**PRACTICAL 15**

**HARSH VEKARIYA**

**23162121024**

**BATCH 24**

**CODE :**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

// Base class with file handling methods

class university {

protected:

    string filename;

public:

    university(const string& filename) : filename(filename) {}

    // Function to write data to file

    void writeToFile(const string& data) {

        ofstream file(filename, ios::app);

        if (file.is\_open()) {

            file << data << endl;

            file.close();

        } else {

            cout << "Unable to open file." << endl;

        }

    }

    // Function to read data from file

    string readFromFile() {

        string data;

        ifstream file(filename);

        if (file.is\_open()) {

            string line;

            while (getline(file, line)) {

                data += line + "\n";

            }

            file.close();

        } else {

            cout << "Unable to open file." << endl;

        }

        return data;

    }

};

// Template function module

class student : public university {

public:

    student(const string& filename) : university(filename) {}

    // Function to display data based on user's choice

    template <typename T>

    void displayData(const T& data) {

        cout << "Data: " << data << endl;

    }

};

// Template class module

template <typename T>

class TemplateClassModule : public university {

public:

    TemplateClassModule(const string& filename) : university(filename) {}

    // Function to manipulate data (example: sort)

    void manipulateData(T& data) {

        // Implementation based on T

    }

};

int main() {

    // File names for each module

    string templateFunctionFile = "template\_function\_data.txt";

    string templateClassFile = "template\_class\_data.txt";

    // Creating objects for each module

    student tfModule(templateFunctionFile);

    TemplateClassModule<string> tcModule(templateClassFile);

    // Experimenting with minimum 5 data/records

    for (int i = 1; i <= 5; ++i) {

        // Example: Asking user for data and storing it using template function module

        string data;

        cout << "Enter data " << i << ": ";

        getline(cin, data);

        tfModule.writeToFile(data);

        // Example: Asking user for data and storing it using template class module

        cout << "Enter data " << i << ": ";

        getline(cin, data);

        tcModule.writeToFile(data);

    }

    // Displaying data according to user's choice

    cout << "Choose an option to display data:" << endl;

    cout << "1. Display data from template function module" << endl;

    cout << "2. Display data from template class module" << endl;

    int choice;

    cin >> choice;

    switch (choice) {

        case 1: {

            // Displaying data from template function module

            string data = tfModule.readFromFile();

            tfModule.displayData(data);

            break;

        }

        case 2: {

            // Displaying data from template class module

            string data = tcModule.readFromFile();

            cout << "Data from template class module: " << data << endl;

            break;

        }

        default:

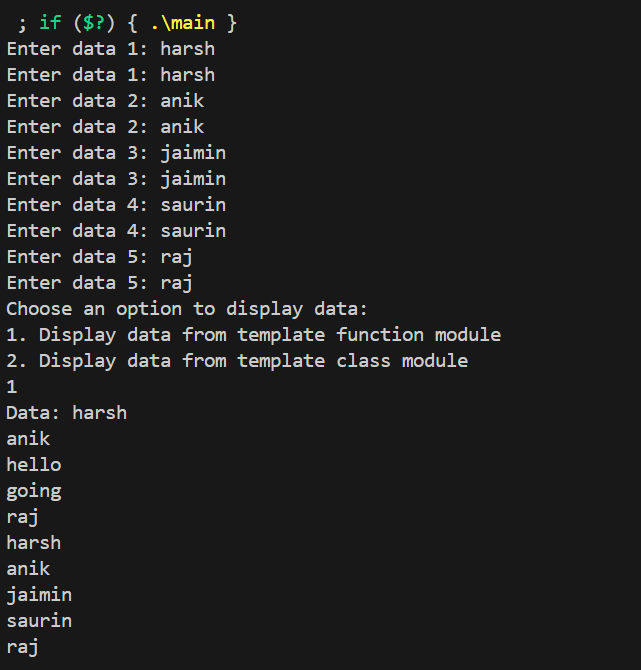
            cout << "Invalid choice." << endl;

    }

    return 0;

}

**OUTPUT:**

****