**Department of Computer Science & Engineering, SDMCET, Dharwad-2**



**AOOP Assignment Submission Report**

**[Submitted as part of CTA Assignment No-1]**

|  |  |  |  |
| --- | --- | --- | --- |
| Course: | Advanced Object-Oriented Programming | Course Code: | 18UCSE508 |
| Semester: | V | Division: | A |

Submitted by:

|  |  |  |  |
| --- | --- | --- | --- |
| USN: | 2SD20CS090 | Name: | Sahana M Bevoor |

**1. Problem Definition:**

1.Write a java program to generate and handle any three built-in exceptions and display appropriate error messages.

PROGRAM:

public class Q1{

public static void main(String[] args){

int a=10;

int b=5;

int c=5;

String s=null;

int d[]=new int[5];

try{

System.out.println(a/(b-c));

}

catch(ArithmeticException ae){

System.out.println("division by zero error"+ae);

}

try{

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

}

catch(NullPointerException ne){

System.out.println("String is null"+ne);

}

try{

d[10]=50;

}

catch(ArrayIndexOutOfBoundsException aoe){

System.out.println("array index exceeded"+aoe);

}

}

}

OUTPUT:



2.Write a java program to read an integer and check whether the number is prime or not. If negative number is entered throw an exception **NegativeNumberNotAllowedException** and if entered number is not prime, then throw **NumberNotPrimeException**.

PROGRAM:

import java.util.Scanner;

public class Q2{

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

Scanner sc=new Scanner(System.in);

System.out.println("Enter an integer");

int n=sc.nextInt();

if(n<0){

throw new numException("NegativeNumberNotAllowedException");

}

for(int i=2;i<n;i++){

if(n%i==0)

throw new numException("NumberNotPrimeException");

}

System.out.println("It is Prime number");

}

}

class numException extends Exception{

String msg;

public numException(String msg){

this.msg=msg;

}

public String toString(){

return "Exception: "+msg;

}

}

OUTPUT:

3.Write a java program to perform the following operations:

a) Read a line of text.

b) Search for substring SDMCET (case insensitive search)

c) If found then print success message.

d) Otherwise throw an exception **SubStringNotAllowedException.**

PROGRAM:

import java.util.\*;

import java.util.Scanner;

/\*\*

\* Q3

\*/

public class Q3 {

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

Scanner sc= new Scanner(System.in);

System.out.print("Enter the String =");

String testString = sc.nextLine();

testString = testString.toUpperCase();

String subString="SDMCET";

int i=0,j=0;

while(i<testString.length()){

if(testString.charAt(i)==subString.charAt(j) && j<subString.length()-1){

i++;

j++;

}else{

i++;

}

} //end of while

if(j == subString.length()-1){

System.out.println("Substring is present");

}else{

throw new SubstringNotFoundException("Substring is not found !! please enter the valid input");

}

}

}

class SubstringNotFoundException extends Exception{

String str;

SubstringNotFoundException(String str){

this.str = str;

}

public String toString() {

return this.str;

}

}

4.Write a java program to perform the following operations:

a) Create a file named **Aphabets.txt** and insert appropriate data into it.

b) Read the and copy all the consonants into another file named **Consonants.txt**

c) If vowels is encountered, throw an exception **VowelNotAllowedException** and continue until end of life.

PROGRAM:

import java.util.Scanner;

import java.io.\*;

public class Q4 {

public static void main(String[] args) {

try{

FileWriter w = new FileWriter("Alphabets.txt");

Scanner sc = new Scanner(System.in);

System.out.print("Enter the data to write in the file :");

String str = sc.nextLine();

w.write(str);

w.close();

File file = new File("Alphabets.txt");

Scanner reader = new Scanner(file);

StringBuilder s = new StringBuilder();

FileWriter write = new FileWriter("Consonate.txt");

while(reader.hasNext()){

String data = reader.next();

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

if(isVowel(data.charAt(i))){

System.out.println("vowel found " + data.charAt(i));

}else{

s.append(data.charAt(i));

}

}

write.write(s.toString());

}

write.close();

}

catch(VowelNotFoundException v){

System.out.println("vowel found");

}catch(FileNotFoundException e){

System.out.println(e);

}catch(IOException ex){

System.out.println(ex);

}

}

static boolean isVowel(char c) throws VowelNotFoundException{

if(c == 'a' || c =='e' || c =='i' || c =='o' || c =='u' || c =='A' || c =='E' || c =='I'|| c =='O'|| c =='U'){

return true;

}else{

return false;

}

}

class VowelNotFoundException extends Exception{

String str;

VowelNotFoundException(String str){

this.str = str;

}

public String toString() {

return this.str;

}

}

}

5.Write a java program to implement the following scenario:

a) Create a file named **Integers.txt** and insert n-random integers into it.

b) Create three threads TI, T2 and T3 that read n/3 integers in sequence of occurances of numbers from the file and sort the read n/3 integers

c) Thread T4 waits for all the threads T1, T2 and T3 to complete sorting, then sorts and outputs the entire list of sorted numbers to another file named **SortedIntegers.txt**