Ganpat University

Faculty of Engineering & Technology

Computer Science & Engineering

***Name:-* Dwij Vatsal Desai**

***Sem:-* 2**

***Sub: -* ESFP-II**

***Enrollment No.:-* 23162121027**

***Prac:-* 15**

Practical 15

Definition:

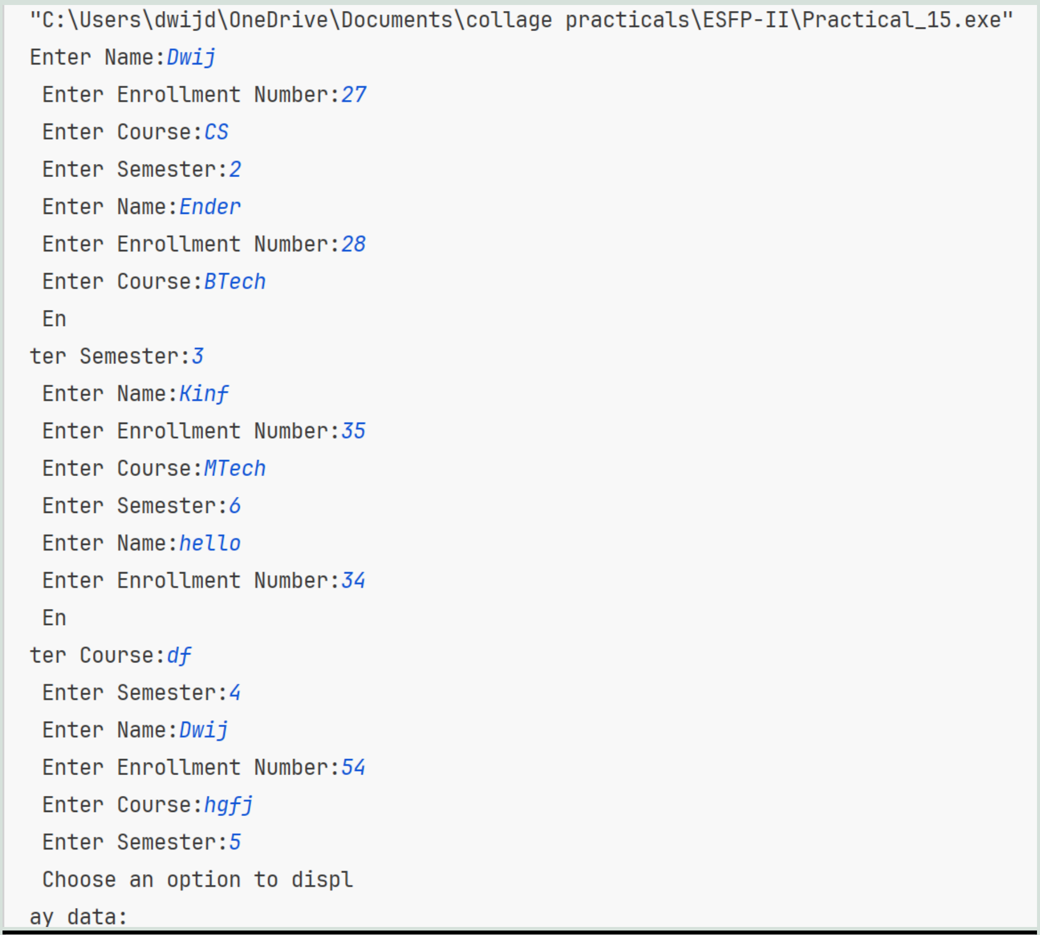
Complete the code for the object assigned to you to satisfy the following specifications.

1. For the solving purpose of given topic practical, you need to create minimum two classes, rest as per your need / requirement for the practical.
2. You must declare minimum one constructor and rest; you can declare as per your requirements.
3. You must use access specifier for data member and member function declaration in program.
4. You have to use the concept of inheritance also.
5. Create two separate modules for your problem definition, one module should implement the concept of “template function” and another module should implement the concept of “template class”. Data member and member function should be declared as per your module (based on definition) requirements.
6. You can use file handling concept, wherever is required to store data in file.
7. Experiments with minimum 5 data/record from the user and display according to the choice of user category wise. (Minimum five different options should be there for displaying information, and if you want more, as per program requirement you can add more choices).

