**Name**: Shradha Mallikarjun Patil

**USN:** 2GI20CS144

**TERMWORK 8.2:**

8.2) Design a class CustomerAccount that has acctNum, custName and balance as member variables and a constructor to initialize these. Implement withDraw and depositAmount methods that accepts amount as argument and it must throw an user defined exception called InsufficientBalalance/InvalidAmount exception when amount is greater than balance/ amount is negative respectively. Design two classes InsufficientBalance and InvalidAmount that extend the Exception class and override toString method. Demonstrate the working of the user defined exceptions by instantiating an object of customerAccount class and invoking withDraw and depoistAmount in try… catch.. finally block.

Exception

InsufficientBalance

+getMessage()

InvalidAmount

+getMessage( )

CustomerAccount

-accNum -balance -custName

+withDraw(float amount ) throws InsufficientBalance

+depositAmount(float amount ) throws InvalidAmount

**Program:**

DemoClass { public static void main(String[] args){

CustomerAccount newCustomer=new CustomerAccount( );

try{ newCustomer.withDraw(7000);

newCustomer.depositAmount(-500);

}catch(InsufficientBalance e) { e.getMessage( );}

catch(InvalidAmount e) { e.getMessage( ); } finally{ println(“Transaction failed…”); }

} }

import java.util.Scanner;

class CustomerAccount{

int accNo;

String custName;

double balance;

CustomerAccount(int accNo,String custName,double balance)

{

this.accNo = accNo;

this.custName = custName;

this.balance = balance;

}

void withdrawAmt(double Amt)

{

try{

if(balance-Amt <0)

throw new InsufficientBalException("Insufficient Balance");

balance-=Amt;

System.out.println("Balance in the account is "+ balance);

}

catch (InsufficientBalException e)

{

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

System.out.println(e);

System.out.println();

}

}

void depositAmt(double Amt)

{

try

{

if(Amt<0)

throw new InvalidAmtException("Invalid amount");

balance+=Amt;

System.out.println("Balance in the account is "+ balance);

}

catch( InvalidAmtException e)

{

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

System.out.println(e);

System.out.println();

}

}

}

class InsufficientBalException extends Exception{

InsufficientBalException(String msg)

{

super(msg);

}

public String toString()

{

return "Amount can't be withdrawm as there is insufficient Balance";

}

}

class InvalidAmtException extends Exception{

InvalidAmtException(String msg)

{

super(msg);

}

public String toString()

{

return "Can't deposit as amount is negative";

}

}

public class Main {

public static void main(String[] args) {

CustomerAccount c =new CustomerAccount(101,"Harry Styles",1000);

c.depositAmt(-500);

c.withdrawAmt(1200);

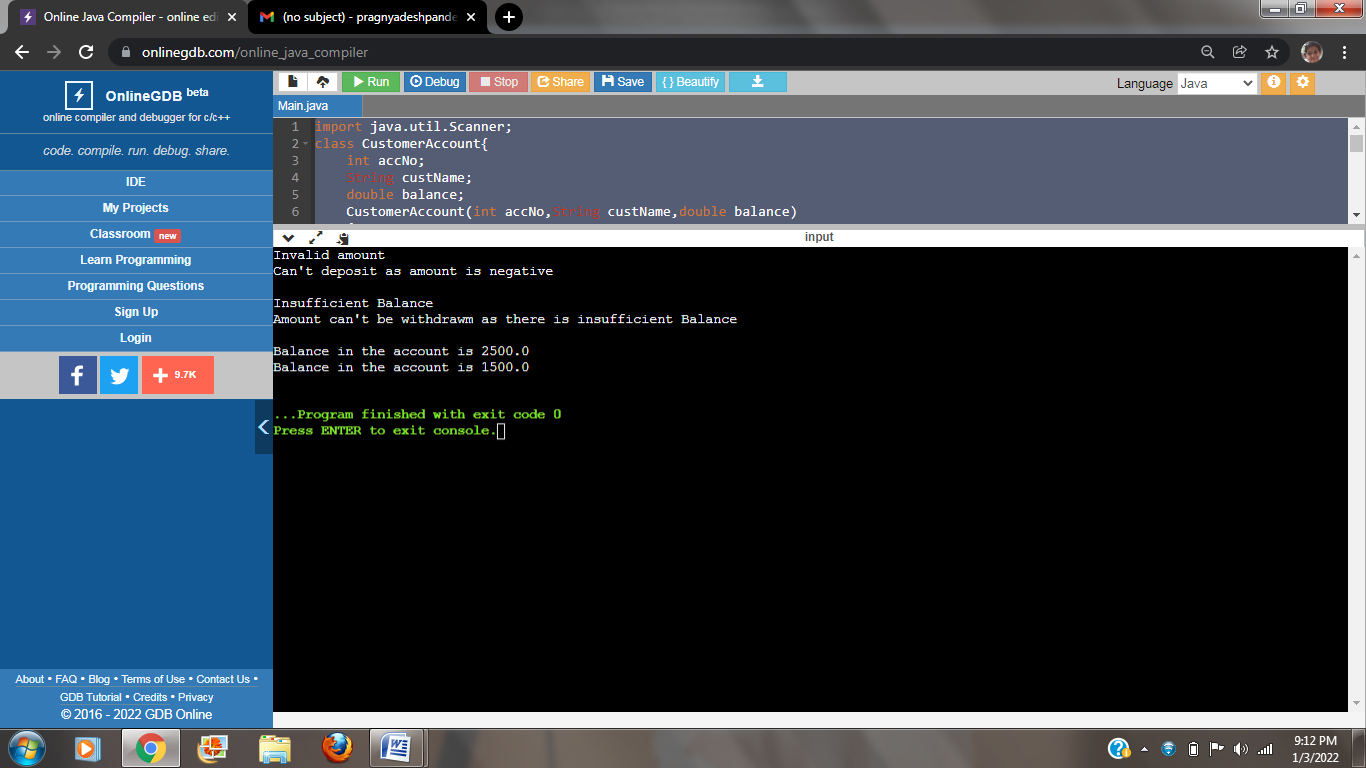
c.depositAmt(1500);

