import java.util.\*;

import java.lang.Exception;

class WrongAge extends Exception{

public String toString(){

return "Wrong age! Please enter the Right age";}

}

class Father{

int age;

Father(int agef){

age=agef;

System.out.println("Father age:"+age);

}

}

class Son extends Father{

int age;

Son(int agef,int ages){

super(agef);

age=ages;

System.out.println("Son's age:"+age);

}

}

class Exception1{

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

Scanner sc=new Scanner(System.in);

System.out.println("Enter the Father's Age");

int agef=sc.nextInt();

System.out.println("Enter the Son's Age");

int ages=sc.nextInt();

if(agef<0||ages>=agef){

throw new WrongAge();

}

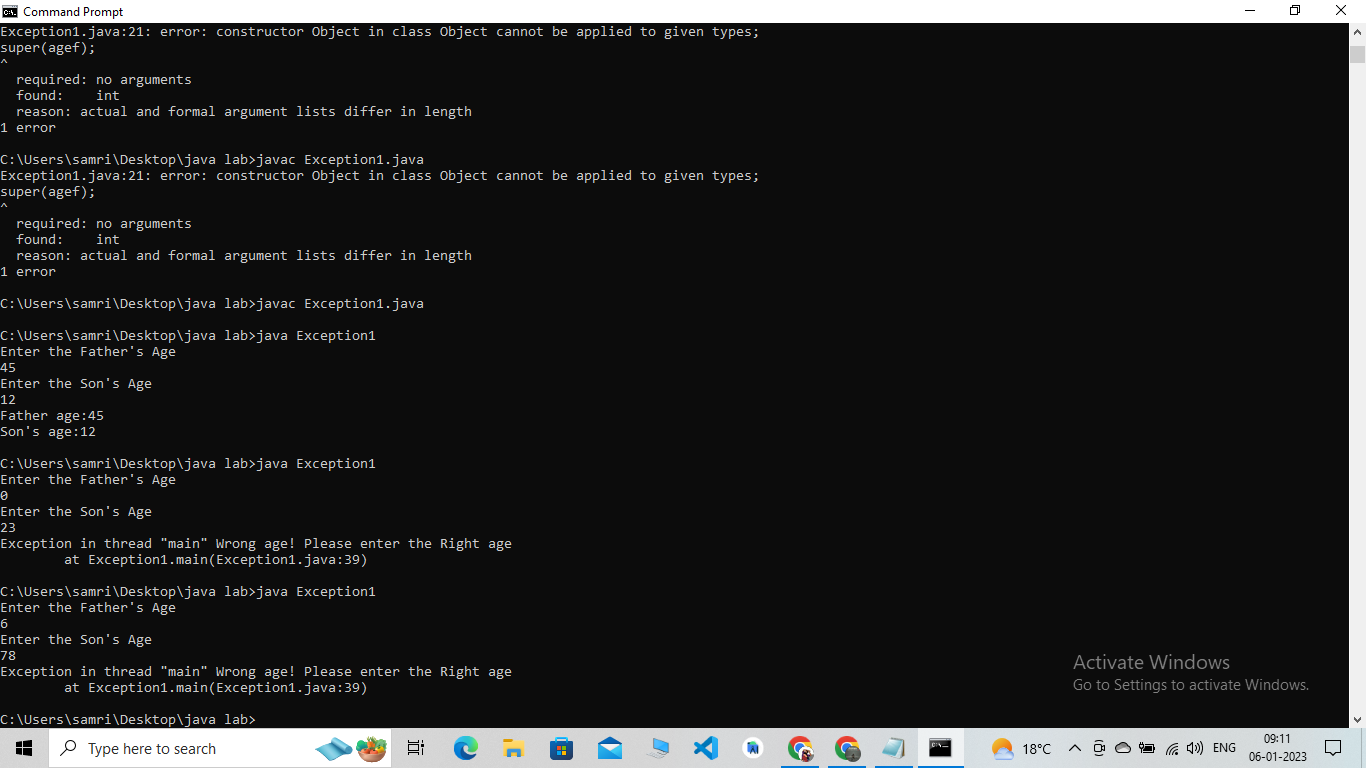
else{

Son s=new Son(agef,ages);

}

}

}



import java.util.\*;

import java.lang.Exception;

class MyFactorialException extends Exception{

public String toString(){

return "Error,It can't be greater than 15!!";}

}

class factorial{

public int ComputeFactorial(int n) throws MyFactorialException{

int fact=1;

if(n>15){

throw new MyFactorialException();

}

else{

for(int i=n;i>0;i--){

fact\*=i;

}

}

return fact;

}

}

class Exception2{

public static void main(String args[]){

factorial f=new factorial();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the Number to find the factorial");

int n=sc.nextInt();

try{

System.out.println("The factorial of "+n+" is "+f.ComputeFactorial(n));

