## CSA0980 -- PROGRAMMING IN JAVA FOR IDL TECHNOLOGY :-

### M . NAGA KESHAVA

### REG NO :-192125019

**1)** Write a java program

i. to compare two strings lexicographically, ignoring case differences.

ii. to check whether a given string ends with the contents of another string.

iii. to print current date and time in the specified format.

iv. to get the index of all the characters of the alphabet.

v. To replace each substring of a given string that matches the given regular

expression with the given replacement. In the below string replace all the fox with

cat.

vi. to get a substring of a given string between two specified positions.

vii. to trim any leading or trailing whitespace from a given string.

viii. to convert all the characters in a string to lowercase.

ix. to get the length of a given string.

x. to check whether two String objects contain the same data

Sample string: &quot;The quick brown fox jumps over the lazy dog.&quot;

**PROGRAMS :-**

i. to compare two strings lexicographically, ignoring case differences.

**PROGRAM :-**

public class twostrings {

        public static void main(String[] args) {

           String str1 = "Hello";

           String str2 = "hello";

           int result = str1.compareToIgnoreCase(str2);

           if (result == 0) {

              System.out.println("Strings are equal.");

           }

           else if (result < 0) {

              System.out.println("String 1 comes before string 2.");

           }

           else {

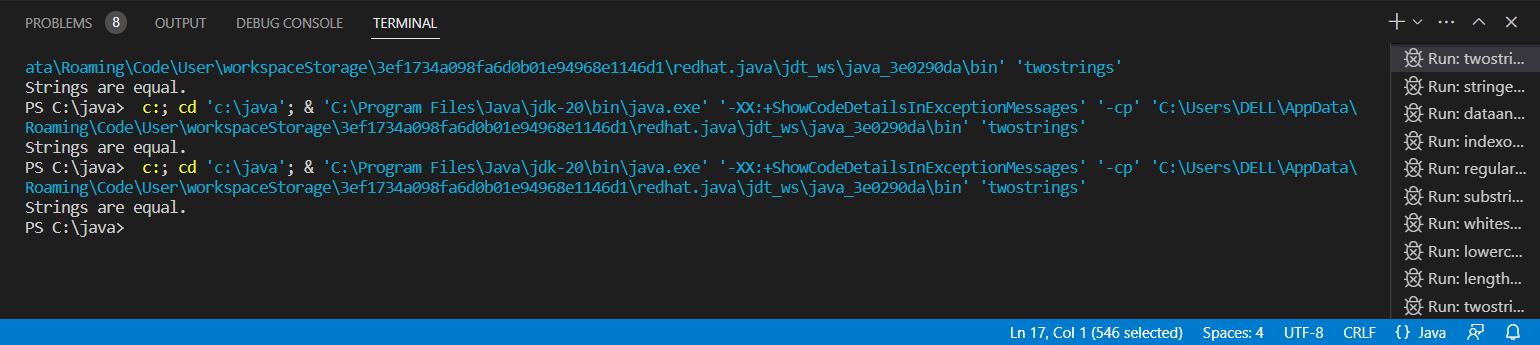
              System.out.println("String 1 comes after string 2.");

           }

        }

     }

**OUTPUT:-**



ii. to check whether a given string ends with the contents of another string.

**PROGRAM :-**

public class stringend {

        public static void main(String[] args) {

           String str1 = "Hello World";

           String str2 = "World";

           boolean result = str1.endsWith(str2);

           if (result) {

              System.out.println("String 1 ends with string 2.");

