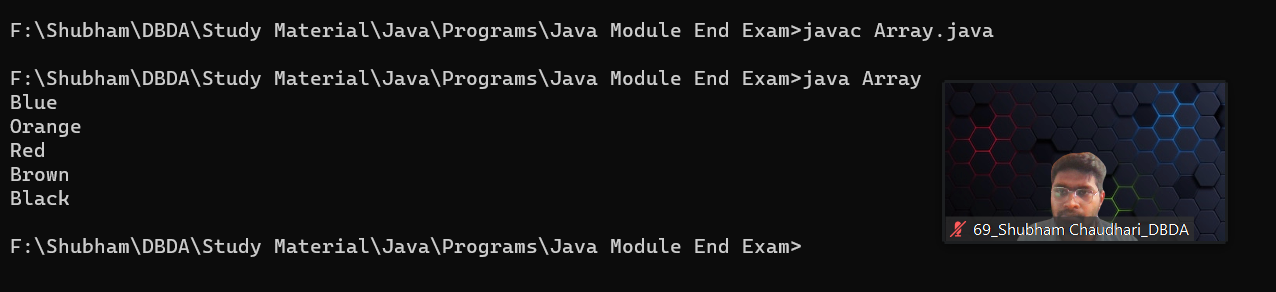
**Question 1 : -**

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

**Program : -**



**Output : -**

****

**Question 2 : -**

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.

**Program : -**

**class Balance extends Exception**

**{**

**Balance(String s)**

**{**

**super(s);**

**}**

**}**

**class BankAccount1**

**{**

**int accno;**

**double balance;**

**BankAccount1(int a, double b)**

**{**

**accno=a;**

**balance=b;**

**}**

**void widthdraw(int width)**

**{**

**if (width > balance)**

**{**

**try**

**{**

**throw new Balance("Insufficient Balance Process terminated");**

**}**

**catch(Balance e)**

**{**

**System.out.println(e.getMessage());**

**}**

**}**

**else**

**{**

**balance=balance-width;**

**System.out.println("New Balance after widthdrawal : " +width + " is " +balance);**

**}**

**}**

**void deposit(int dep)**

**{**

**balance=balance + dep;**

**System.out.println("New balance after depositing is : " +balance);**

**}**

**void display()**

**{**

**System.out.println(" Your account no is : " +accno);**

**System.out.println("Available balance is : " +balance);**

**}**

**public static void main(String args[])**

**{**

**BankAccount1 obj=new BankAccount1(2401, 10000);**

**obj.widthdraw(15000);**

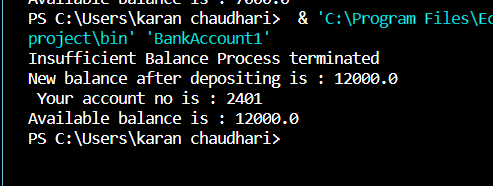
**obj.deposit(2000);**

**obj.display();**

**}**

**}**

**Output : -**

****

**Question 3 : -**

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.

**Program : -**

class Shape

{

void draw()

{

System.out.println("Shape is Drawing");

}

void erase()

{

System.out.println("Shape is Erasing");

}

}

class Circle extends Shape

{

void draw()

{

System.out.println("Circle is drawing");

}

void erase()

{

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

}

}

class Triangle extends Shape

{

void draw()

{

System.out.println("Triangle is Drawing");

}

void erase()

{

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

}

}

class Square extends Shape

{

void draw()

{

System.out.println("Shape is Drawing");

}

void erase()

{

System.out.println("Shape is Erasing");

}

}

class NewShape

{

public static void main(String[] args )

{

Shape obj1=new Circle();

Shape obj2=new Triangle();

Shape obj3=new Square();

obj1.draw();

obj1.erase();

obj2.draw();

obj2.erase();

