

Department of Computer Science & Engineering (CSE)

Project Report

Course Code : CSE-214

Course Title : Object Oriented Programming Sessional

Submission Date : 27-11-2024

Submitted To:

Mousumi Hasan,

Assistant Professor,

Dept. of CSE , BAIUST.

Submitted By:

Name : Md. Shafin Ahammad Hredoy

ID : 0822220105101009

Level-Term : 2-1

Section : A

Department : CSE

***Project Requirement Analysis***

***Purpose:***

The project aims to implement a program for analyzing words and questions based on Bloom's Taxonomy. It enables users to search for a specific word and analyze questions to determine the corresponding Bloom's Taxonomy level.

***Features:***

1. Word Search: Users can input a word, and the program identifies the associated Bloom's Taxonomy level.
2. Question Analysis: Users can input a question, and the program analyzes each word's Bloom's Taxonomy level.

***Technical Requirements:***

* C++ programming language
* File I/O for storing and retrieving predefined Bloom's Taxonomy content
* Object-oriented programming principles for class design

***Classes Identified***

***Class 1: BloomsTaxonomy***

* Responsibility: Manages file operations for reading and writing Bloom's Taxonomy content.
* Methods:
  + Constructor: Opens the file for writing with predefined content.
  + Display: Displays the found level in a formatted way.
  + Destructor: Closes the file.

***Class 2: BloomsTaxonomyWord (Derived from BloomsTaxonomy)***

* Responsibility: Handles word search functionality.
* Methods:
  + Constructor: Opens the file for reading.
  + Run: Accepts a word, searches for its Bloom's Taxonomy level, and displays the result.
  + Display: Displays the found level in a formatted way.
  + Destructor: Closes the file.

***Class 3: BloomsTaxonomyQuestion (Derived from BloomsTaxonomy)***

* Responsibility: Handles question analysis functionality.
* Methods:
  + Run: Accepts a question, analyzes each word's Bloom's Taxonomy level, and displays the result.
  + ExtractWords: Extracts words from a sentence.
  + Display: Displays the found level in a formatted way.

***Relations Among Classes and Methods***

BloomsTaxonomyWord and BloomsTaxonomyQuestion are derived classes from BloomsTaxonomy, inheriting file management functionality.

Both derived classes utilize the Display method from BloomsTaxonomy for presenting results.

BloomsTaxonomyWord uses file operations provided by the base class for reading predefined Bloom's Taxonomy content.

BloomsTaxonomyQuestion uses the ExtractWords method to process the input question.

The main function interacts with the user, creating instances of BloomsTaxonomyWord or BloomsTaxonomyQuestion based on user input.

This design promotes code reusability and modularity, separating concerns based on class responsibilities. The relationship among classes allows for efficient file handling and consistent display methods.

***Code***

/\*

    Bangladesh Army International University of Science and Technology.

    Project name: Bloom's Taxonomy (Word and question level analyzation).

    Name: MD Shafin Ahammad Hredoy

    ID: 0822220105101009

    Level: 02 , Term: 01

    Section: A

\*/

#include <bits/stdc++.h>

using namespace std;

// Base class for Bloom's Taxonomy

class Blooms\_Taxonomy {

protected:

    fstream file; // File stream to read and write

public:

    Blooms\_Taxonomy() {

        // Constructor: Open the file for writing with predefined Bloom's Taxonomy content

        file.open("Blooms.txt", ios::out);

        file << "1.Remembering :\tDefine\tList\tRecall\tMemorize\tIdentify\tRepeat\tName " << endl << endl;

        file << "2.Understanding :\tExplain\tSummarize\tParaphrase\tInterpret\tInfer\tClassify\tCompare\tContrast " << endl << endl;

        file << "3.Applying :\tApply\tUse\tImplement\tSolve\tDemonstrate\tIllustrate\tOperate\tCalculate " << endl << endl;

        file << "4.Analyzing :\tAnalyze\tCompare\tContrast\tDifferentiate\tDistinguish\tExamine\tBreak down " << endl << endl;

        file << "5.Evaluating :\tEvaluate\tJudge\tAssess\tCritique\tJustify\tValidate " << endl << endl;

        file << "6.Creating :\tCreate\tDesign\tInvent\tGenerate\tCompose\tDevelop\tConstruct " << endl << endl;

    }

    void Display(string s) {

        cout << "Found in level " << s << "." << endl;

