)

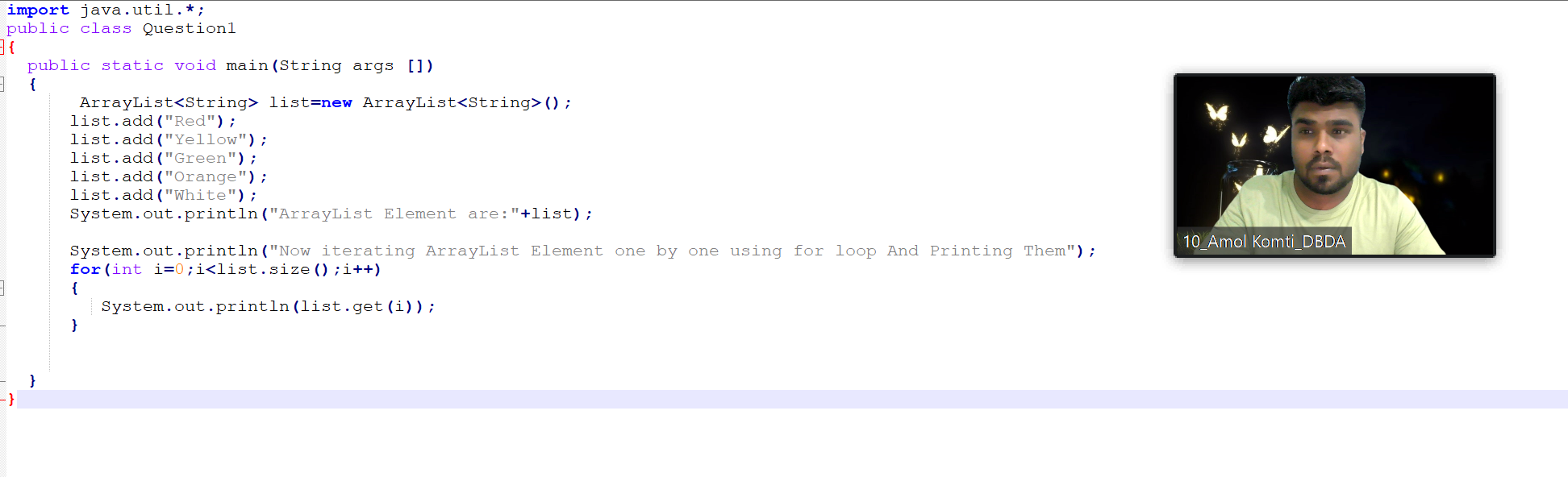
{

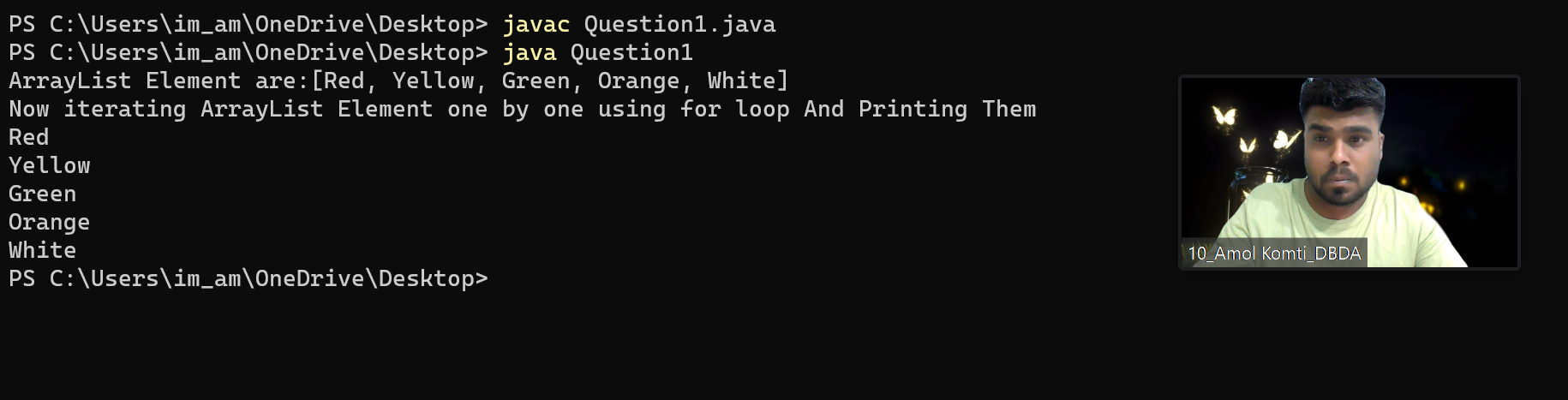
System.out.println(list.get(i));

