**Smt. Chandaben Mohanbhai Patel Institute of Computer Applications**

**Sub: CA314: Object Oriented Programming through JAVA**

**Practical Assignment-8**

**Unit-V**

**Exception Handling**

**Date: 12.10.2021**

ID : 19BCA063

/\*

1. Demonstrate the use of throw & throws keyword in exception handling.

\*/

package bca;

class UseofThrow{

static void fun()

{

try

{

throw new NullPointerException("demo");

}

catch(NullPointerException e)

{

System.out.println("Caught inside fun().");

throw e;

}

}

}

public class Assignment8\_1 {

public static void main(String[] args) {

UseofThrow obj=new UseofThrow();

try

{

obj.fun();

}

catch(NullPointerException e)

{

System.out.println("Caught in main.");

}

}

}

/\*

2. Explain the use of finally block in exception handling.

\*/

package bca;

public class Assignment8\_2 {

public static void main(String[] args)

{

try {

System.out.println("inside try block");

System.out.println(34 / 0);

}

catch (ArithmeticException e) {

System.out.println(e);

}

finally {

System.out.println(

"finally : i execute always.");

}

}

}

/\*

3. Write a java program to accept name from user.

If the name contains a non-alphabetical character, raise an Exception “InvalidNameException.

\*/

package bca;

import java.util.Scanner;

import java.io.\*;

class NameNotValidException extends Exception

{

public String validname()

{

return("Name is not Valid … Please ReEnter the Name");

}

}

class Student

{

String name;

Student()

{

name=null;

}

Student(String n)

{

int l,temp=0;

l=n.length();

for(int i=0;i<l;i++)

{

char ch;

ch=n.charAt(i);

if(ch<'A' || ch>'Z' && ch<'a' || ch>'z')

temp=1;

}

try

{

if(temp==1)

throw new NameNotValidException();

else

name=n;

}

catch(NameNotValidException e2)

{

System.out.println(e2);

}

}

void display()

{

System.out.print("Name: ");

System.out.println(name);

}

}

public class Assignment8\_3 {

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

{

Scanner scan=new Scanner(System.in);

String n;

System.out.println("Enter Your name: ");

n=scan.nextLine();

Student s=new Student(n);

s.display();

}

}

/\*

4. Write a class Driver with attributes vehicle no, name & age. Initialize values through parameterized constructor.

If age of driver is less than 18 then generate user-defined exception “AgeNotWithinTheRange”.

\*/

package bca;

import java.io.\*;

import java.util.Scanner;

class AgeNotWithInRangeException extends Exception

{

public String toString()

{

return("Age is not In Range: ");

}

}

class Driver{

int age;

String vehicleno,Name;

Driver()

{

vehicleno=null;

age=0;

Name=null;

}

Driver(String no,int age,String n)

{

vehicleno=no;

Name=n;

try

{

if(age>18)

this.age=age;

else

throw new AgeNotWithInRangeException();

}

catch(AgeNotWithInRangeException e1)

{

System.out.println(e1);

}

}

void display()

{

System.out.print("\nVehicle No: ");

System.out.println(vehicleno);

System.out.print("Age: ");

System.out.println(age);

System.out.print("Name: ");

System.out.println(Name);

}

}

public class Assignment8\_4 {

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

Scanner Scan=new Scanner(System.in);

String vehicle,Name;

int Age;

System.out.println("Enter Your Vehicle Number: ");

vehicle=Scan.nextLine();

System.out.println("Enter Name of Driver: ");

Name=Scan.nextLine();

System.out.println("Enter Age of Driver: ");

Age=Integer.parseInt(Scan.nextLine());

Driver obj\_D=new Driver(vehicle,Age,Name);

obj\_D.display();

}

}

/\*

5. Write a program to demonstrate use of user-defined exception,

the CheckingAccount class contains a withdraw() method that throws an InsufficientFundsException when fund is less than amount.

\*/

package bca;

import java.util.Scanner;

class InSufficientFundException extends RuntimeException {

private String message;

public InSufficientFundException(String message) {

this.message = message;

}

public InSufficientFundException(Throwable cause, String message) {

super(cause);

this.message = message;

}

public String getMessage() {

return message;

}

}

class CheckingAmount {

private int balance = 5000;

public int balance() {

return balance;

}

public void withdraw(int amount) throws InSufficientFundException {

if (amount > balance) {

throw new InSufficientFundException(String.format(

"Current balance %d is less than requested amount %d",

balance, amount));

}

balance = balance - amount;

}

}

public class Assignment8\_5 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

CheckingAmount obj=new CheckingAmount();

System.out.println("Current balance : " + obj.balance());

System.out.println("Enter Amount to Withdraw: ");

int amount=Integer.parseInt(sc.nextLine());

obj.withdraw(amount);

System.out.println("New Current balance is: " + obj.balance());

}

}

/\*

6. Define an exception called “NoEqualException” that is thrown when a float value is not equal to 3.14.

Write a program that uses the above user defined exception.

