CSE18R272-LAB MANUAL

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

COMPUTER SCIENCE AND EDUCATION

Date : 23-10-2020

Name : Jadapalli Karthik Kumar

Regno : 9919004113

Course Name : Java Programming

Course Code : CSE18R272

Section : A5

***PROGRAM 1:***

import java.io.\*;

public class Main

{

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

int n1,n2,d;

String num1,num2;

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 e){

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

}

}

}

***OUTPUT:***

***20***

***0***

***Divide by Zero error***

***PROGRAM 2:***

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(123,"micheal",25000);

Account a2=new Account(123,"jackson",30000);

try

{

a1.withdrawl(16800);

a2.withdrawl(12000);

}

catch(BalanceCheck b)

{

System.out.println(b);

}

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

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

a1.deposit(10000);a2.deposit(5000);

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

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

}

}

***OUTPUT:***

***transaction was succesfully done***

***transaction was succesfully done***

***a1 balance =8200.0***

***a2 balance =18000.0***

***a1 balance =18200.0***

***a2 balance =23000.0***

***PROGRAM 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();

}

}

}

***OUTPUT:***

***23***

***Valid***

***PROGRAM 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***