**D.Ujwala Akanksha**

**9919004069**

23/10/2020

1.Write a program that creates a user interface to perform integer divisions. The user

enters two numbers Num1 and Num2. If Num1 or Num2 is not an integer, the

program would throw NumberFormatException. If Num2 is Zero, the program would

throw an ArithmeticException. Display the exception.

Code :

import java.io.\*;

public class Main {

public static void main (String args[]) throws IOException {

String num1,num2;

int n1,n2,d;

BufferedReader br = new BufferedReader (new InputStreamReader(System.in));

try{

num1 = br.readLine();

num2 = br.readLine();

n1 = Integer.parseInt(num1);

n2 = Integer.parseInt(num2);

d = n1/n2;

System.out.println(" Output " + d);

}

catch(NumberFormatException e)

{

System.out.println("Inputs are not valid");

}

catch(ArithmeticException ae)

{

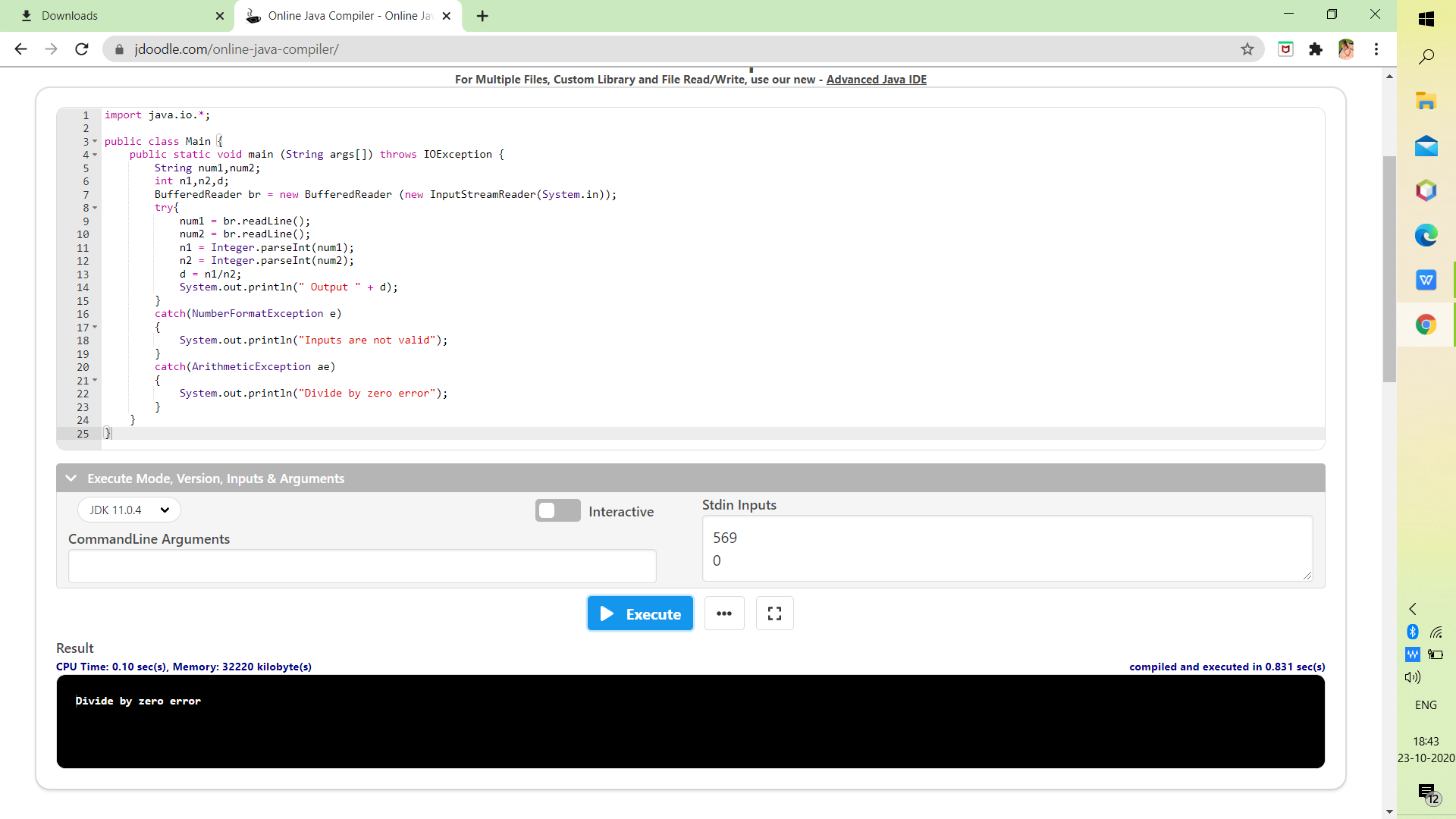
System.out.println("Divide by zero error");

