**BCS122 Data Structure and Algorithms**

Linked List lab

**Lab #4** List Processing and pointers to structure

Objective of the lab:

 Understanding of Pointers to structure –

 Structure in link list form

 Various operations in link list

In this lab students are required to complete the following assignments:

**Q. No. 1.** Write a program to store student id, name and gpa using link-list Store data of

10 student and apply following operations.

 Enter data of ten students in the link list

 Enter a new record in between the third and fourth record of the list

 Display data of students in list form

 Search for a particular student

 Delete a particular student from the list

Take Home Assignment

Write a program to simulate a dictionary using linked list. It should be a menu driven

program with the options for adding a word and its meanings, searching a word and

displaying the dictionary. Steps to develop the program are as given below:

1. Declare a structure with the fields as

- a word,

- meaning of a word

- counter that holds the number of meanings

- link to the next node.

Each word added to the list can have maximum 5 meaning(s). Hence, variable used to

store meaning(s) of a word would be a two dimensional character array.

2. The program should have following menu.

- Add a word

- Search for a word

- Show dictionary

- Exit

import java.util.Scanner;

class DictionaryNode {

String word;

String[] meanings; // Array to store multiple meanings

int meaningCount; // Counter to hold the number of meanings

DictionaryNode next;

// Constructor

public DictionaryNode(String word, String[] meanings, int meaningCount) {

this.word = word;

this.meanings = new String[5]; // Max 5 meanings

for (int i = 0; i < meaningCount; i++) {

this.meanings[i] = meanings[i];

}

this.meaningCount = meaningCount;

this.next = null;

}

}

class DictionaryLinkedList {

private DictionaryNode head;

public DictionaryLinkedList() {

this.head = null;

}

// Add a new word and its meanings

public void addWord(String word, String[] meanings, int meaningCount) {

DictionaryNode newNode = new DictionaryNode(word, meanings, meaningCount);

if (head == null) {

head = newNode;

} else {

DictionaryNode current = head;

while (current.next != null) {

current = current.next;

}

current.next = newNode;

}

System.out.println("Word added successfully!");

}

// Search for a word

public void searchWord(String word) {

DictionaryNode current = head;

while (current != null) {

if (current.word.equals(word)) {

System.out.println("Word: " + current.word);

System.out.println("Meanings:");

for (int i = 0; i < current.meaningCount; i++) {

System.out.println((i + 1) + ". " + current.meanings[i]);

}

return;

}

current = current.next;

}

System.out.println("Word not found in the dictionary.");

}

// Display the entire dictionary

public void displayDictionary() {

if (head == null) {

System.out.println("The dictionary is empty.");

return;

}

DictionaryNode current = head;

while (current != null) {

System.out.println("Word: " + current.word);

System.out.println("Meanings:");

for (int i = 0; i < current.meaningCount; i++) {

System.out.println((i + 1) + ". " + current.meanings[i]);

}

System.out.println();

current = current.next;

}

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

DictionaryLinkedList dictionary = new DictionaryLinkedList();

int choice;

do {

System.out.println("\n--- Dictionary Menu ---");

System.out.println("1. Add a word");

System.out.println("2. Search for a word");

System.out.println("3. Show dictionary");

System.out.println("4. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter word: ");

String word = scanner.nextLine();

System.out.print("How many meanings? (Max 5): ");

int meaningCount = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (meaningCount > 5) {

System.out.println("You can only add a maximum of 5 meanings.");

break;

}

String[] meanings = new String[5];

for (int i = 0; i < meaningCount; i++) {

System.out.print("Enter meaning " + (i + 1) + ": ");

meanings[i] = scanner.nextLine();

}

dictionary.addWord(word, meanings, meaningCount);

break;

case 2:

System.out.print("Enter the word to search: ");

String searchWord = scanner.nextLine();

dictionary.searchWord(searchWord);

break;

case 3:

System.out.println("\n--- Dictionary ---");

dictionary.displayDictionary();

break;

case 4:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Please try again.");

}

} while (choice != 4);

scanner.close();

}

}