}

}

}





**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 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.

// Class 1

// Bank class

// Defining the banking transaction

class Bank {

// Initial balance $100

int total = 100;

// Money withdrawal method. Withdraw only if

// total money greater than or equal to the money

// requested for withdrawal

// Method

// To withdraw money

void withdrawn(String name, int withdrawal)

{

if (total >= withdrawal) {

System.out.println(name + " withdrawn "

+ withdrawal);

total = total - withdrawal;

System.out.println("Balance after withdrawal: "

+ total);

// Making the thread sleep for 1 second after

// each withdrawal

// Try block to check for exceptions

try {

// Making thread t osleep for 1 second

Thread.sleep(1000);

}

// Catch block to handle the exceptions

catch (InterruptedException e) {

// Display the exception along with line

// number

// using printStacktrace() method

e.printStackTrace();

}

}

// If the money requested for withdrawal is greater

// than the balance then deny transaction\*/

else {

// Print statements

System.out.println(name

+ " you can not withdraw "

+ withdrawal);

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

// Making the thread sleep for 1 second after

// each transaction failure

// Try block to check for exceptions

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

// Method - to deposit money

// Accept money whenever deposited

void deposit(String name, int deposit)

{

System.out.println(name + " deposited " + deposit);

total = total + deposit;

System.out.println("Balance after deposit: "

+ total);

// Making the thread sleep for 1 second after

// each deposit

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

// Class 2

// main class

class Bank2 {

// Main driver method

public static void main(String[] args)

{

// Declaring an object of Bank class and calling the

// withdarwn and deposit methods with suitable

// parameters

// Creating object of class 1 inside main()

Bank obj = new Bank();

// Custom input - Transactions

obj.withdrawn("Ajay Fatpure", 20);

obj.withdrawn("Aishwarya Bhalbhar", 40);