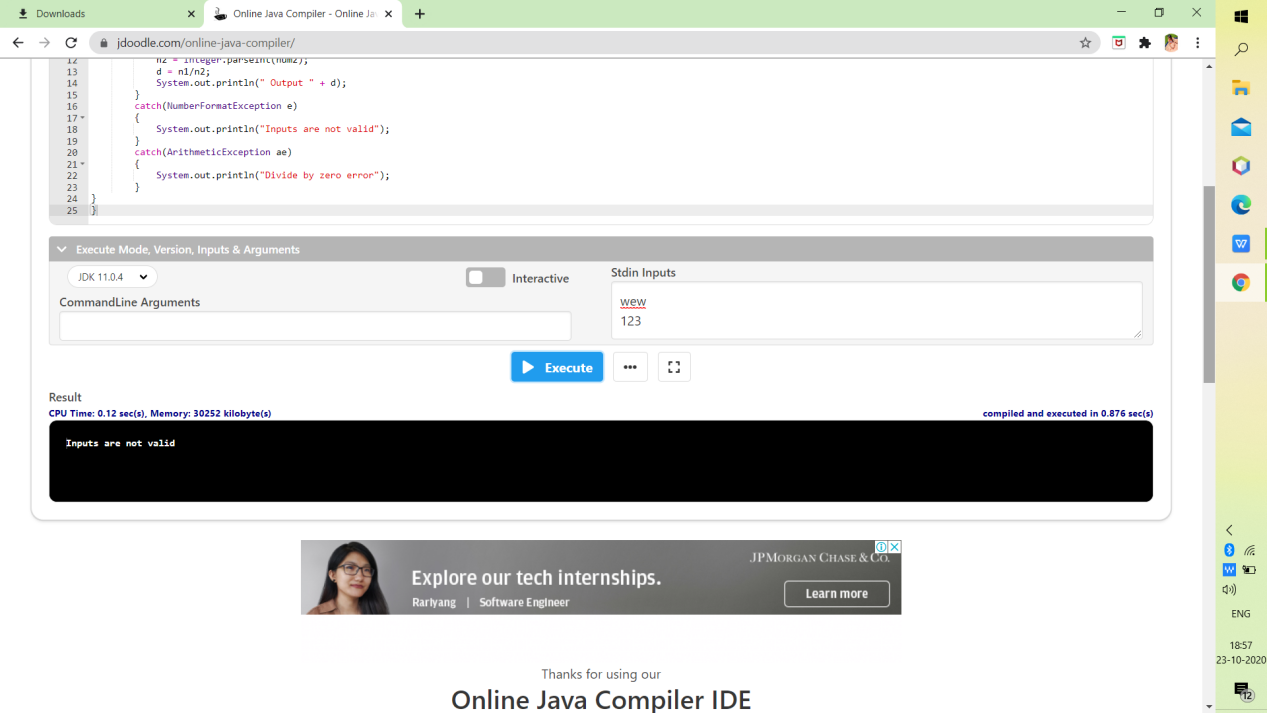
}

}

}

Out put :





2. Java programs to create an bank account with minimum balance, deposit amount,

withdraw amount and throws LessBalanceException, create a

LessBalanceException class which returns a statement says withdraw amount is

not valid, creates 2 accounts and try to withdraw more money than account and

see which type of exception occurs.