           }

           else {

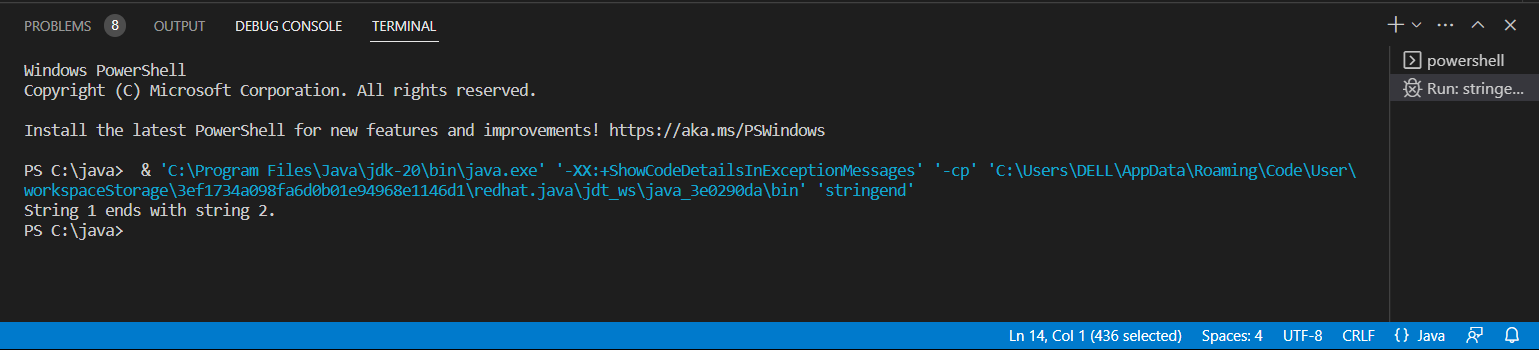
              System.out.println("String 1 does not end with string 2.");

           }

        }

     }

**OUTPUT :-**



iii. to print current date and time in the specified format.

**PROGRAM :-**

import java.text.SimpleDateFormat;

import java.util.Date;

public class dataandtime {

   public static void main(String[] args) {

      SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

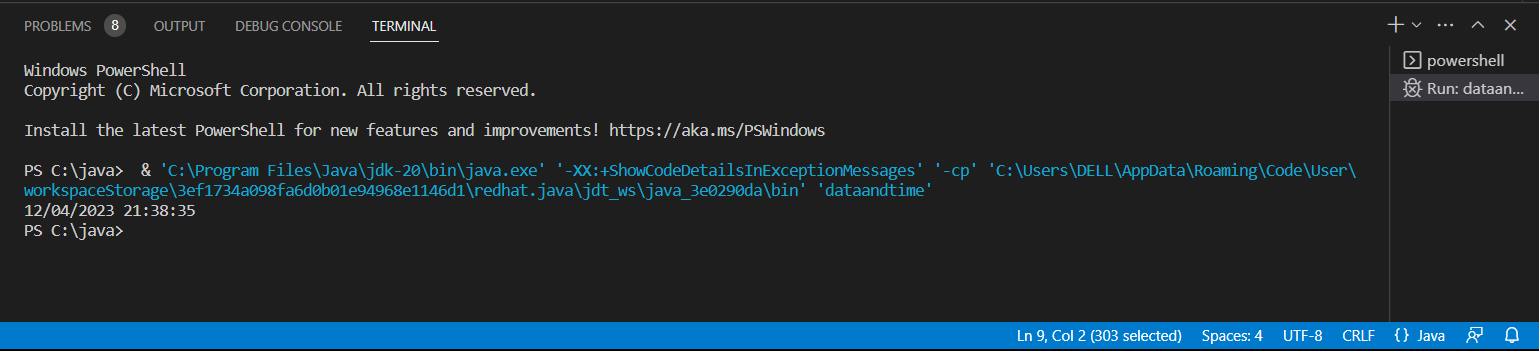
      Date date = new Date();

      System.out.println(formatter.format(date));

   }

}

**OUTPUT :-**



iv. to get the index of all the characters of the alphabet.

