**Experiment 1**

|  |  |
| --- | --- |
| **Student Name:** Sambhav Mahajan | **UID:** 23BCS11290 |
| **Branch:** B.E. C.S.E. | **Section/Group:** |
| **Semester:** 4th | **Date of Performance:** 17th March 2025 |
| **Subject Name:** OOP’s using Java | **Subject Code:** 23CSP-202 |

1. **Aim:** Mr. Styles is working on a sentiment analysis program to understand the sentiments conveyed in various sentences.

He needs your assistance in developing a program that analyzes the sentiment of a given sentence and classifies it as positive, negative, or neutral.

* + positive keywords = happy, good, excellent, positive.
  + negative keywords = sad, bad, terrible, negative.
  + Anything else is Neutral.

Create a file named input.txt, the input is written into the file input.txt. The program then reads the sentence from input.txt and sentiment analysis is performed based on predefined positive and negative keywords.

The classified sentiment (positive, negative, or neutral) is written to a new file named **output.txt** and displayed.

Note: The program should use input and output streams (FileInputStream, FileOutputStream, DataInputStream, DataOutputStream) to handle file operations.

1. **Objective:** To develop a sentiment analysis program that reads a sentence from a file, analyzes its sentiment based on predefined positive and negative keywords, and classifies the sentence as positive, negative, or neutral. The program demonstrates the use of file handling techniques in Java by utilizing FileInputStream, FileOutputStream, DataInputStream, and DataOutputStream for reading and writing data through input/output streams.
2. **Java Code:**

import java.io.\*; import java.util.\*; class SentimentAnalyzer {

Set<String> positiveWords = new HashSet<>(Arrays.asList("happy", "good", "excellent",

"positive"));

Set<String> negativeWords = new HashSet<>(Arrays.asList("sad", "bad", "terrible", "negative"));

void getUserInput() throws IOException { Scanner sc = new Scanner(System.in);

System.out.print("Enter a sentence: ");

String input = sc.nextLine();

FileOutputStream fos = new FileOutputStream("input.txt"); DataOutputStream dos = new DataOutputStream(fos); dos.writeUTF(input); dos.close(); sc.close();

}

String analyzeSentiment() throws IOException {

FileInputStream fis = new FileInputStream("input.txt");

DataInputStream dis = new DataInputStream(fis); String sentence = dis.readUTF(); dis.close();

String lower = sentence.toLowerCase(); String sentiment = "neutral"; for (String word : positiveWords) { if (lower.contains(word)) { sentiment = "positive"; break;

}

}

if (sentiment.equals("neutral")) { for (String word : negativeWords) { if (lower.contains(word)) { sentiment = "negative"; break;

}

}

}

FileOutputStream fos = new FileOutputStream("output.txt"); DataOutputStream dos = new DataOutputStream(fos); dos.writeUTF(sentiment); dos.close(); return sentiment;

}

void displayResult() throws IOException {

FileInputStream fis = new FileInputStream("output.txt");

DataInputStream dis = new DataInputStream(fis); String sentiment = dis.readUTF(); dis.close();

System.out.println("Sentiment: " + sentiment);

}

} public class Main {

public static void main(String[] args) {

try {

SentimentAnalyzer analyzer = new SentimentAnalyzer(); analyzer.getUserInput(); analyzer.analyzeSentiment(); analyzer.displayResult(); } catch (IOException e) { e.printStackTrace();

}

}

}

**4. Output:**