c.withdrawAmt(1000);

}

}

**OUTPUT:**

****

**PRAGNYA DESHPANDE**

**2GI20CS088**

**03/01/2022**

**TERMWORK 8.4:**

8.4).Write java program that takes the value of num variable and checks it is odd, then the throw keyword will raise the user defined exception and the catch block will get execute. OddNumberException class is derived from the Exception class. To implement user defined exception throw an exception object explicitly.

**CODE:**

**import java.util.Scanner;**

**class oddNumberException extends Exception**

**{**

**oddNumberException()**

**{**

**super(" Odd Number Exception ");**

**}**

**oddNumberException(String msg)**

**{**

**super(msg);**

**}**

**}**

**public class Main {**

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

**{**

**int num;**

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

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

**num=ob.nextInt();**

**try {**

**if (num%2==0)**

**System.out.println("Number is Even");**

**else**

**throw(new oddNumberException());**

**}**

**catch(oddNumberException e)**

**{**

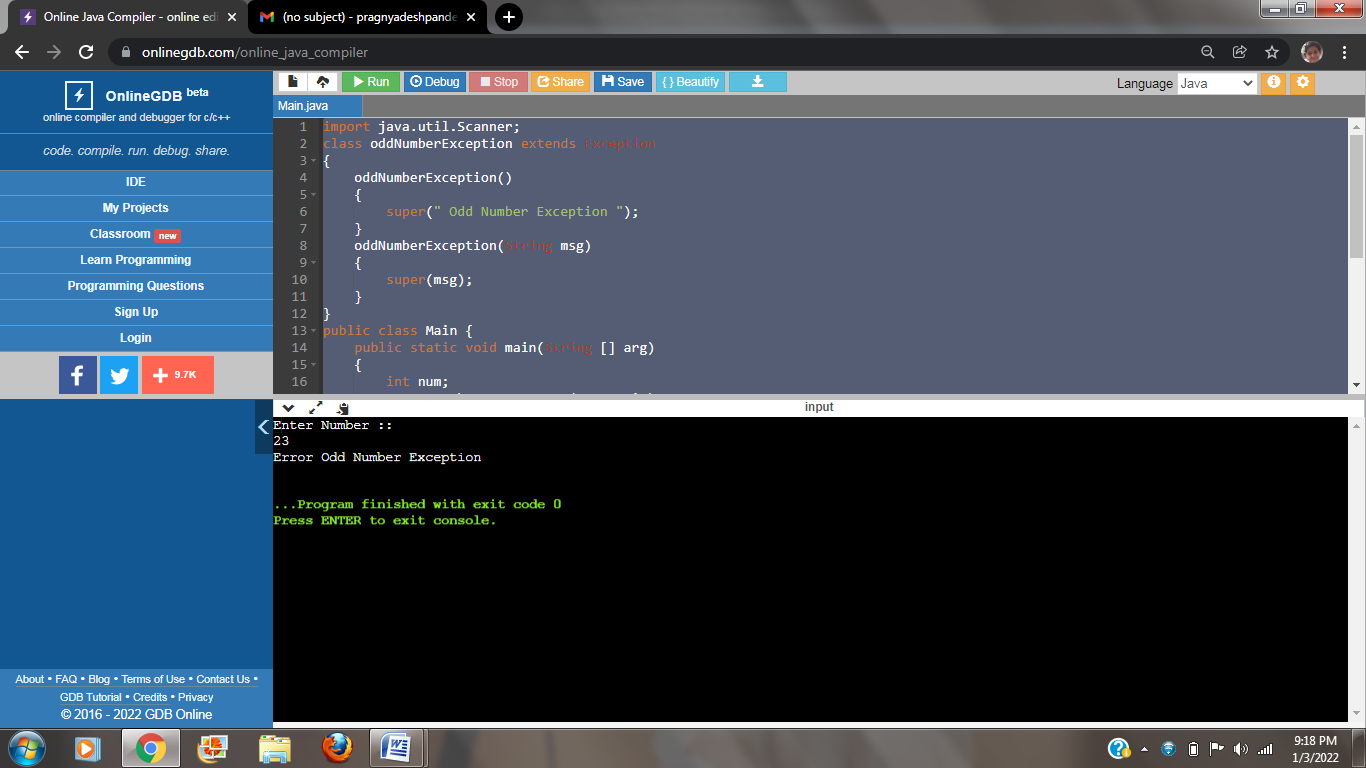
**System.out.println("Error"+e.getMessage());**