**PROGRAM :-**

public class indexofallcharacter {

        public static void main(String[] args) {

           String str = "The quick brown fox jumps over the lazy dog.";

           for (char ch = 'a'; ch <= 'z'; ch++) {

              int index = str.indexOf(ch);

              if (index != -1) {

                 System.out.println(ch + ": " + index);

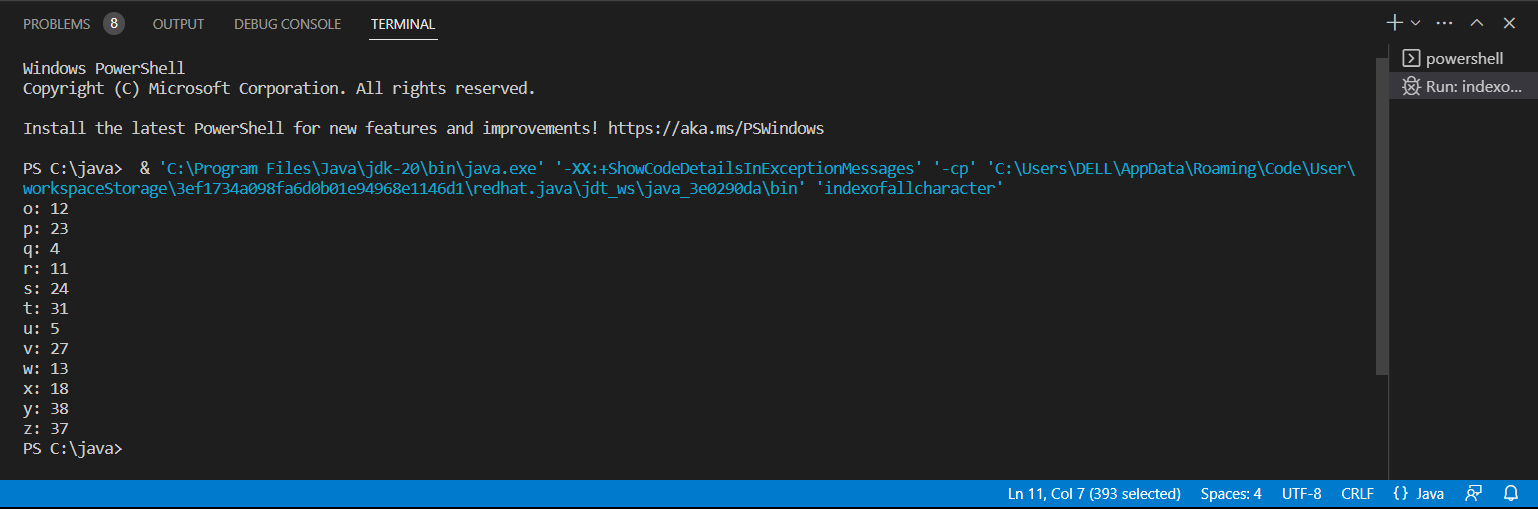
              }

           }

        }

     }

**OUTPUT :-**



v. To replace each substring of a given string that matches the given regular

expression with the given replacement. In the below string replace all the fox with

cat

**PROGRAM :-**

public class regularexpression {

        public static void main(String[] args) {

           String str = "The quick brown fox jumps over the lazy dog.";

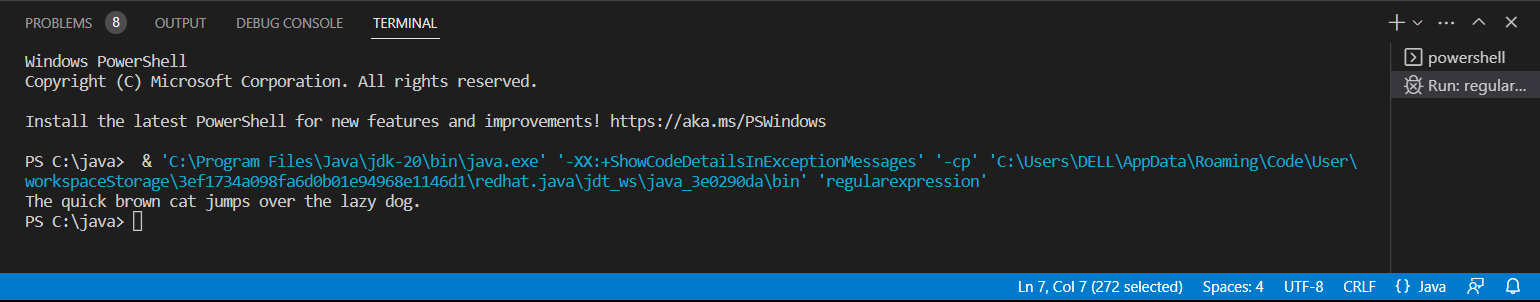
           String newStr = str.replaceAll("fox", "cat");