}

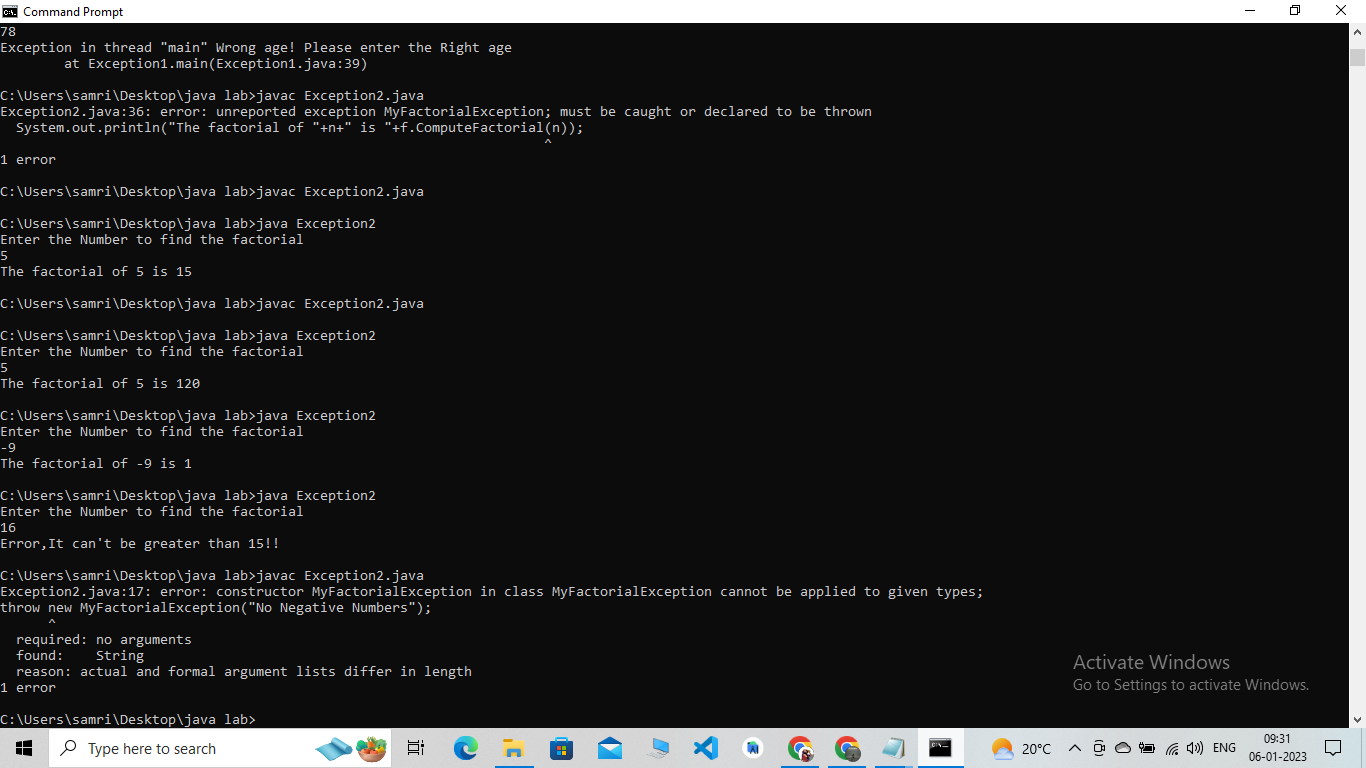
catch(Exception e){

System.out.println(e);

}

}

}



import java.util.\*;

import java.lang.Exception;

class MyaccountException extends Exception{

public String toString(){

return "Balance is Less,please update it and withdraw";}

}

class account{

Scanner sc=new Scanner(System.in);

String name;

double bal;

account(){

System.out.println("Enter Your Name");

name=sc.next();

System.out.println("Enter the intial amount");

bal=sc.nextDouble();

System.out.println("Account Created Succesfully");

}

public void withdraw() throws MyaccountException{

System.out.println("Enter the amount to widthdraw");

double amt=sc.nextDouble();

if(amt>bal){

throw new MyaccountException();

}

bal=bal-amt;

System.out.println("Amount Withdrawn "+amt);

System.out.println("Balance "+bal);

}

public void deposit() {

System.out.println("Enter the amount to deposit");

double amt=sc.nextDouble();

bal=bal+amt;

System.out.println("Amount deposited "+amt);

System.out.println("Balance "+bal);

}

}

class AccountExc{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

account a=new account();

System.out.println("1.Deposit");

System.out.println("2.withdraw");

int choice=sc.nextInt();

switch(choice){

case 2:

try{

a.withdraw();}

catch(Exception e){

System.out.println(e);

