Department of Computer Science & Engineering, SDMCET, Dharwad

AOOP Assignment Submission Report

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

Submitted by:

USN:	2SD20CS035	Name:	C S DHANYASHREE

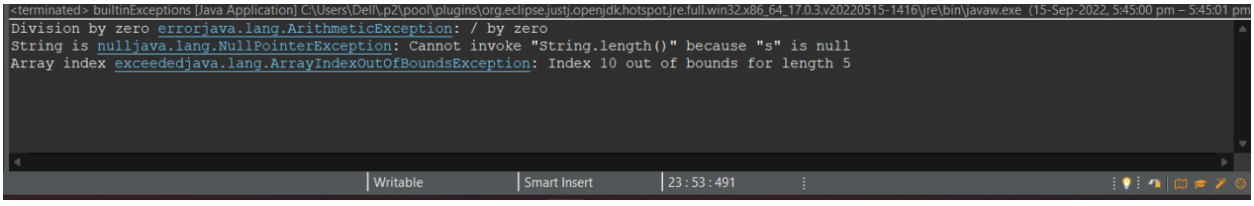
1. Problem definition:

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

2. Java Program:

```
public class builtinExceptions {

	int a=3;
	int b=0;
	String s = null;
	int d[]=new int[5];
	try {
		System.out.println(a/b);
	}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);
	}


}
```

3. Screenshot of excecution:



1.Problem definition:

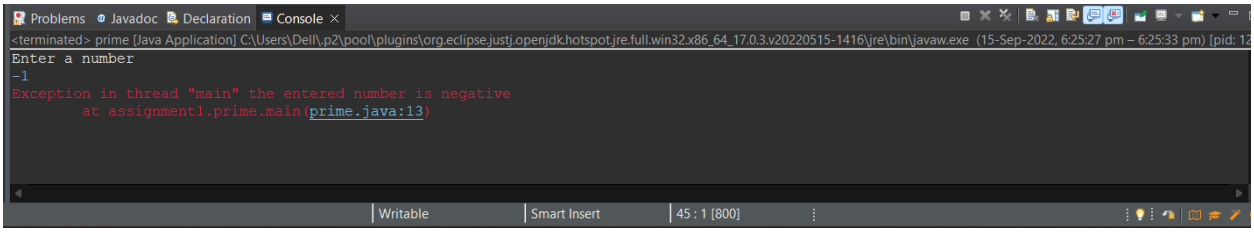
2. Write a Java program to read an integer and check whether the number is prime or not. If negative number is entered, throw an exception NegativeNumberNotAllowedException and if entered number is not prime, then throw NumberNotPrimeException.

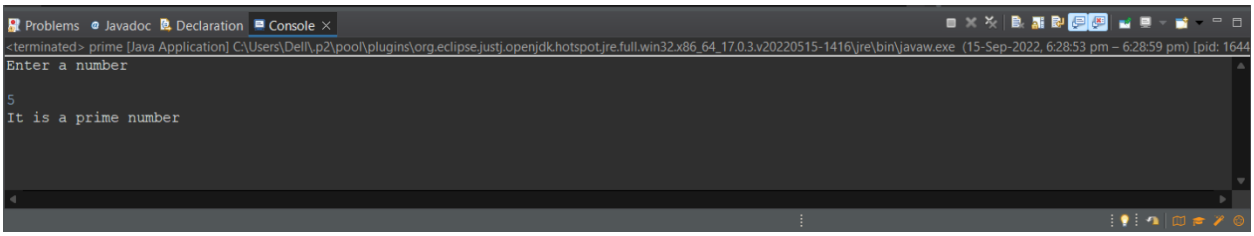
2.Java Program:

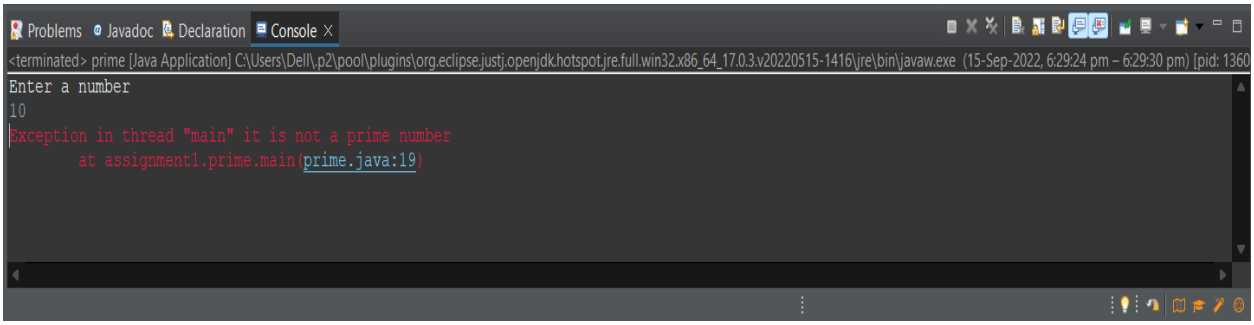
```
       package assignment1;

import java.util.Scanner;

public class prime extends Exception{

     public static void main(String[] args) throws Exception {
          Scanner sc = new Scanner(System.in);
          System.out.println("Enter a number");
          int n=sc.nextInt();

          if(n<0)
               throw new negativeNumberException();
          else
          {
               for(int i=2;i<=n/2;i++) {
                    if(n%i==0)
                    {
                      throw new notPrime();
                    }
                    else
                         System.out.println("It is a prime
number");
               }
          }

     }

}
class negativeNumberException extends Exception{

     void negativeNumberExcaption(){
     }
     public String toString() {
          return "the entered number is negative";
          }
}

class notPrime extends Exception{
     notPrime() {
}
     public String toString() {
          return "it is not a prime number";
```

```
        }
    }
}
```