Code :

class BalanceCheck extends Exception

{

BalanceCheck()

{

super("Transaction failed less balance ");

}

}

class Account

{

int accno;

String name;

double balance;

static int minimum=500;

Account(int ano,String n,double bal)

{

accno=ano;name=n;balance=bal;

}

void withdrawl(int amt)throws BalanceCheck

{

if((balance-amt)>minimum)

{

balance-=amt;

System.out.println("transaction was succesfully done");

}

else

{

throw new BalanceCheck();

}

}

void deposit(int amt)

{

balance+=amt;

}

double checkBalance()

{

return balance;

}

}

public class MyClass {

public static void main(String args[]) {

Account a1=new Account(4069,"akki ",60000);

Account a2=new Account(4163,"roshini ",50000);

try

{

a1.withdrawl(16000);

a2.withdrawl(2000);

}

catch(BalanceCheck b)

{

System.out.println(b);

}

System.out.println("a1 balance ="+a1.checkBalance());

System.out.println("a2 balance ="+a2.checkBalance());

a1.deposit(9000);a2.deposit(9000);

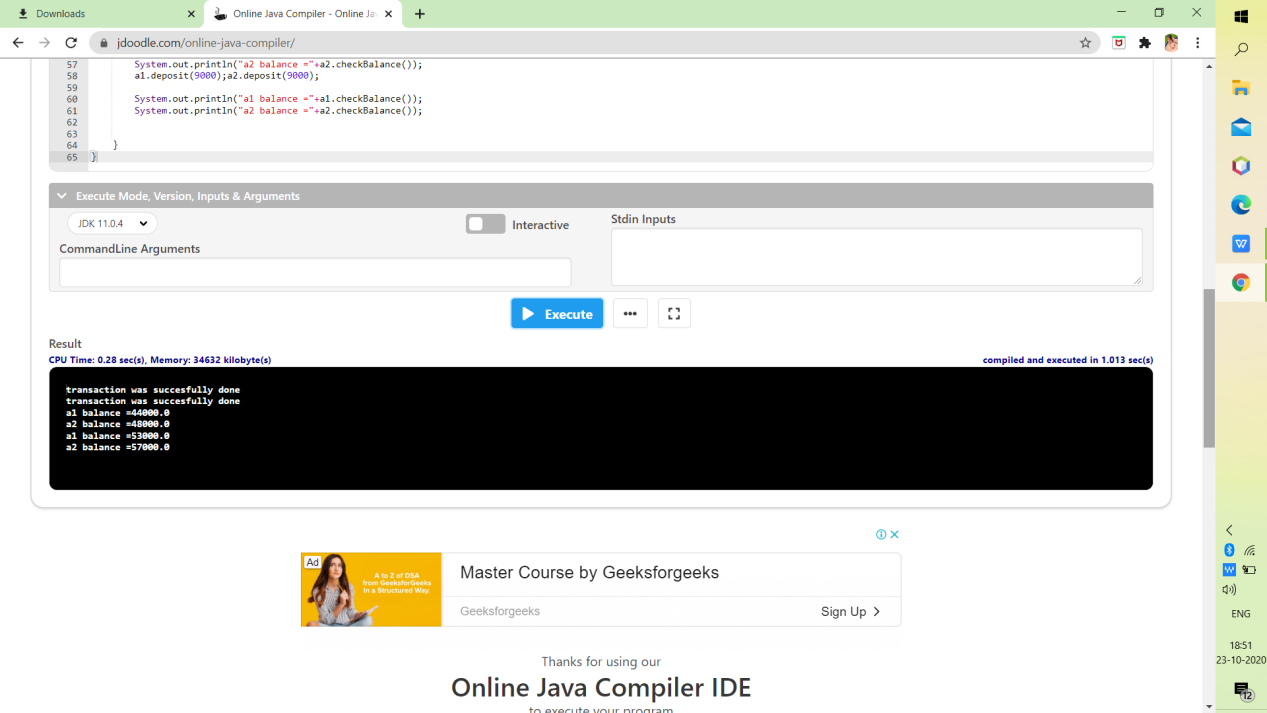
System.out.println("a1 balance ="+a1.checkBalance());

System.out.println("a2 balance ="+a2.checkBalance());

}

}

Out put :



3. Write a Java program to check whether the age entered is a valid number by?

Code :

import java.util.Scanner;

class AgeCheck extends Exception

{

AgeCheck()

{

super("Exception : invalid Age");

}

}

public class MyClass{

public static void main(String args[]) {

int age;

Scanner s = new Scanner(System.in);

age = s.nextInt();

boolean b;

try{

b= CheckAge (age);

System.out.println("valid");

}

catch (AgeCheck ag)

{

System.out.println(ag);