}

break;

case 1:a.deposit();

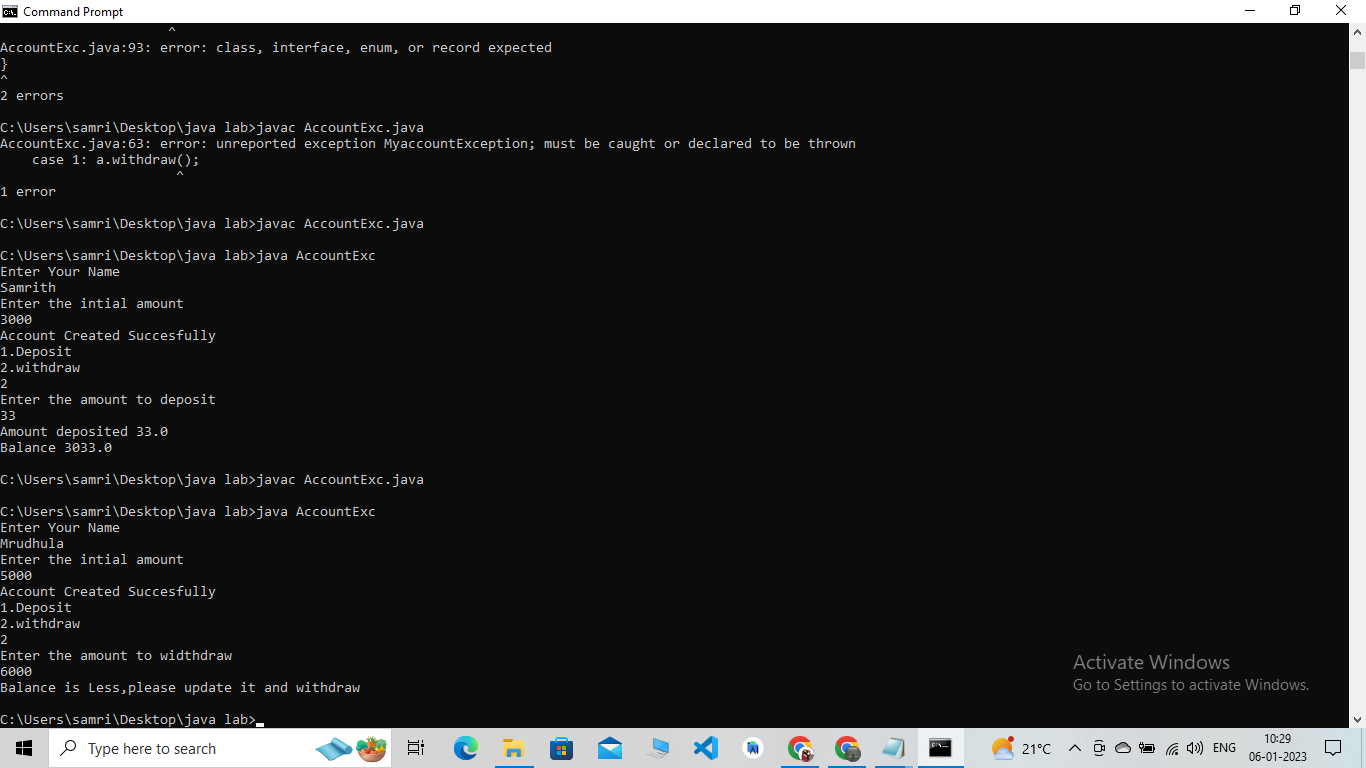
break;

default:System.out.println("Enter the corrct choice");

}

}

}



import java.util.\*;

interface stackinterface

{

void push(int item);

int pop();

}

class Stack implements stackinterface

{

Scanner sc=new Scanner(System.in);

private int stack[ ];

private int top;

Stack(int size)

{

stack=new int[size];

top=-1;

}

public void push(int item)

{

if(top==stack.length-1)

{

System.out.println("Stack Overflows");

}

else{

stack[++top]=item;

}

}

public int pop()

{

if(top<0)

{

System.out.println("Stack Underflows");

return 0;

}

else

return stack[top--];

}

}

class StackOperations

{

public static void main(String args[ ])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the Size of the Stack");

int n=sc.nextInt();

Stack s=new Stack(n);

for(int i=0;i<n;i++){

System.out.println("Enter the value "+(i+1));

int val=sc.nextInt();

s.push(val);

}

System.out.println(" Stack Contents are.");

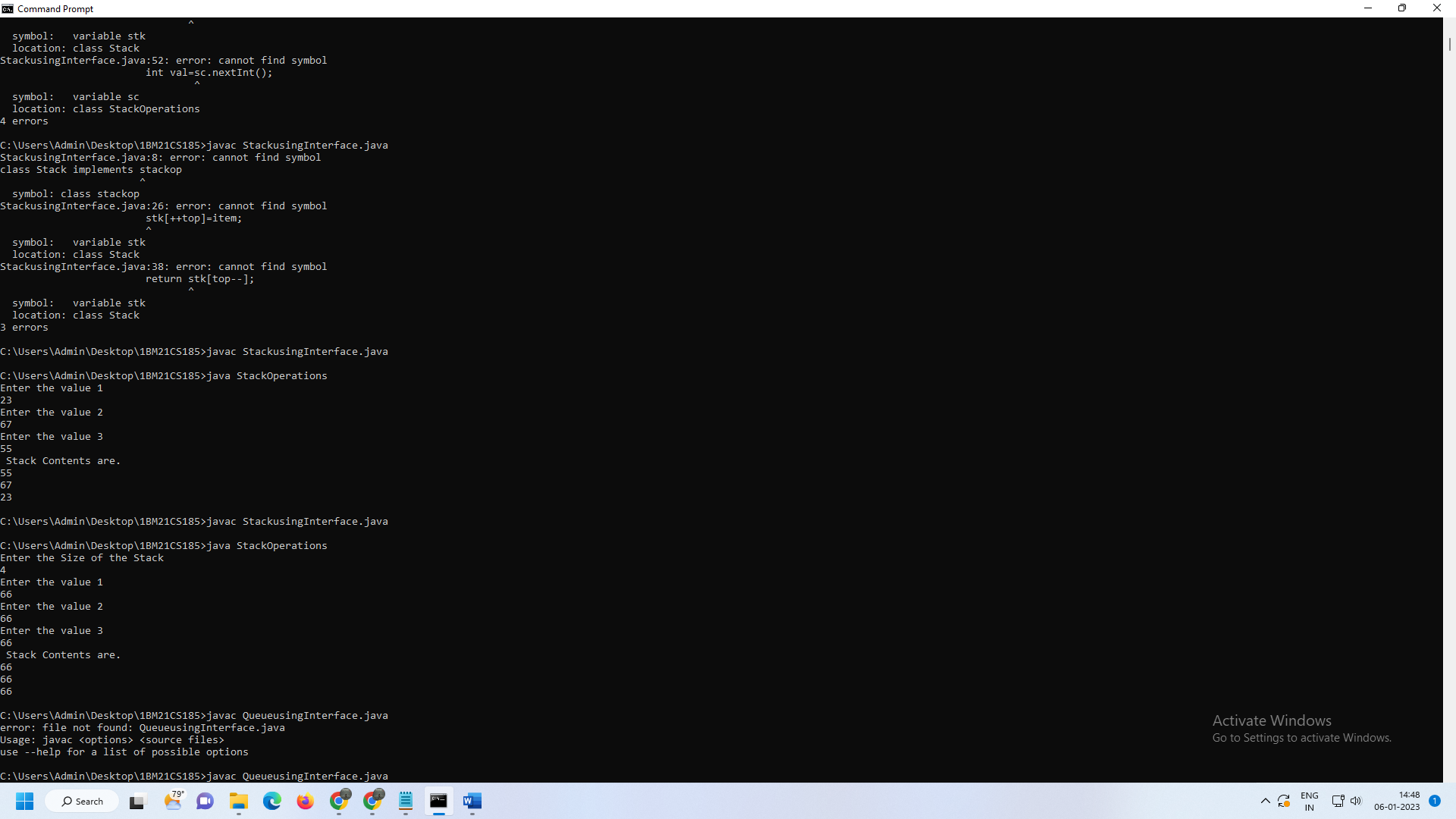
for(int i=0;i<n;i++){

System.out.println(s.pop());

}

}

}



import java.util.\*;

interface queueinterface

{

void enqueue(int item);

int dequeue();

}

class Queue implements queueinterface

{

Scanner sc=new Scanner(System.in);

private int queue[ ];

private int front;

private int rear;

Queue(int size)

{

queue=new int[size];

front=-1;

rear=-1;

}

public void enqueue(int item)

{

if(rear==queue.length-1)

{

System.out.println("Queue Overflows");

}

else{

queue[++rear]=item;

}

}

public int dequeue()

{

if((front > rear))

{

System.out.println("Queue Underflows");

return 0;

}

else

return queue[++front];

}

}

class QueueOperations

{

public static void main(String args[ ])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the Size of the Queue");

int n=sc.nextInt();

Queue q=new Queue(n);

for(int i=0;i<n;i++){

System.out.println("Enter the value "+(i+1));

int val=sc.nextInt();

q.enqueue(val);

}

System.out.println(" Stack Contents are.");

for(int i=0;i<n;i++){

System.out.println( q.dequeue());

}

}

}