           System.out.println(newStr);

        }

     }

**OUTPUT :-**



vi. to get a substring of a given string between two specified positions.

**PROGRAM :-**

public class substringbetweentwo {

        public static void main(String[] args) {

           String str = "The quick brown fox jumps over the lazy dog.";

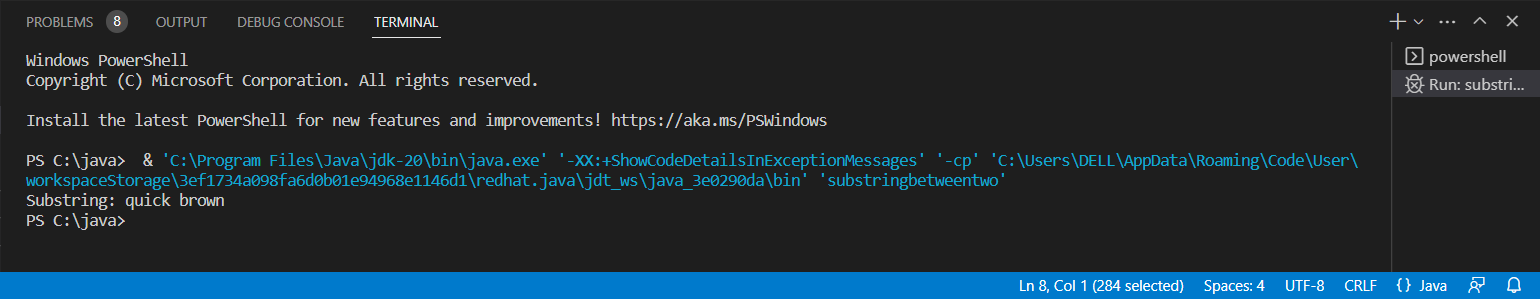
           String substr = str.substring(4, 15);

           System.out.println("Substring: " + substr);

        }

     }

**OUTPUT :-**



vii. to trim any leading or trailing whitespace from a given string.

**PROGRAM :-**

public class whitespace {

        public static void main(String[] args) {

           String str = "   The quick brown fox jumps over the lazy dog.   ";

           String trimmedStr = str.trim();

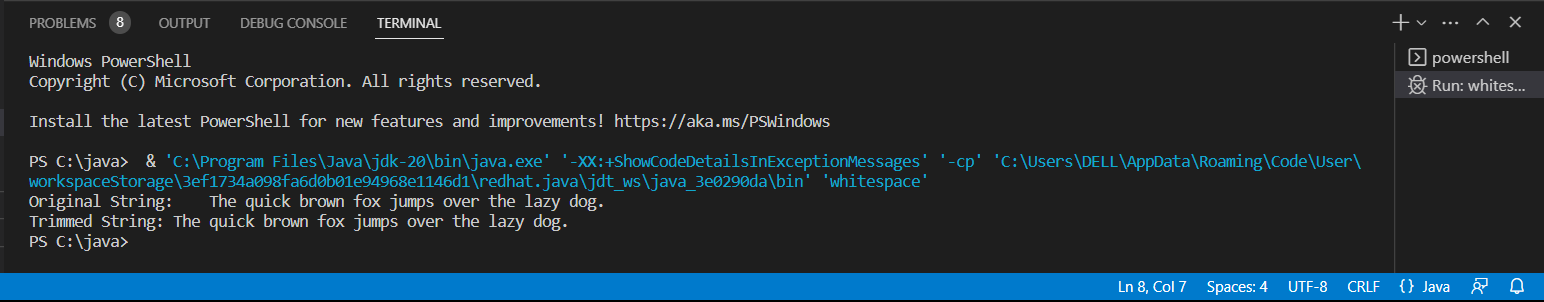
           System.out.println("Original String: " + str);