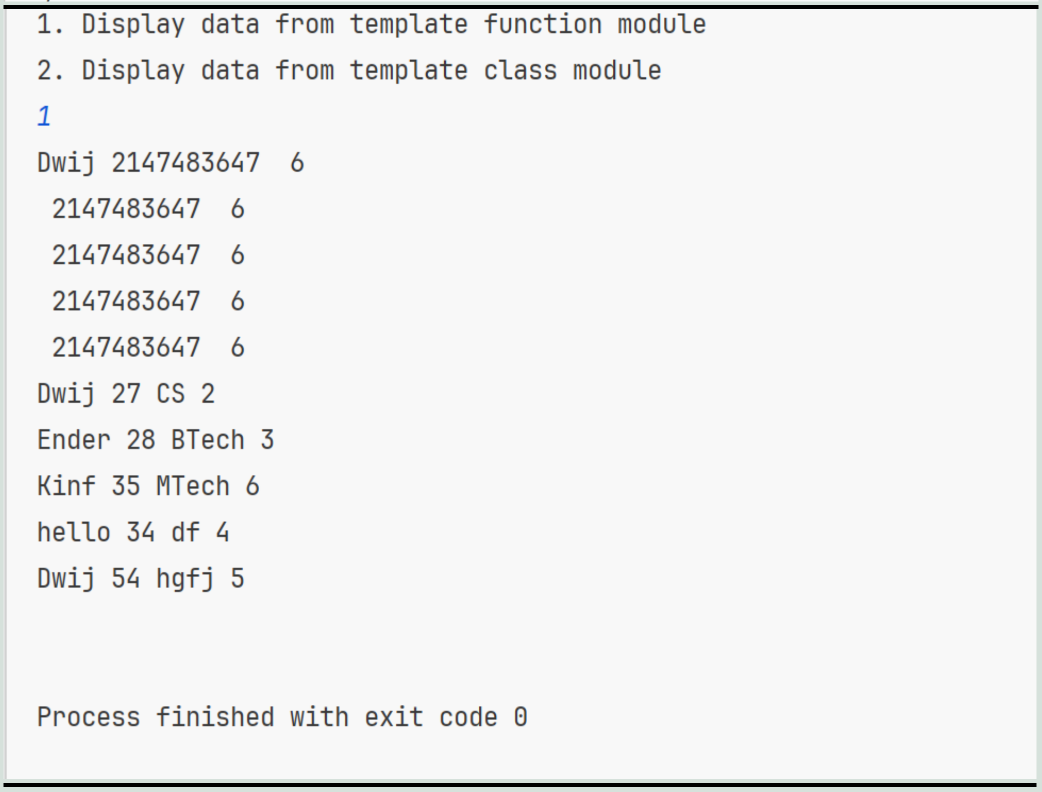
.Code:-

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>

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;
    }
};

// Class to represent a student
class Student {
public:
    string Name;
    int enroll_no;
    string course;
    int sem;

    // Friend function to allow input/output operations
    friend istream& operator>>(istream& is, Student& s);
    friend ostream& operator<<(ostream& os, const Student& s);
};

// Input/output operator overloads
istream& operator>>(istream& is, Student& s) {
    cout << "Enter Name: ";
    is >> s.Name;
    cout << "Enter Enrollment Number: ";
    is >> s.enroll_no;
    cout << "Enter Course: ";
    is >> s.course;
    cout << "Enter Semester: ";
    is >> s.sem;
    return is;
}

ostream& operator<<(ostream& os, const Student& s) {
    os << s.Name << " " << s.enroll_no << " " << s.course << " " << s.sem;
    return os;
}

// Template function module
class StudentModule : public University {
public:
    StudentModule(const string& filename) : University(filename) {}

    // Function to display data based on user's choice
    template <typename T>
    void displayData(const T& data) {
        cout << 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
    StudentModule tfModule(templateFunctionFile);
    TemplateClassModule<Student> 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
        Student s;
        cin >> s;
        string data = (ostringstream() << s).str();
        tfModule.writeToFile(data);

        // Example: Asking user for data and storing it using template class module
        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();
            istringstream iss(data);
            Student s;
            while (iss >> s) {
                cout << s << endl;
            }
            break;
        }
        default:
            cout << "Invalid choice." << endl;
    }

    return 0;
}
```

Output:-



```
"C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II\Practical_15.exe"
Enter Name:Dwij
 Enter Enrollment Number:27
 Enter Course:CS
 Enter Semester:2
 Enter Name:Ender
 Enter Enrollment Number:28
 Enter Course:BTech
 En
ter Semester:3
 Enter Name:Kinf
 Enter Enrollment Number:35
 Enter Course:MTech
 Enter Semester:6
 Enter Name:hello
 Enter Enrollment Number:34
 En
ter Course:df
 Enter Semester:4
 Enter Name:Dwij
 Enter Enrollment Number:54
 Enter Course:hgfj
 Enter Semester:5
 Choose an option to displ
ay data:
```



```
1. Display data from template function module
2. Display data from template class module
1
Dwij 2147483647  6
 2147483647  6
 2147483647  6
 2147483647  6
 2147483647  6
Dwij 27 CS 2
Ender 28 BTech 3
Kinf 35 MTech 6
hello 34 df 4
Dwij 54 hgfj 5


Process finished with exit code 0
```