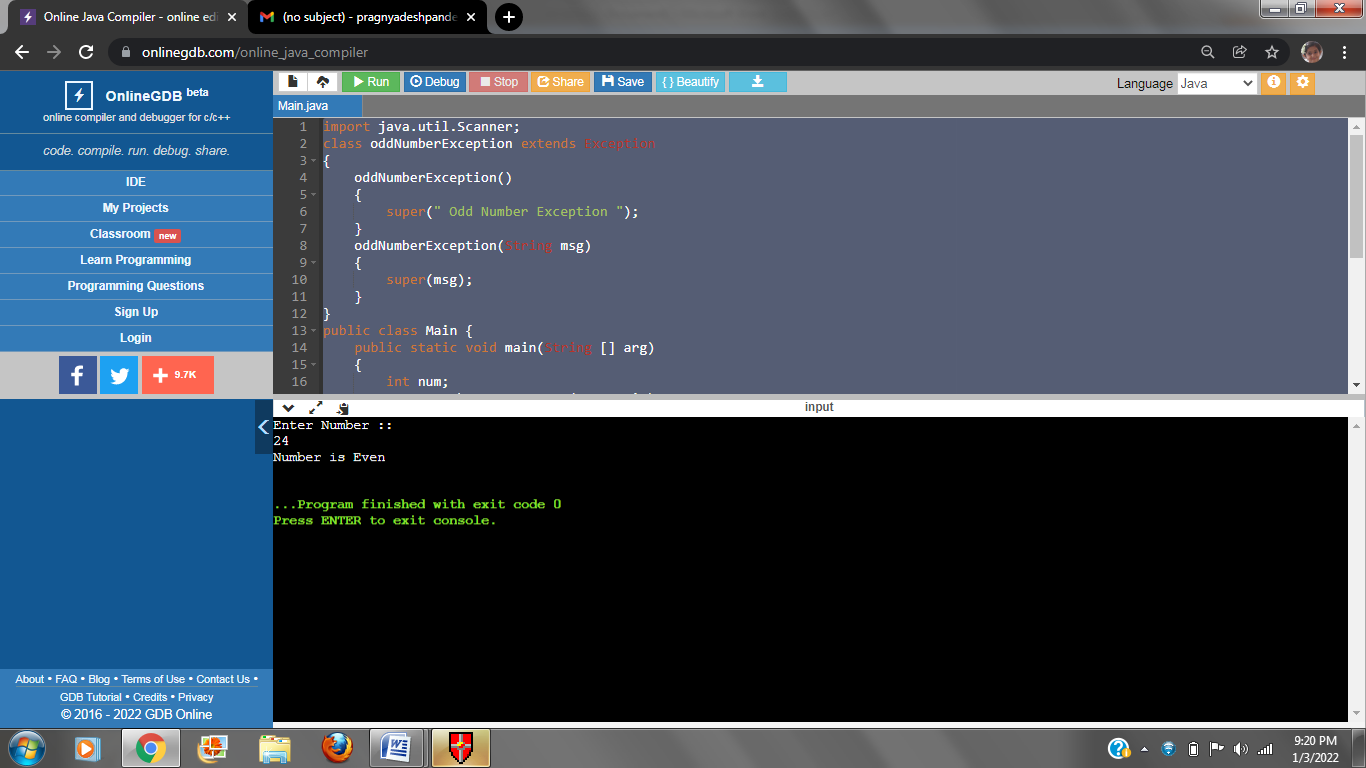
**}**

**}**

**}**

**OUTPUT:**

****

****