           System.out.println("Trimmed String: " + trimmedStr);

        }

     }

**OUTPUT :-**



viii. to convert all the characters in a string to lowercase.

**PROGRAM :-**

public class lowercase {

        public static void main(String[] args) {

           String str = "The quick brown Fox Jumps Over The LAZY Dog.";

           String lowercaseStr = str.toLowerCase();

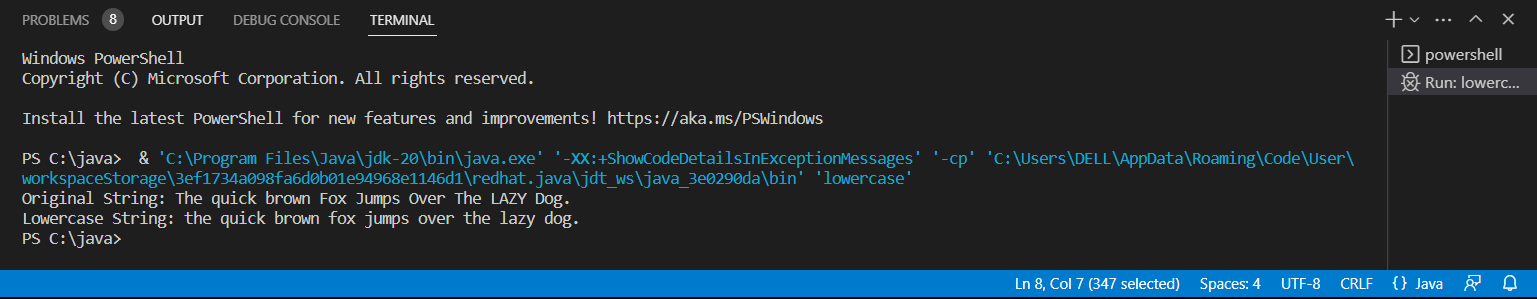
           System.out.println("Original String: " + str);

           System.out.println("Lowercase String: " + lowercaseStr);

        }

     }

**OUTPUT :-**



ix. to get the length of a given string.

**PROGRAM :-**

public class lengthstring {

        public static void main(String[] args) {

           String str = "The quick brown fox jumps over the lazy dog.";

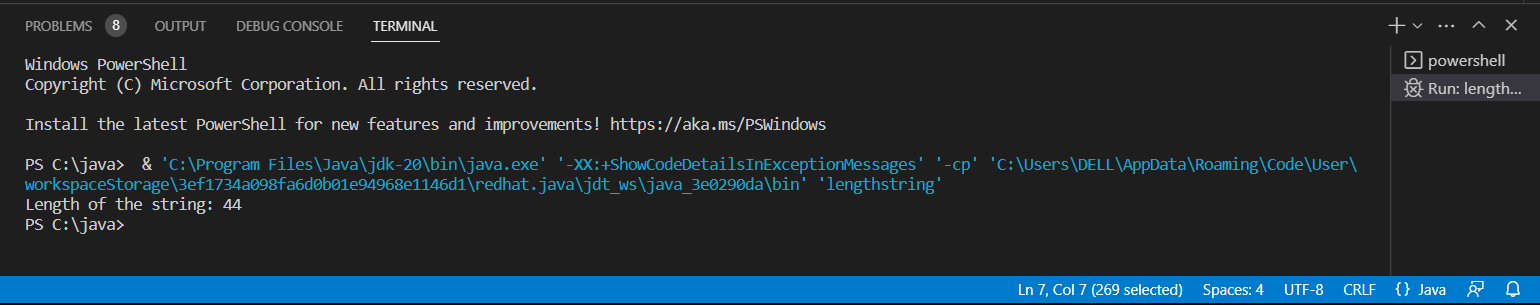
           int len = str.length();