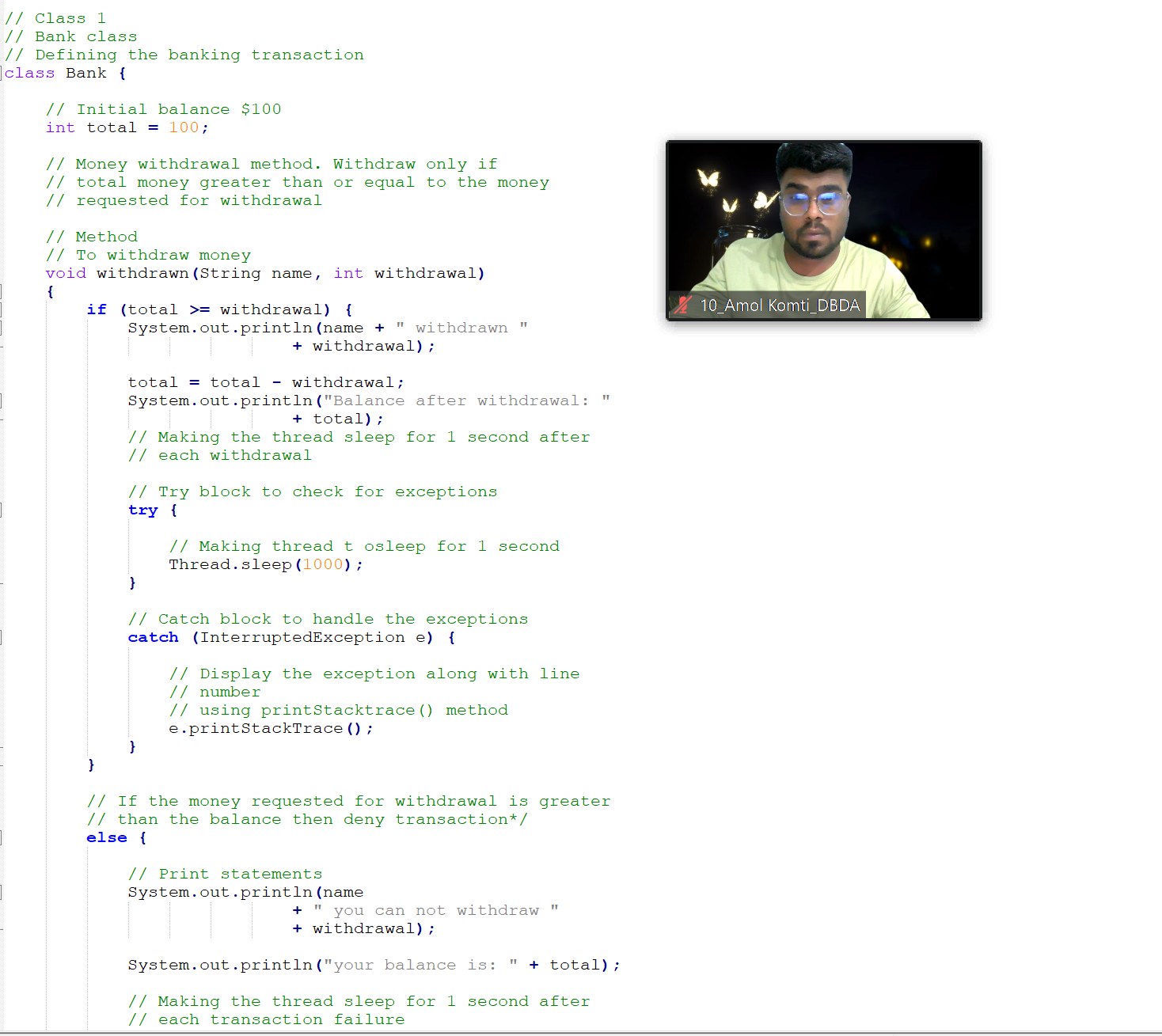
obj.deposit("Aman kumar", 35);

