Practical No – 3

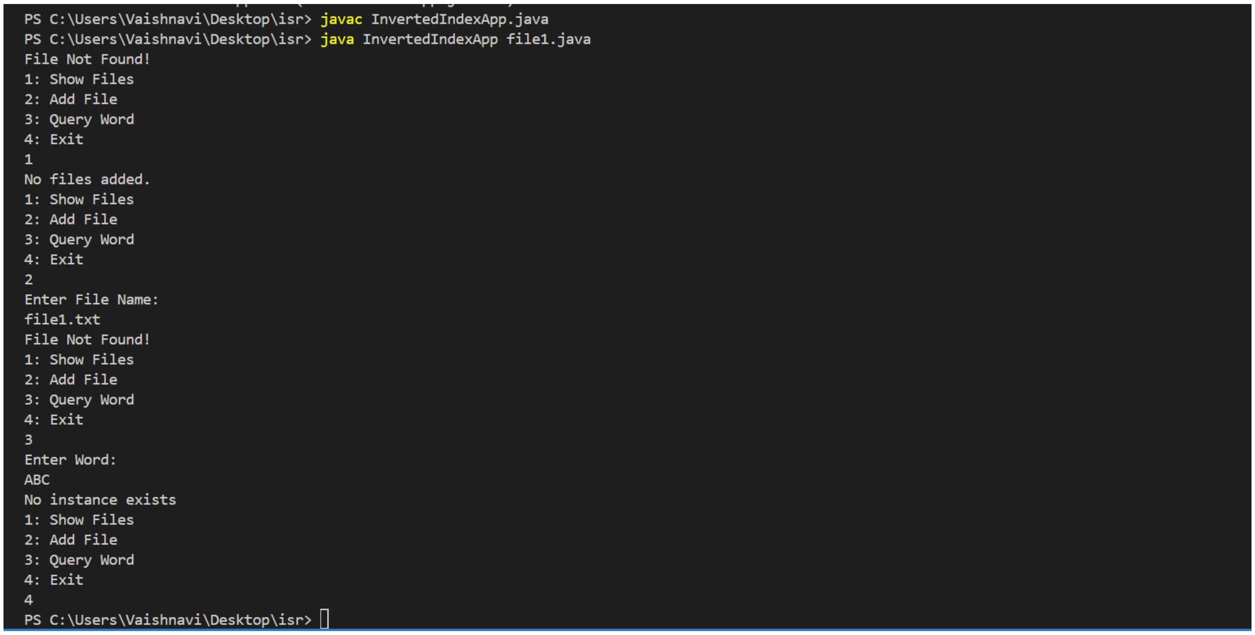
Title – Implement a program for retrieval of documents using inverted files.

Program :

```
import java.io.*;
import java.util.*;

class WordPosition {
    String fileName;
    int line;
    int index;

    WordPosition(String fileName, int line, int index) {
        this.fileName = fileName;
        this.line = line;
        this.index = index;
    }
}

class InvertedIndex {
    private Map<String, List<WordPosition>> dictionary = new HashMap<>();
    private List<String> fileList = new ArrayList<>();

    public void addFile(String fileName) {
        try (BufferedReader reader = new BufferedReader(new FileReader(fileName + ".txt"))) {
            fileList.add(fileName);
            String line;
            int lineNumber = 0;

            while ((line = reader.readLine()) != null) {
```

```
            lineNumber++;
            String[] words = line.split("\\s+");
            for (int wordIndex = 0; wordIndex < words.length; wordIndex++) {
                String word = words[wordIndex];
                WordPosition position = new WordPosition(fileName, lineNumber, wordIndex + 1);

                dictionary.computeIfAbsent(word, k -> new ArrayList<>()).add(position);
            }
        }
    } catch (FileNotFoundException e) {
        System.out.println("File Not Found!");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public void showFiles() {
    if (fileList.isEmpty()) {
        System.out.println("No files added.");
    } else {
        for (int i = 0; i < fileList.size(); i++) {
            System.out.println(i + ": " + fileList.get(i));
        }
    }
}


public void search(String word) {
    if (!dictionary.containsKey(word)) {
            System.out.println("No instance exists");
            return;
        }

        List<WordPosition> positions = dictionary.get(word);
        for (int i = 0; i < positions.size(); i++) {
            WordPosition pos = positions.get(i);
            System.out.println(i + ":");
            System.out.println(" Filename: " + pos.fileName);
            System.out.println(" Line Number: " + pos.line);
            System.out.println(" Index: " + pos.index);
        }
    }
}

public class InvertedIndexApp {
    public static void main(String[] args) {
        InvertedIndex data = new InvertedIndex();

        for (String arg : args) {
            data.addFile(arg);
        }

        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("1: Show Files\n2: Add File\n3: Query Word\n4: Exit");
            choice = sc.nextInt();
            switch (choice) {
                case 1:
                    data.showFiles();
                    break;

                case 2:
                    System.out.println("Enter File Name: ");
                    String name = sc.next();
                    data.addFile(name);
                    break;

                case 3:
                    System.out.println("Enter Word: ");
                    String word = sc.next();
                    data.search(word);
                    break;

                case 4:
                    break;

                default:
                    System.out.println("Invalid choice, please try again.");
            }
        } while (choice != 4);

        sc.close();
    }
}
```

OUTPUT :



```
PS C:\Users\Vaishnavi\Desktop\isr> javac InvertedIndexApp.java
PS C:\Users\Vaishnavi\Desktop\isr> java InvertedIndexApp file1.java
File Not Found!
1: Show Files
2: Add File
3: Query Word
4: Exit
1
No files added.
1: Show Files
2: Add File
3: Query Word
4: Exit
2
Enter File Name:
file1.txt
File Not Found!
1: Show Files
2: Add File
3: Query Word
4: Exit
3
Enter Word:
ABC
No instance exists
1: Show Files
2: Add File
3: Query Word
4: Exit
4
PS C:\Users\Vaishnavi\Desktop\isr> 
```