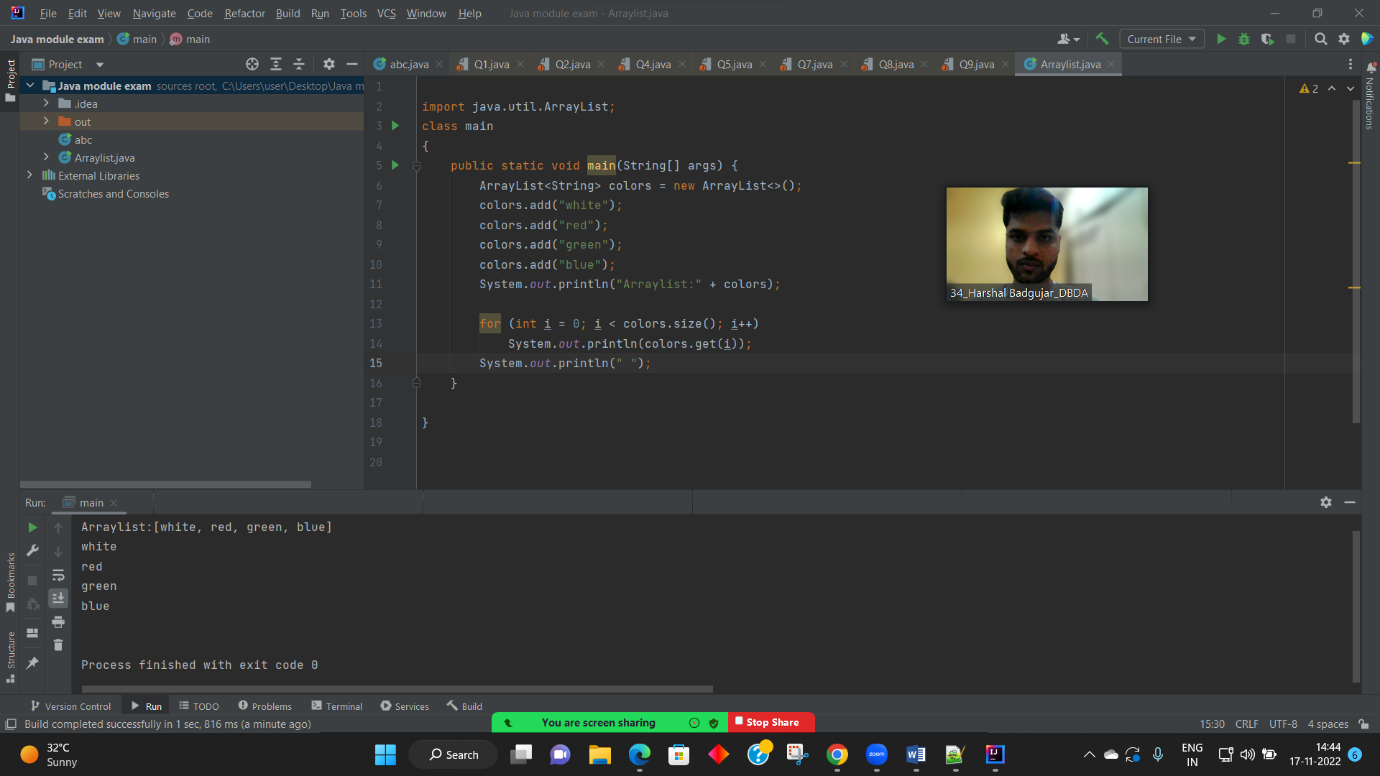
Q1 : Write a Java program to create a new array list, add some elements (string) and print out the collection by using for-each loop. (10 Marks

import java.util.ArrayList;  
class main  
{  
 public static void main(String[] args) {  
 ArrayList<String> colors = new ArrayList<>();  
 colors.add("white");  
 colors.add("red");  
 colors.add("green");  
 colors.add("blue");  
 System.*out*.println("Arraylist:" + colors);  
  
 for (int i = 0; i < colors.size(); i++)  
 System.*out*.println(colors.get(i));  
 System.*out*.println(" ");  
 }  
  