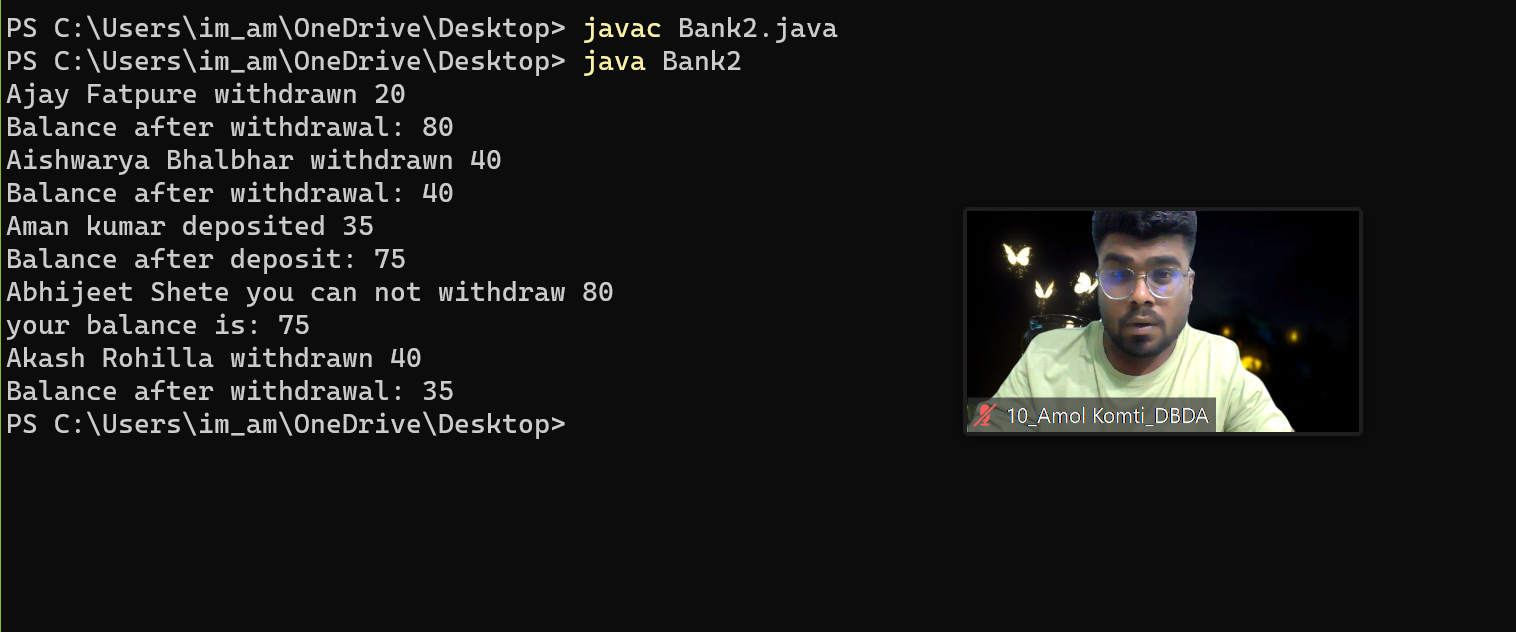
obj.withdrawn("Abhijeet Shete", 80);

