

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Science & Engineering

Subject Name: Java Programming**Semester:** 3**Subject Code:** CSE201**Academic year:** 2024 – 25**PART – 6 (File Handling & Streams)**

No.	Aim of the Practical
27.	<p>Write a program that will count the number of lines in each file that is specified on the command line. Assume that the files are text files. Note that multiple files can be specified, as in "java Line Counts file1.txt file2.txt file3.txt". Write each file name, along with the number of lines in that file, to standard output. If an error occurs while trying to read from one of the files, you should print an error message for that file, but you should still process all the remaining files.</p> <p><u>PROGRAM CODE:</u></p> <pre>import java.io.*; public class pra27 { public static void main(String[] args) { if (args.length == 0) { System.out.println("No files specified."); return; } for (String fileName : args) { try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) { int lineCount = 0; while (reader.readLine() != null) { lineCount++; } System.out.println(fileName + ": " + lineCount + " lines"); } } } }</pre>

```

        } catch (IOException e) {
            System.err.println("Error reading file " + fileName + ": " + e.getMessage());
        }
    }
}

```

OUTPUT:

```

C:\Users\shrey\OneDrive\Desktop\3SEM JAVA>javac pra27.java

C:\Users\shrey\OneDrive\Desktop\3SEM JAVA>java pra27 file1.txt file2.txt file3.txt
file1.txt: 0 lines
file2.txt: 2 lines
file3.txt: 8 lines

```

CONCLUSION:

In This Practical We We Use Command Line To Specific File And Then Find Lines From Txt Files.

28.

Write an example that counts the number of times a particular character, such as e, appears in a file. The character can be specified at the command line. You can use xanadu.txt as the input file.

PROGRAM CODE:

```

import java.io.*;
public class pra28 {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: java pra28 <filename> <character>");
            return;
        }

        String filename = args[0];
        String characterInput = args[1];

        if (characterInput.length() != 1) {
            System.out.println("Error: Please specify a single character.");
            return;
        }
    }
}

```

```

char character = characterInput.charAt(0);
int count = countCharacterInFile(filename, character);

    if (count >= 0) {
        System.out.println("The character " + character + " appears " + count + " times
in " + filename + ".");
    }
}

private static int countCharacterInFile(String filename, char character) {
    int count = 0;

    try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
        String line;
        while ((line = reader.readLine()) != null) {
            for (char c : line.toCharArray()) {
                if (c == character) {
                    count++;
                }
            }
        }
    }
    catch (IOException e)
    {
        System.err.println("Error reading the file: " + e.getMessage());
        return -1;
    }

    return count;
}
}

```

OUTPUT:

```

C:\Users\shrey\OneDrive\Desktop\3SEM JAVA>java pra28 xanadu.txt e
The character 'e' appears 9 times in 'xanadu.txt'.

```

CONCLUSION:

In This Practical We Count Characters From Txt File.

29.

Write a Java Program to Search for a given word in a File. Also show use of Wrapper Class with an example.

PROGRAM CODE:

```
import java.io.*;
public class pra29 {

    public static void main(String[] args) {
        if (args.length < 1) {
            System.out.println("Usage: java pra29 <word>");
            return;
        }

        String filename = "pra29.txt";
        String wordToSearch = args[0];
        int wordCount = searchWordInFile(filename, wordToSearch);

        if (wordCount > 0) {
            System.out.println("The word \"" + wordToSearch + "\" was found " +
wordCount + " times.");
        } else {
            System.out.println("The word \"" + wordToSearch + "\" was not found in the
file.");
        }
    }

    public static int searchWordInFile(String filename, String word) {
        int count = 0;

        try {
            File file = new File(filename);
            Scanner scanner = new Scanner(file);

            while (scanner.hasNextLine()) {
                String line = scanner.nextLine();
                String[] words = line.split("\\W+");
                for (String w : words) {
                    if (w.equalsIgnoreCase(word)) {
                        count++;
                    }
                }
            }
        }
    }
}
```

```

        scanner.close();
    } catch (FileNotFoundException e) {
        System.out.println("File not found: " + filename);
    }

    return count;
}
}

```

OUTPUT:

```

C:\Users\shrey\OneDrive\Desktop\3SEM JAVA>java pra29 to
The word "to" was not found in the file.

C:\Users\shrey\OneDrive\Desktop\3SEM JAVA>java pra29 file
The word "file" was found 3 times.