}

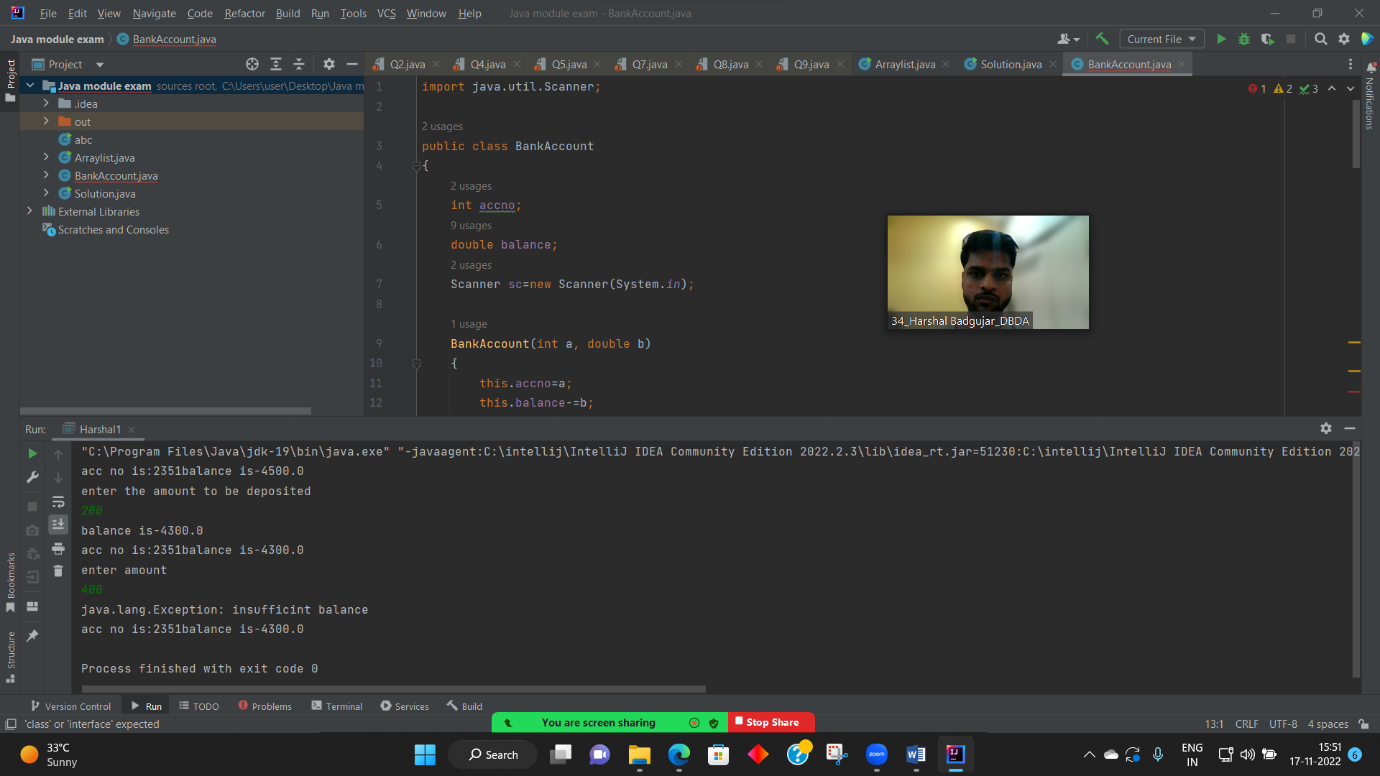


Q3. Write a program to create a class named shape. In this class we have three sub classes circle, triangle and square, each class has two member function named draw () and erase (). Create these using Runtime Polymorphism concepts. (10 Marks

class Shape {  
 void draw() {  
 System.*out*.println("Drawing Shape");  
 }  
 void erase() {  
 System.*out*.println("Erasing Shape");  
 }  
 }  
 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) {  
 // TADO STUD  
 Shape c = new Circle();  
 Shape t = new Triangle();  
 Shape s = new Square();  
 c.draw(); c.erase();  
 t.draw(); t.erase();  
 s.draw(); s.erase();  
 }  
 }

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 a, double b)  
 {  
 this.accno=a;  
 this.balance-=b;  
 }  
 void withdraw()throws Exception  
 {  
 double amount;  
  
 System.*out*.println("enter amount");  
 amount=sc.nextInt();  
 if(balance>=amount)  
 {  
 balance=balance-amount;  
 System.*out*.println(balance);  
 }  
 else{  
 throw new Exception("insufficint balance");  
 }  
 }  
 void deposit()  
 {  
 int amount;  
 System.*out*.println("enter the amount to be deposited");  
 amount=sc.nextInt();  
 if(amount>=0)  
 {  
 balance=balance+amount;  
 System.*out*.println("balance is"+balance);  
 }  
 else{  
 System.*out*.println("invalid amount");  
 }  
 }  
 void show()  
 {  
 System.*out*.println("acc no is:"+accno+"balance is"+balance);  
 }  
  
}  
class Harshal1{  
 public static void main(String[] args) {  
 BankAccount bank = new BankAccount(2351, 4500);  
 Scanner sc = new Scanner(System.*in*);  
 bank.show();  
 bank.deposit();  
 bank.show();  
 try {  
 bank.withdraw();  
 } catch (Exception b) {  
 System.*out*.println(b);  
 }  
 bank.show();  
 }  
}



Q4.constructor channing

class GrandParent {  
 String grandfathername;  
 String grandmothername;  
  
 GrandParent(String GFN,String GMN){  
 grandfathername=GFN;  
 grandmothername=GMN;  
  
 System.*out*.println("grand father name:"+grandfathername);  
 System.*out*.println("grand mother name:"+grandmothername);  
 }  
}  
class Parent extends GrandParent{  
 String fathername;  
 String mothername;  
  
 Parent(String FN,String MN,String GFN,String GMN){  
 super(GFN,GMN);  
  
 fathername=FN;  
 mothername=MN;  
  
 System.*out*.println("father name:"+fathername);  
 System.*out*.println("mother name:"+mothername);  
 }  
}  
class child extends Parent {  
  
 child(String FN,String MN,String GFN,String GMN){  
 super(FN,MN,GFN,GMN);  
 }  
}