obj.withdrawn("Akash Rohilla", 40);

}

}





**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.

import java.math.\*;

class Shape

{

void draw()

{

System.out.println("Shape draw()");

}

void erase()

{

System.out.println (" Shape erase()");

}

}

class Circle extends Shape

{

void draw()

{

System.out.println ("Circle draw()");

}

void erase()

{

System.out.println ("Circle erase()");

}

}

class Triangle extends Shape

{

void draw()

{

System.out.println("Triangle erase()");

}

}

class Square extends Shape

{

void draw()

{

System.out.println ("Square draw()");

}

void erase()

{

System.out.println ("Square erase()");

}

}

class Shapes

{

public static Shape randshape()

{

switch((int)(Math.random()\*3))

{

case 0: return new Circle();

case 1: return new Square();

case 2: return new Triangle();

default : System.out.println("default");

return new Shape();

}

}

public static void main (String arg[])

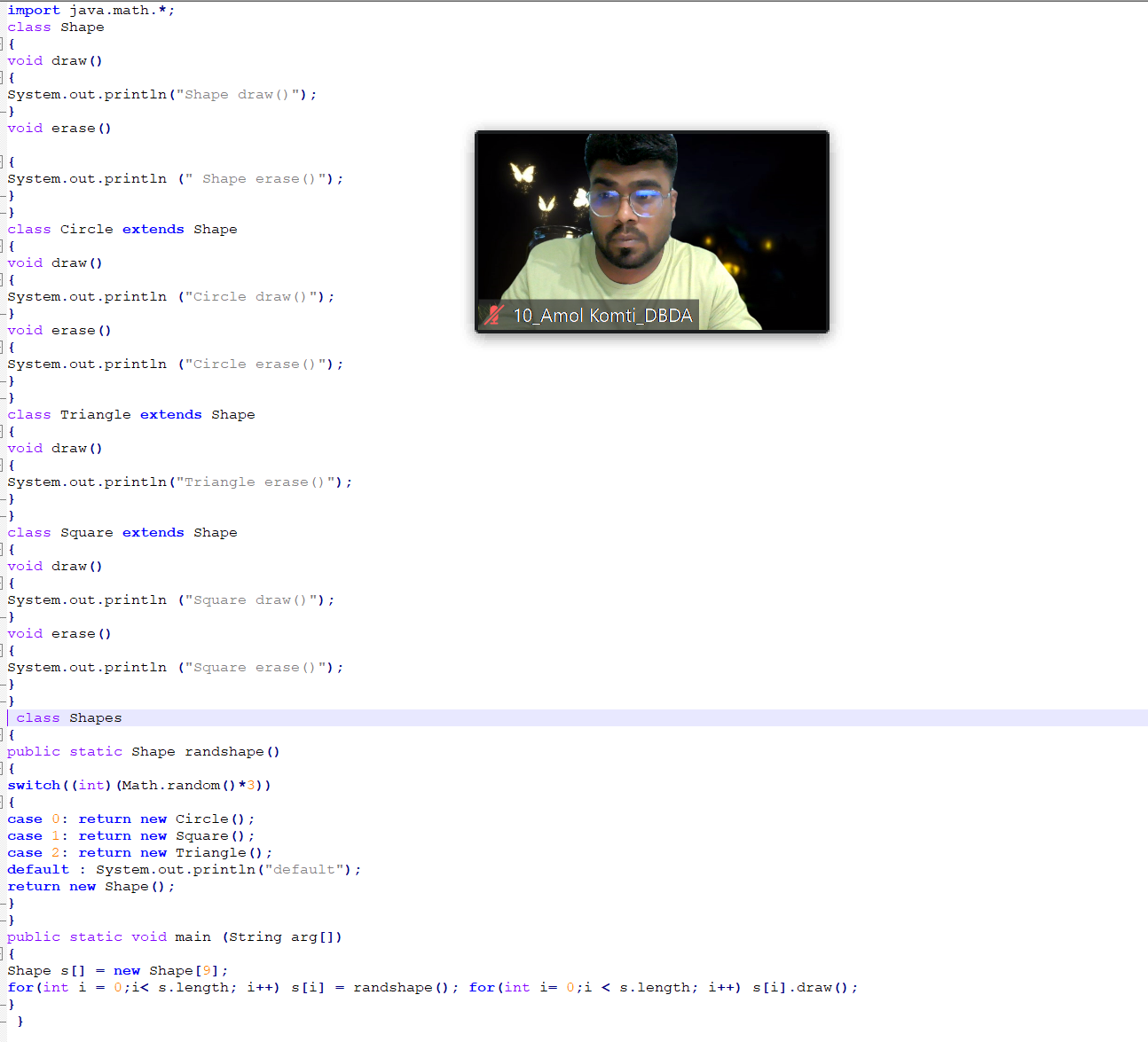
{

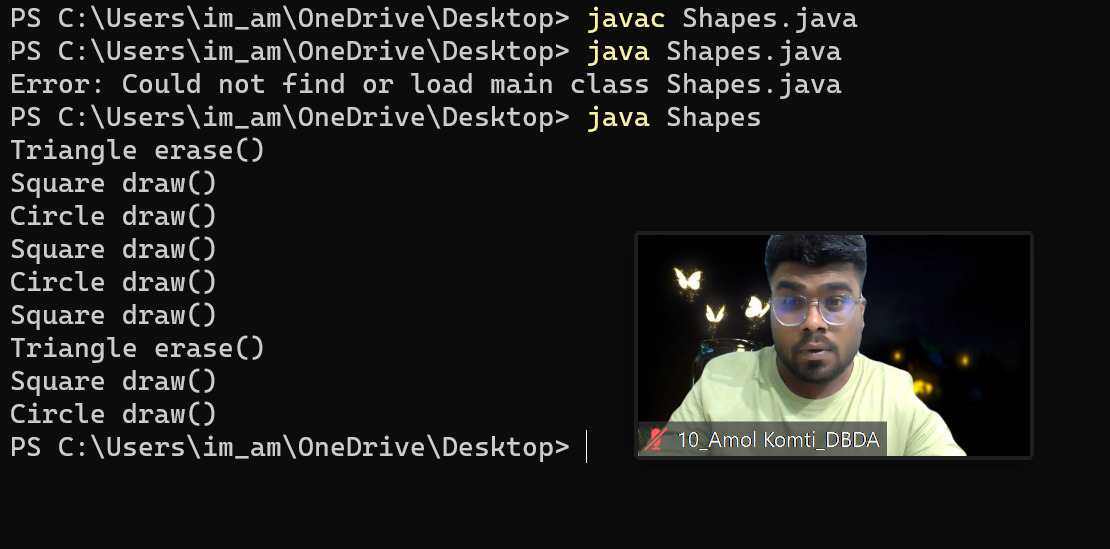
Shape s[] = new Shape[9];

for(int i = 0;i< s.length; i++) s[i] = randshape(); for(int i= 0;i < s.length; i++) s[i].draw();

}

}





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