```

CONCLUSION:

In This Practical We Find User Given Word From File And Tell User If It Is In File Or Not.

30. Write a program to copy data from one file to another file. If the destination file does not exist, it is created automatically.

PROGRAM CODE:

```

import java.io.*;
public class pra30 {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: java pra30 <source-file> <destination-file>");
            return;
        }

        String sourceFile = args[0];
        String destinationFile = args[1];

        try {
            copyFile(sourceFile, destinationFile);
            System.out.println("Data copied from " + sourceFile + " to " +
            destinationFile + ".");
        } catch (IOException e) {
            System.err.println("Error occurred while copying the file: " +
            e.getMessage());
        }
    }
}

```

```

    }

    private static void copyFile(String source, String destination) throws IOException
    {
        try (FileInputStream fis = new FileInputStream(source);
            FileOutputStream fos = new FileOutputStream(destination)) {
            byte[] buffer = new byte[1024];
            int bytesRead;

            while ((bytesRead = fis.read(buffer)) != -1) {
                fos.write(buffer, 0, bytesRead);
            }
        }
    }
}

```

OUTPUT:

```

C:\Users\shrey\OneDrive\Desktop\3SEM JAVA>java pra30 source.txt destination.txt
Data copied from 'source.txt' to 'destination.txt'.

```

CONCLUSION:

In This Practical We Performed Writing Data One Txt File To Another Txt File.

31 Write a program to show use of character and byte stream. Also show use of BufferedReader/BufferedWriter to read console input and write them into a file.

PROGRAM CODE:

```

import java.io.*;
public class pra31
{
    public static void main(String[] args)
    {
        String fileName = "output.txt";

        try (InputStreamReader = new InputStreamReader(System.in);
            LineNumberReader consoleReader =
            new LineNumberReader(inputStreamReader);
            FileWriter fileWriter = new FileWriter(fileName))
        {
            System.out.println("Enter text (type 'done' to finish):");

```

```

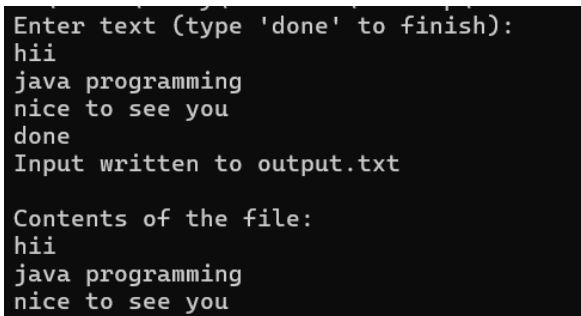
String inputLine;
while (!(inputLine = consoleReader.readLine()).equalsIgnoreCase("done"))
{
    fileWriter.write(inputLine + "\n");
}

System.out.println("Input written to " + fileName);

} catch (IOException e)
{
    e.printStackTrace();
}
System.out.println("\nContents of the file:");
try (FileReader fileReader = new FileReader(fileName))
{
    int character;
    while ((character = fileReader.read()) != -1)
    {
        System.out.print((char) character);
    }
}
catch (IOException e)
{
    e.printStackTrace();
}
}
}

```

OUTPUT:

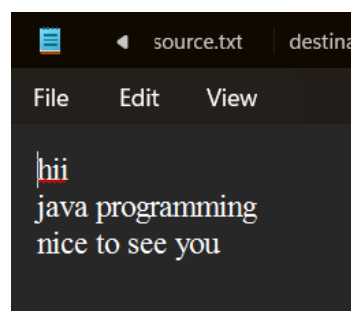


```

Enter text (type 'done' to finish):
hii
java programming
nice to see you
done
Input written to output.txt

Contents of the file:
hii
java programming
nice to see you

```



```

source.txt
File Edit View
hii
java programming
nice to see you

```

CONCLUSION:

In This Practical We Write In Console And Data Will Also Written In Txt File.