           System.out.println("Length of the string: " + len);

        }

     }

**OUTPUT :-**



x. to check whether two String objects contain the same data

**PROGRAM :-**

public class twostringobtain {

        public static void main(String[] args) {

           String str1 = "The quick brown fox jumps over the lazy dog.";

           String str2 = "The quick brown fox jumps over the lazy dog.";

           if (str1.equals(str2)) {

              System.out.println("Strings are equal.");

           }

           else {

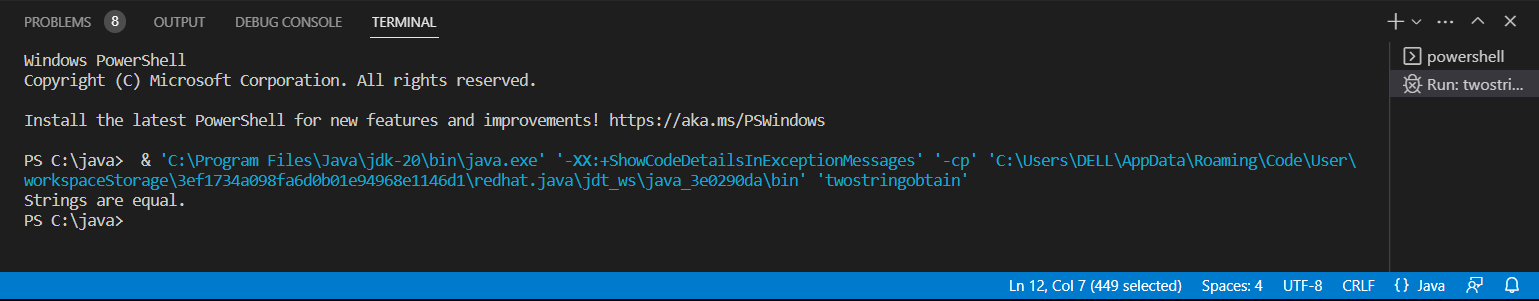
              System.out.println("Strings are not equal.");

           }

        }

     }

**OUTPUT :-**



**2)** Implement a class Account. An account has

* a balance
* functions to add
* and withdraw money,
* And a function to inquire the current balance.

Condition:

1. Pass a value into a constructor to set an initial balance.

2. If no value is passed the initial balance should be set to $0.

3. Charge a $5 penalty if an attempt is made to withdraw more money than available

in the account.

4. Enhance the Account class to compute interest on the current balance.

**PROGRAM :-**

import java.util.\*;

public class account {

        private double balance;

        private double interestRate;

        private static final double PENALTY = 5.0;

        public account() {

            this(0.0);

        }

        public account(double initialBalance) {

            balance = initialBalance;

            interestRate = 0.0;

        }

        public void deposit(double amount) {

            balance += amount;

        }

        public void withdraw(double amount) {

            if (balance >= amount) {

                balance -= amount;

            } else {

                balance -= PENALTY;

                System.out.println("Withdrawal amount exceeds account balance. A $5 penalty has been charged.");

            }

        }

        public double getBalance() {

            return balance;

        }

        public void setInterestRate(double rate) {

            interestRate = rate;

        }

        public void addInterest() {

            double interest = balance \* interestRate / 100.0;

            balance += interest;

        }

    }

**3) Questions for Debugging a code in Java**

Given two strings needle and haystack, return the index of the first occurrence of

needle in haystack, or -1 if needle is not part of haystack.