3.Screenshot of excecution:







1.Problem definition:

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

a) Read a line of text

b) Search for a sub-string SDMCET (case insensitive search)

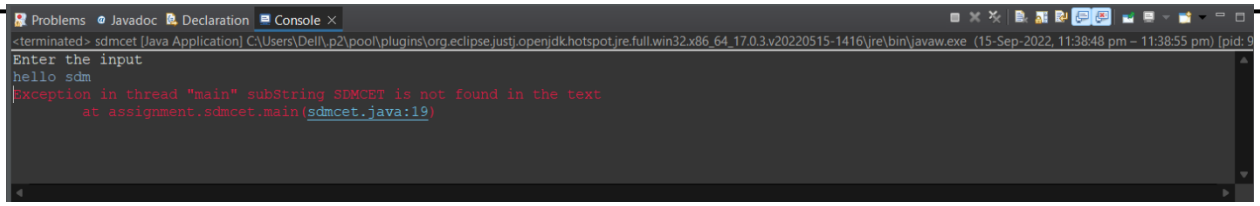
c) If found, then print success message

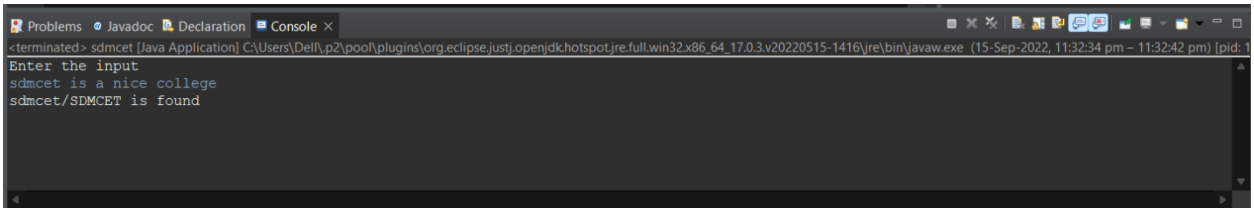
d) Otherwise throw an exception SubStringNotFoundException with appropriate message

2.Java Program:

```
 package assignment;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class sdmcet {

	public static void main(String[] args) throws Exception {
		String text;
		InputStreamReader ir=new InputStreamReader(System.in);
		BufferedReader br = new BufferedReader(ir);
		System.out.println("Enter the input");
		text=br.readLine();
		boolean a =
text.toLowerCase().contains("sdmcet".toLowerCase());
		if(a==true) {
			System.out.println("sdmcet/SDMCET is found");

		}
			if(a==false)
				throw new stringNotFoundException();

		}
}
class stringNotFoundException extends Exception{
	stringNotFoundException(){
}
	public String toString() {
		return "subString SDMCET is not found in the text";
	}

}
```

3.Screenshot of excecution:





1. Problem definition:

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

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

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

c) If vowel is encountered, throw an exception VowelNotAllowedException and continue until end of file

2.Java Program:

```
package assignment2;

public class alphabet {

	public static void main(String[] args) {
		try{
			FileInputStream fin=new
FileInputStream("E:\Users\Dell\assignment1\assignment2\alphabet.txt");
				FileOutputStream fout=new
FileOutputStream("E:\Users\Dell\assignment1\assignment2\consonant.txt");
				int ch;
				while(ch=fin.read()!=-1){
					if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){
						throw new vowelNotAllowedException();
					}
					else
						fout.write(ch);
					}
				}catch(vowelNotAllowedException e){
					System.out.println(e.toString());
				}
	}
	}
	class vowelNotAllowedException extends Exception{
		public String toString(){
			return "vowels are not allowed";
		}
	  }
```

3.Screenshot of excecution: