**NAME = K LEENA**

**REG NO = 9919004159**

OUESTION 1

import java.io.\*;

public class MyClass {

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("input are not valid");

}

catch (ArithmeticException ae)

{

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

}

}

}

INPUT

123

WSE

OUTPUT

input are not valid

QUESTION 2

import java.io.\*;

class BalanceCheck extends Exception{

BalanceCheck(){

super("Transaction Denied: No min balance found");

}

}

class Bank{

int accountno;

String name;

double balance;

static int min\_amount=500;

Bank(int ano,String nm,double bal,int min){

accountno=ano;

name=nm;

balance=bal;

}

void Withdraw(double cash) throws BalanceCheck{

if((balance-cash)>=min\_amount){

balance=balance-cash;

System.out.println("Transaction Succesful");

System.out.println("the balance after withdrawl is "+ balance);

}

else{

throw new BalanceCheck();

}

}

void Deposit(double cash){

balance = balance+cash;

System.out.println("Transaction Succesful");

System.out.println("the balance after deposit is "+balance);

}

void CheckBal(){

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

}

}

public class Main

{

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

Bank b1 = new Bank(4159,"leena",25000,500);

Bank b2 = new Bank(4163,"shalu",20000,500);

try{

b1.Withdraw(4300);

b2.Withdraw(7000);

}

catch(BalanceCheck b){

System.out.println(b);

}

b1.CheckBal();

b2.CheckBal();

b1.Deposit(6000);

b2.Deposit(9000);

b1.CheckBal();

b2.CheckBal();

}

}

OUTPUT

Transaction Succesful

the balance after withdrawl is 20700.0

Transaction Succesful

the balance after withdrawl is 13000.0

the balance is20700.0

the balance is13000.0

Transaction Succesful

the balance after deposit is 26700.0

Transaction Succesful

the balance after deposit is 22000.0

the balance is26700.0

the balance is22000.0

question 3

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();

}

}

}

Input

34

Output

Valid

Question 4

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);

}

}

}

}

Output

1 0 0 0 0 0 0 0 0 0

1 2 0 0 0 0 0 0 0 0

1 2 3 0 0 0 0 0 0 0

1 2 3 4 0 0 0 0 0 0

1 2 3 4 5 0 0 0 0 0

1 2 3 4 5 6 0 0 0 0

1 2 3 4 5 6 7 0 0 0

1 2 3 4 5 6 7 8 0 0

1 2 3 4 5 6 7 8 9 0

1 2 3 4 5 6 7 8 9 10

FullStack: Stack is Full

FullStack: Stack is Full

10 9 8 7 6 5 4 3 2 1 EmptyStack: Stack is Empty

EmptyStack: Stack is Empty