Example 1:

Input: haystack = &quot;sadbutsad&quot;, needle = &quot;sad&quot;

Output: 0

Explanation: &quot;sad&quot; occurs at index 0 and 6.

The first occurrence is at index 0, so we return 0.

Example 2:

Input: haystack = &quot;leetcode&quot;, needle = &quot;leeto&quot;

Output: -1

Explanation: &quot;leeto&quot; did not occur in &quot;leetcode&quot;, so we return -1.

Constraints:

1 &lt;= haystack.length, needle.length &lt;= 104

haystack and needle consist of only lowercase English characters.

Given a string s consisting of words and spaces, return the length of the last word in the

string.

A word is a maximal

substring

consisting of non-space characters only.

Example 1:

Input: s = &quot;Hello World&quot;

Output: 5

Explanation: The last word is &quot;World&quot; with length 5.

Example 2:

Input: s = &quot; fly me to the moon &quot;

Output: 4

Explanation: The last word is &quot;moon&quot; with length 4.

Example 3:

Input: s = &quot;luffy is still joyboy&quot;

Output: 6

Explanation: The last word is &quot;joyboy&quot; with length 6.

Constraints:

1 &lt;= s.length &lt;= 104

s consists of only English letters and spaces &#39; &#39;.

There will be at least one word in s.

**1) PROGRAM :-**

public class firstoccurance {

        public int strStr(String haystack, String needle) {

            if (needle.isEmpty()) {

                return 0;

            }

            int needleLength = needle.length();

            for (int i = 0; i <= haystack.length() - needleLength; i++) {

                if (haystack.substring(i, i + needleLength).equals(needle)) {

                    return i;

                }

            }

            return -1;

        }

    }

**2) PROGRAM :-**

import java.util.\*;

public class lengthoflastword {

        public static int lengthOfLastWord(String s) {

        s = s.trim();

        int lastSpaceIndex = s.lastIndexOf(' ');

        if (lastSpaceIndex == -1) {

            return s.length();

        } else {

            return s.substring(lastSpaceIndex + 1).length();

        }

    }

        public static void main(String[] args) {

            String input = "Hello World";

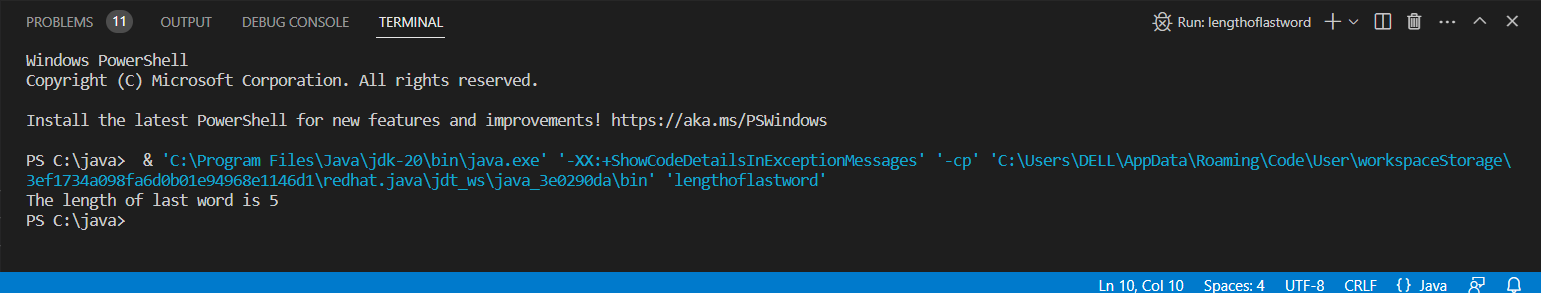
            lengthoflastword a = new lengthoflastword();

            System.out.println("The length of last word is "+a.lengthOfLastWord(input));