    }

    ~Blooms\_Taxonomy() {

        file.close(); // Destructor: Close the file

    }

};

// Derived class for word search in Bloom's Taxonomy

class Blooms\_Taxonomy\_word : public Blooms\_Taxonomy {

private:

    ifstream myfile;

    string given\_word;

    string file\_word;

    string level;

public:

    Blooms\_Taxonomy\_word() {

        myfile.open("Blooms.txt", ios::in);

    }

    void Run() {

        cout << "\t\_\_\_WORD SEARCH\_\_\_" << endl << endl;

        cout << "Enter a Word: ";

        cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Ignore input buffer

        cin >> given\_word;

        transform(given\_word.begin(), given\_word.end(), given\_word.begin(), ::tolower);

        bool found = false;

        while (myfile >> file\_word) {

            transform(file\_word.begin(), file\_word.end(), file\_word.begin(), ::tolower);

            if (file\_word[0] >= '1' && file\_word[0] <= '6')

                level = file\_word;

            if (file\_word == given\_word) {

                Display(level);

                found = true;

            }

        }

        if (!found)

            cout << "The word is not of Bloom's Taxonomy library!!!" << endl;

        system("Pause");

        system("cls"); // Clear the console screen

    }

    void Display(string s) {

        // Display the found level in a formatted way

        cout << "The word found in level " << s[0] << " which is (" << (char)(s[2] - 32);

        for (int i = 3; i < s.length(); i++)

            cout << s[i];

        cout << ") level of Bloom's Taxonomy." << endl;

    }

    ~Blooms\_Taxonomy\_word() {

        myfile.close();

    }

};

// Derived class for analyzing questions in Bloom's Taxonomy

class Blooms\_Taxonomy\_question : public Blooms\_Taxonomy {

private:

    ifstream myfile;

    string question;

    vector<string> qs\_word;

    string file\_word;

    string level;

public:

    // Function to run the question analysis

    void Run() {

        cout << "\t\_\_\_QUESTION ANALYSIS\_\_\_" << endl << endl;

        cout << "Enter a Question: ";

        cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Ignore input buffer

        getline(cin, question);

        transform(question.begin(), question.end(), question.begin(), ::tolower);

        qs\_word = extractWords(question);

        bool found = false;

        for (const auto &word : qs\_word) {

            ifstream myfile("Blooms.txt", ios::in);

            while (myfile >> file\_word) {

                transform(file\_word.begin(), file\_word.end(), file\_word.begin(), ::tolower);

                if (file\_word[0] >= '1' && file\_word[0] <= '6')

                    level = file\_word;

                if (file\_word == word) {

                    Display(level);

                    found = true;

                }

            }

            myfile.close();

        }

        if (!found)

            cout << "The question is not of Bloom's Taxonomy library!!!" << endl;

        system("Pause");

        system("cls"); // Clear the console screen

    }

    // Function to extract words from a sentence

    vector<string> extractWords(const string &sentence) {

        vector<string> words;

        string word;

        for (char ch : sentence) {

            if (ch == ' ' || ch == '\t') {

                if (!word.empty()) {

                    words.push\_back(word);

                    word.clear();

                }

            } else {

                word += ch;

            }

        }

        if (!word.empty()) {

            words.push\_back(word);

        }

        return words;

    }

    void Display(string s) {

        // Display the found level in a formatted way

        cout << "The question refers to the level " << s[0] << " (";

        for (int i = 2; i < s.length(); i++)

            cout << s[i];

        cout << ") of Bloom's Taxonomy." << endl;

    }

};

int main() {

    UP:

    int num;

    cout << endl << endl << "\t\_\_\_BLOOM's TAXONOMY\_\_\_" << endl << endl;

    cout << "Enter 1 for Word Search." << endl;

    cout << "Enter 2 for Question Analysis." << endl;

    cout << "Enter 3 for Exit." << endl;

    cout << ">>> ";

    cin >> num;

    if (num == 1) {

        Blooms\_Taxonomy\_word a;

        a.Run();

    } else if (num == 2) {

        Blooms\_Taxonomy\_question a;

        a.Run();

    } else if (num == 3) {

        return 0;

    } else {

        cout << "Invalid Input! Please Try again!" << endl;

