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.

package ExamJava;

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 Draw");

}

void Erase(){

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

}

}

class Square extends Shape{

void Draw(){

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

}

void Erase(){

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

}

}

public class Question3 {

public static void main(String[] args) {

Shape sha =new Shape();

Circle cir=new Circle();

Triangle tri =new Triangle();

Square squ = new Square();

System.*out*.println("Runtime polymorphism :");

System.*out*.println("When sha object created from class Shape:");

sha.Draw();

sha.Erase();

System.*out*.println("When cir object created from class Circle:");

cir.Draw();

cir.Erase();

System.*out*.println("When tri object created from class Triangle:");

tri.Draw();

tri.Erase();

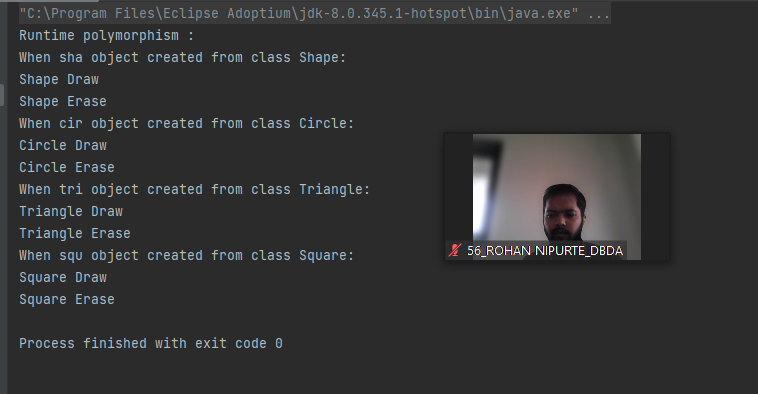
System.*out*.println("When squ object created from class Square:");

squ.Draw();

squ.Erase();

}

}



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.

package ExamJava;

import java.util.\*;

public class Question1 {

public static void main(String[] args) {

List<String> listStrings=new ArrayList<String>();

System.*out*.println("Adding Fruits:");

listStrings.add("Apple");

listStrings.add("Banana");

listStrings.add("Kiwi");

listStrings.add("Grapes");

listStrings.add("Cherry");

listStrings.add("Watermelon");

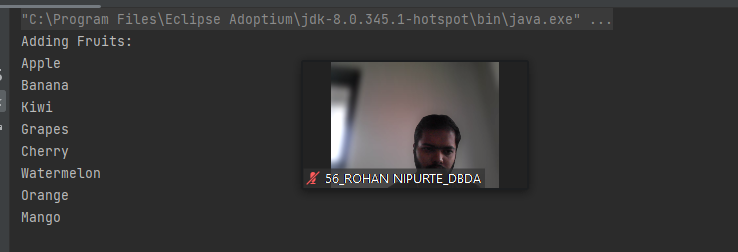
listStrings.add("Orange");

listStrings.add("Mango");

listStrings.forEach((fruit)-> System.*out*.println(fruit));

}

}



Q4 : Constructor chaining

package ExamJava;

class Grandparent{

String GrandFatherName;

String GrandMotherName;

public Grandparent(String GFname,String GMname){

this.GrandFatherName=GFname;

this.GrandMotherName=GFname;

System.*out*.println("Name of GrandFather:"+this.GrandFatherName);

System.*out*.println("Name of GrandMotherName:"+this.GrandMotherName);

}

}

class Parent extends Grandparent {

String FatherName;

String MotherName;

public Parent(String Fname,String Mname,String GFname,String GMname) {

super(GFname,GMname);

this.FatherName=Fname;

this.MotherName=Mname;

System.*out*.println("Father name:" + this.FatherName);

System.*out*.println("Father name:" + this.MotherName);

}

}

class Child extends Parent{

Child(String Fname,String Mname,String GFname,String GMname){

super(Fname,Mname,GFname,GMname);

}

}

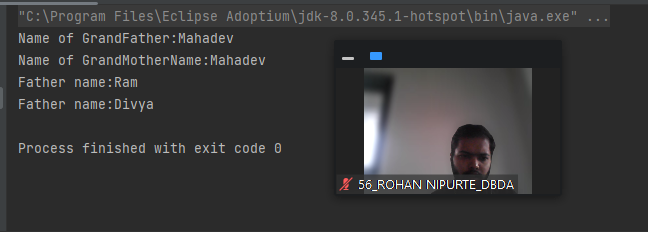
public class Question4 {

public static void main(String[] args) {

Child chi=new Child("Ram","Divya","Mahadev","Seeta");

}

}



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.

package ExamJava;

import java.util.\*;

class BankAccount{

int accno;

double balance;

public BankAccount(int accno,double Balance){

this.accno=accno;

this.balance=balance;

}

public void withdraw(double amount){

if (balance<amount) {

try {

System.*out*.println("invalid ammount");

} catch (Exception e) {

System.*out*.println("Withrdawal ammount shoulde less than balance");

}

}

else {

balance=balance-amount;

System.*out*.println("After withdrawl:"+balance);

}

}

public void deposite(double amount){

balance=balance-amount;

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

}

public void show(){

System.*out*.println("Account no:"+accno);

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

}

}

public class Question2 {

public static void main(String[] args) {

BankAccount ba =new BankAccount(10,80);

ba.show();

ba.deposite(1000);

ba.withdraw(1000);

}

}