        }

    }

**OUTPUT :-**

****

**4) Questions for Finding error in Java to determine the factor**

import java.io.\*;

import java.util.\*;

class factor {

public static void main(String args[]) {

try {

Scanner sc=new Scanner(System.in);

in count=0,n=100,i,j=0,m=4;

int []a=new int [10];

System.out.println(&quot;Enter the number:&quot;);

n=sc.nextInt();

if(n&lt;=0)

{

System.out.println(&quot;Enter valid number&quot;);

}

else {

for(i=1;i&lt;=n;i--);

{

if(n%i!=0)

{

a[j] = i;

System.out.println(&quot;...&quot; + i);

count++;

j++;

}

}

System.out.println(&quot;The number of factors:&quot;+count);

}

System.out.println(m + &quot;th item &quot; + a[m-1]);

}

catch(Exception e) {

System.out.println(&quot;Enter only numbers&quot;);

}

}

}

**PROGRAM :-**

import java.util.\*;

public class assignment

 {

        public static void main(String[] args) {

            int count=0,n,i,j=0,m=4;

            Scanner sc = new Scanner(System.in);

            System.out.println("Enter the number");

            n = sc.nextInt();

            int a[] = new int[10];

           try

           {

               if(n<=0)

               {

                   System.out.println("Enter valid number");

               }

               else

               {

                 for(i=1;i<=n;i++)

                 {

    if(n%i==0)

    {

    a[j] = i;

    System.out.println(" "+ i);

    count++;

    j++;

    }

    }

    System.out.println("The no of factors:"+count);

               }

               System.out.println(m +"th item "+ a[m-1]);

            }

           catch (Exception e)

           {

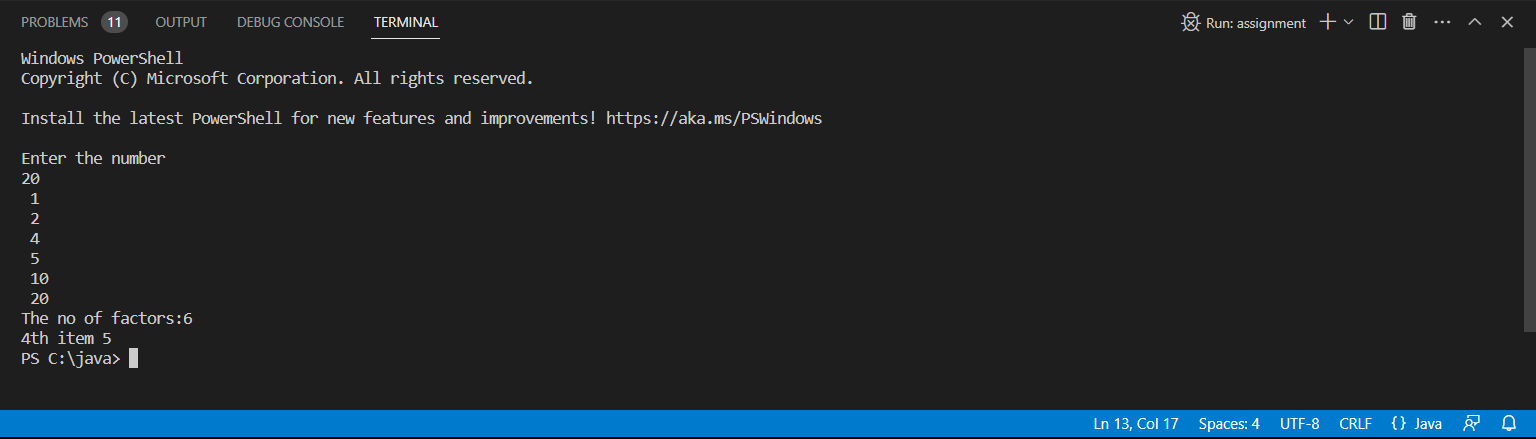
               System.out.println("Enter only numbers");

           }

        }

    }

**OUTPUT :-**

****