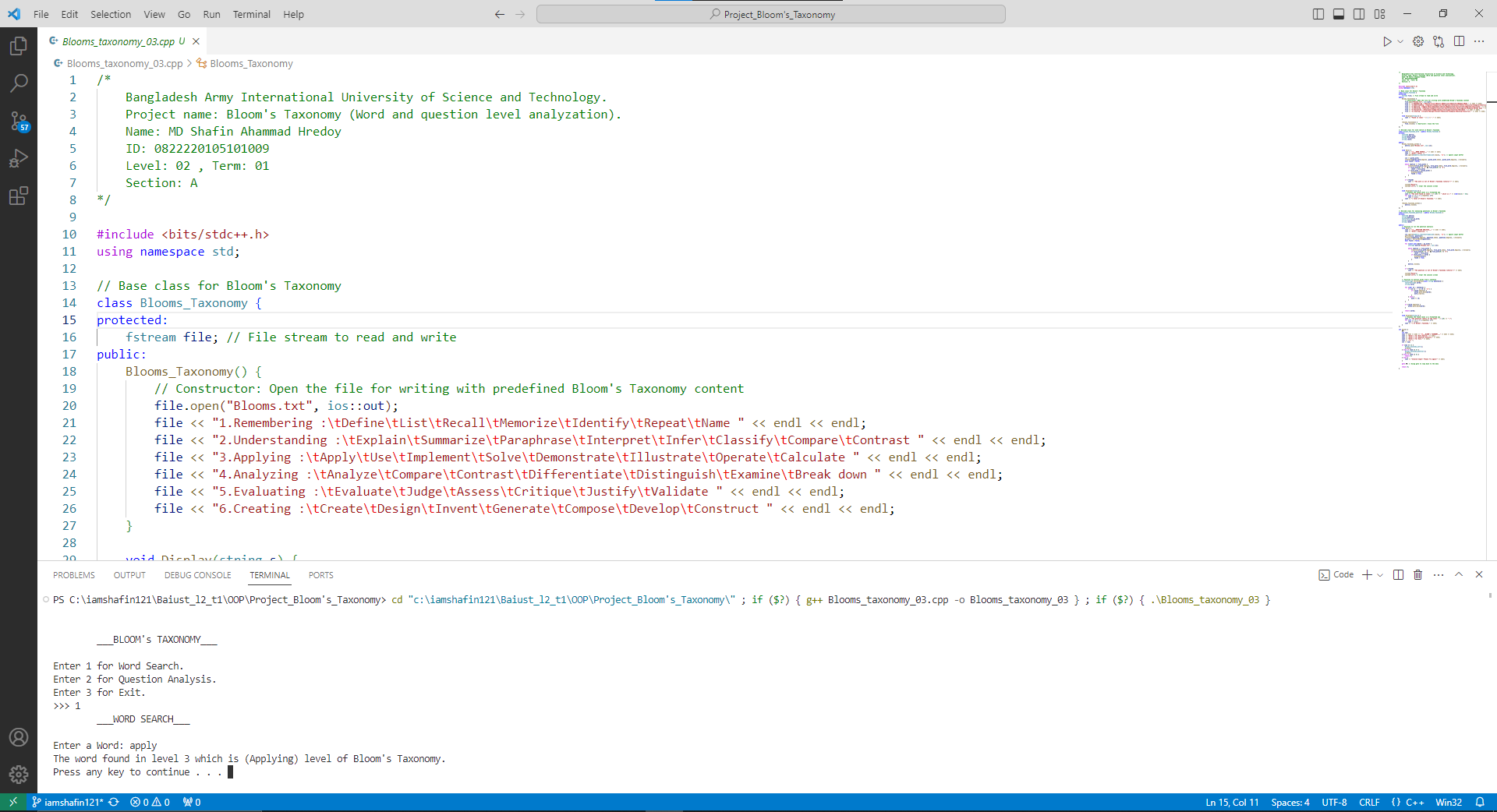
    }

    goto UP; // Using goto to loop back to the menu

    return 0;

}

***Output***

******

***Conclusion***

In conclusion, the Bloom's Taxonomy project efficiently fulfills its objectives, offering users a user-friendly platform for word and question analysis based on educational taxonomy levels. The object-oriented design, comprising BloomsTaxonomy, BloomsTaxonomyWord, and BloomsTaxonomyQuestion classes, promotes code modularity and readability. Adherence to technical requirements, including C++ usage and file I/O, ensures a robust and scalable solution. The program's successful implementation, encapsulating key features and maintaining a clear separation of concerns, establishes it as an effective tool for educational analysis.