\*/

package bca;

import java.io.\*;

import java.util.Scanner;

class NoEqualException extends Exception{

public String toString()

{

return("Both The Float values Are Not Equal.");

}

}

public class Assignment8\_6 {

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

double f,t=3.14;

Scanner scan=new Scanner(System.in);

System.out.println("Enter The Vale of Float: ");

f=Double.parseDouble(scan.nextLine());

try{

if (f==t)

System.out.println("Values Are Same...");

else

throw new NoEqualException();

}

catch(NoEqualException ex){

System.out.println(ex);

}

}

}

/\*

7. Write a class Student with attributes roll no, name, age and course.

Initialize values through parameterized constructor.

If age of student is not in between 15 and 21 then generate user-defined exception “Age Not Within The Range”.

If name contains numbers or special symbols raise exception “Name not valid”.

\*/

package bca;

import java.io.\*;

import java.util.Scanner;

class AgeNotWithInRangeException2 extends Exception

{

public String toString()

{

return("Age is not In Range: ");

}

}

class NameNotValidException2 extends Exception

{

public String validname()

{

return("Name is not Valid … Please ReEnter the Name");

}

}

class Person

{

String name;

int age;

Person(String n,int age)

{

int l,temp=0;

l=n.length();

for(int i=0;i<l;i++)

{

char ch;

ch=n.charAt(i);

if(ch<'A' || ch>'Z' && ch<'a' || ch>'z')

temp=1;

}

try

{

if(temp==1)

throw new NameNotValidException2();

else

name=n;

}

catch(NameNotValidException2 e2)

{

System.out.println(e2);

}

try

{

if(age>=15 && age<=21)

this.age=age;

else

throw new AgeNotWithInRangeException2();

}

catch(AgeNotWithInRangeException2 e1)

{

System.out.println(e1);

}

}

void display()

{

System.out.print("Name: ");

System.out.println(name);

System.out.print("Age: ");

System.out.println(age);

}

}

public class Assignment8\_7 {

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

Scanner scan=new Scanner(System.in);

String n;

int age;

System.out.println("Enter Your name: ");

n=scan.nextLine();

System.out.println("Enter Age: ");

age=Integer.parseInt(scan.nextLine());

Person p=new Person(n,age);

p.display();

}

}

/\*

8. Write a java program to accept a number from the user,

if number is zero then throw user defined exception “Number is 0”

otherwise check whether no is prime or not (Use static keyword).

\*/

package bca;

import java.io.\*;

import java.util.Scanner;

class NumberZeroException extends Exception

{

public String toString()

{

return("Number is 0");

}

}

class PrimeNumber

{

int a;

Scanner scan=new Scanner(System.in);

PrimeNumber()

{

try

{

System.out.println("Enter any integer to check prime ");

a=Integer.parseInt(scan.nextLine());

if(a==0)

throw new NumberZeroException();

}

catch(NumberZeroException ex)

{

System.out.println(ex);

}

}

public void prime()

{

int cnt=0;

for(int i=2;i<=a/2;i++)

if(a%i==0)

{

cnt++;

break;

}

if(cnt==0)

System.out.println(a+" Number is prime");

else

System.out.println(a+" Number is not prime");

}

}

public class Assignment8\_8 {

public static void main(String args[])

{

PrimeNumber pn=new PrimeNumber();

pn.prime();

}

}

/\*

9. Write program to accept rollno, marks of four subjects in an array from user and throw MarksOutOf BoundsException

if marks are<0 or marks> 100. Also check ArrayIndexOutofBoundsException.

\*/

package bca;

import java.io.\*;

import java.util.Scanner;

class MarksOutOfBoundException extends Exception

{

public String toString()

{

return "Enter marks in between 1 & 100";

}

}

class Student1

{

int rno;

float marks[];

Scanner scan=new Scanner(System.in);

Student1()

{

rno=0;

marks=new float[4];

for(int i=0;i<4;i++)

marks[i]=0;

}

public void accept()

{

System.out.println("Enter roll no: ");

try

{

rno=Integer.parseInt(scan.nextLine());

for(int i=0;i<4;i++)

{

System.out.println("Enter marks of subject "+(i+1));

marks[i]=Float.parseFloat(scan.nextLine());

if(marks[i]<0 || marks[i]>100)

throw new MarksOutOfBoundException();

}

}

catch(MarksOutOfBoundException me)

{

System.out.println(me);

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("Array index out of bounds");

}

}

public void display()

{

System.out.println("Roll no\tMarks1\tMarks2\tMarks3\tMarks4");

System.out.print(rno+"\t");

for(int i=0;i<4;i++)

System.out.print(marks[i]+"\t");

System.out.println("\n");

}

}

public class Assignment8\_9 {

public static void main(String args[])

{

Student1 s1=new Student1();

s1.accept();

s1.display();

}

}