}

}

static boolean CheckAge(int age) throws AgeCheck

{

if(age > 0 && age<=120)

return true;

else

{

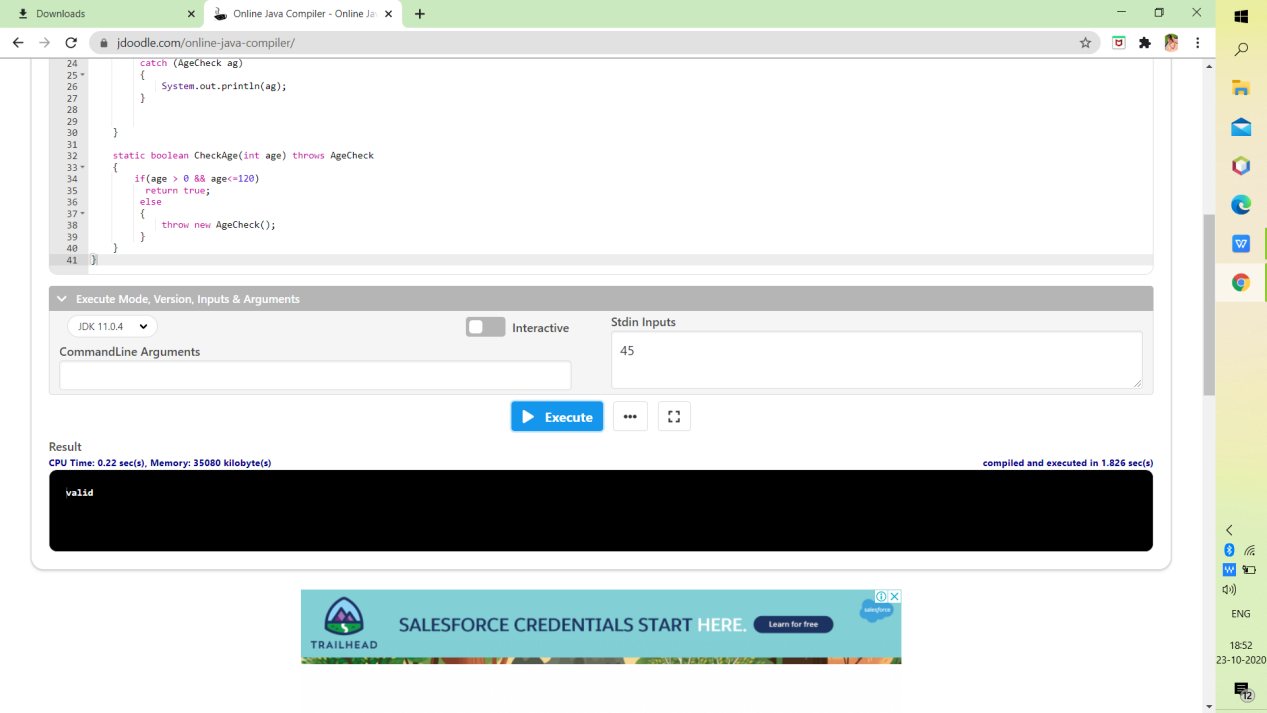
throw new AgeCheck();

}

}

}

Out put :



4. Design a Java interface for ADT Stack. Implement this interface using array.

Provide necessary exception handling in both the implementations.?

Code :

class FullStack extends Exception

{

FullStack()

{

super("Stack is Full");

}

}

class EmptyStack extends Exception

{

EmptyStack()

{

super("Stack is Empty ");

}

}

class Stack

{

int top;

int arr[];

static int max=10;

Stack()

{

top=-1;

arr=new int[max];

}

void push(int x)throws FullStack

{

if(top==max-1)

{

throw new FullStack();

}

else

{

arr[++top]=x;

}

}

int pop()throws EmptyStack

{

if(top==-1)

{

throw new EmptyStack();

}

else

{

return(arr[top--]);

}

}

void print()

{

for(int i=0;i<arr.length;i++)

System.out.print(arr[i]+" ");

System.out.println();

}

}

public class MyClass {

public static void main(String args[]) {

Stack s1=new Stack();int x;

for(int i=1;i<=12;i++)

{

try{

s1.push(i);

s1.print();

}

catch(FullStack fs)

{

System.out.println(fs);

}

}

for (int i=1;i<=12;i++)

{

try

{

x=s1.pop();

System.out.print(x +" ");

}

catch(EmptyStack es)

{

System.out.println(es);

}

}

}

}

Out put :