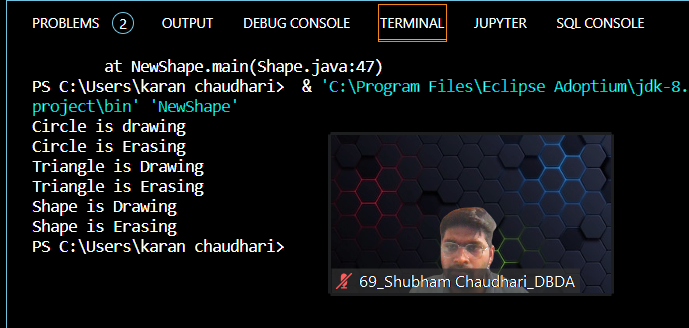
obj3.draw();

obj3.erase();

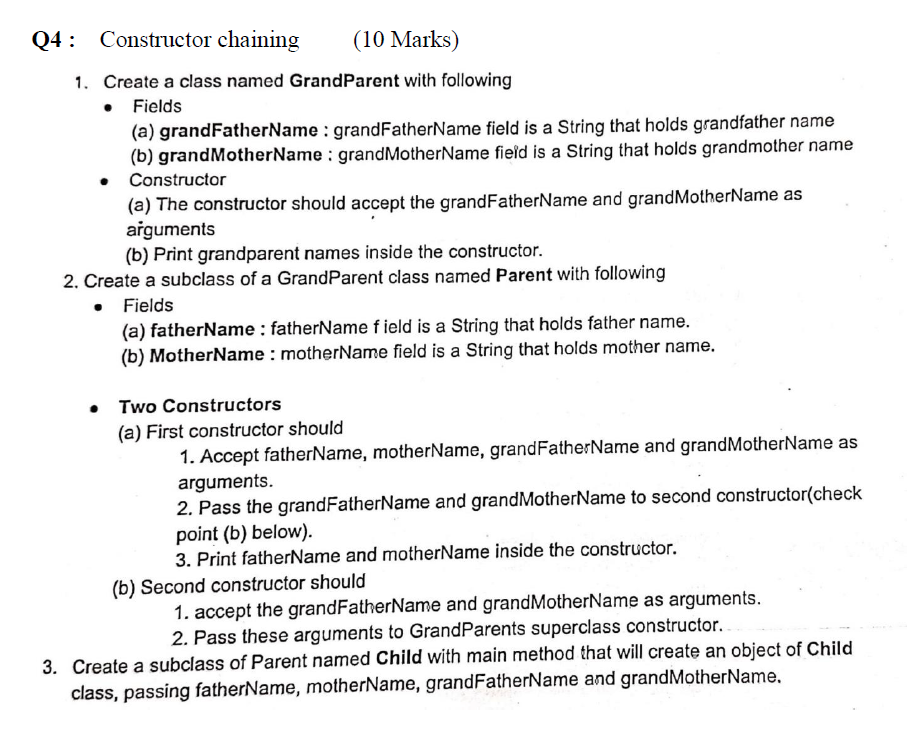
}

}

**Output : -**

****

**Question 4 : -**

****

**Program : -**

class GrandParent

{

String grandFatherName, grandMotherName;

GrandParent(String a, String b)

{

grandFatherName=a;

grandMotherName=b;

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

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

}

}

class Parent extends GrandParent

{

String FatherName, MotherName;

Parent(String c, String d, String a, String b)

{

super(a,b);

FatherName=c;

MotherName=d;

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

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

}

}

class Child extends Parent

{

Child(String c, String d, String a, String b)

{

super(c, d, a, b);

}

public static void main(String args[])

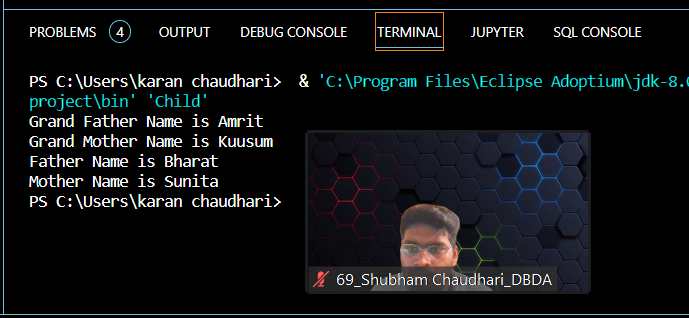
{

Child obj=new Child("Bharat", "Sunita", "Amrit", "Kuusum");

}

}

**Output : -**

****