Coursework-3: OPP in C++

TASK - 2

Submitted by

Name: Ajaj Ahmed

Student ID: 24000864

Cybersecurity and Digital Forensics

Kathmandu, Nepal

April 10, 2025

Task 2:

1. Create a base class `Vehicle` and two derived classes `Car` and `Bike`:
    1. Vehicle has registration number and color
    2. Car adds number of seats
    3. Bike adds engine capacity
    4. Each class should have its own method to write its details to a file
    5. Include proper inheritance and method overriding

```
#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Vehicle

{
protected:

    string registrationNumber;

    string color;

public:

    Vehicle(string regNum, string clr) : registrationNumber(regNum), color(clr)
{}


    virtual void writeToFile()

    {
        ofstream file("vehicle_details.txt", ios::app);

        if (file.is_open())
```

```
        {
            file << "----------------------------" << endl;
            file << "Vehicle Registration Number: " << registrationNumber <<
endl;
            file << "Vehicle Color: " << color << endl;
            file << "----------------------------" << endl;
            file.close();
        }
        else
        {
            cout << "---!!! ERROR!!!--- can't be able to open file!" << endl;
        }
    }

    virtual ~Vehicle() {}
};

class Car : public Vehicle
{
private:

    int numberOfSeats;

public:

    Car(string regNum, string clr, int seats) : Vehicle(regNum, clr),
numberOfSeats(seats) {}

    void writeToFile() override

    {
        ofstream file("vehicle_details.txt", ios::app);


        if (file.is_open())

        {
            file << "----------------------------" << endl;

            file << "Car Details:\n";

            file << "Registration Number: " << registrationNumber << endl;

            file << "Color: " << color << endl;
```

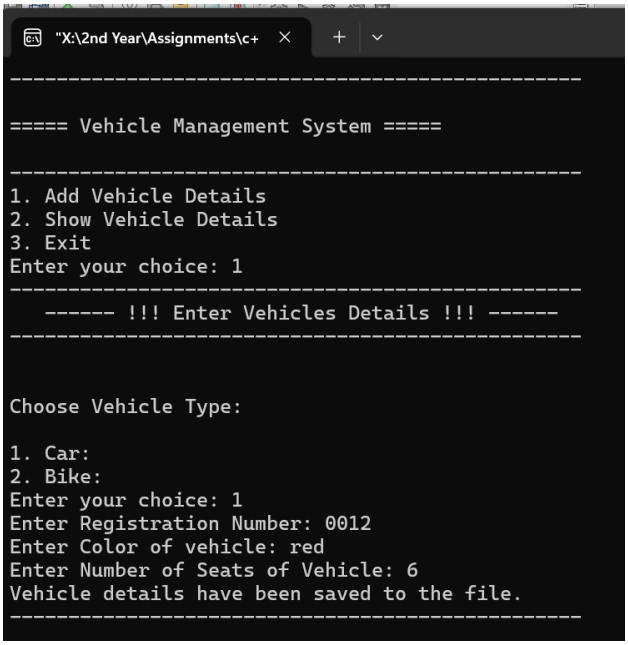
```
            file << "Number of Seats: " << numberOfSeats << endl;

            file << "----------------------------" << endl;

            file.close();
        }

        else

        {

            cout << "---!!! ERROR!!!--- can't be able to open file!" << endl;

        }
    }
};

class Bike : public Vehicle
{

private:

    float engineCapacity;

public:

    Bike(string regNum, string clr, float engineCap) : Vehicle(regNum, clr),
engineCapacity(engineCap) {}


    void writeToFile() override

    {

        ofstream file("vehicle_details.txt", ios::app);

        if (file.is_open())

        {

            file << "----------------------------" << endl;

            file << "Bike Details:\n";

            file << "Registration Number: " << registrationNumber << endl;
```

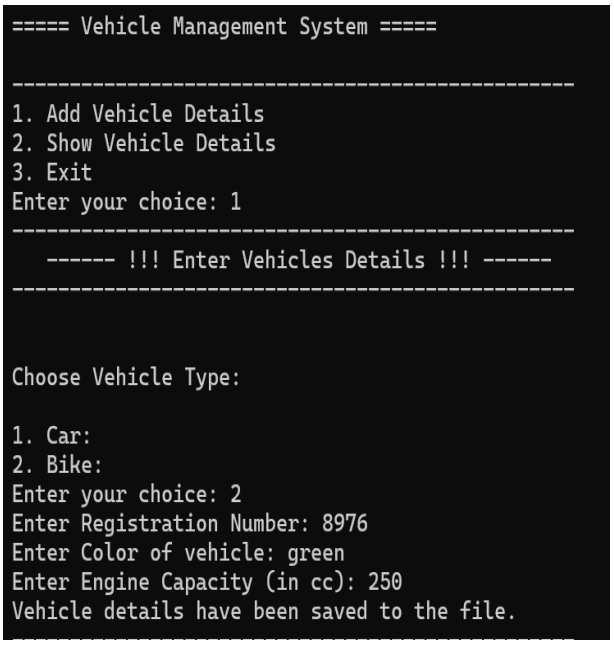
```
            file << "Color: " << color << endl;

            file << "Engine Capacity: " << engineCapacity << " cc" << endl;

            file << "----------------------------" << endl;

            file.close();

        }

        else

            {

                cout << "---!!! ERROR!!!--- can't be able to open file!" << endl;
            }
    }
};

void addVehicleDetails()

{
    int choice;

    cout << "-------------------------------------------------\n";
    cout << "   ------ !!! Enter Vehicles Details !!! ------";
    cout << "-------------------------------------------------\n"<<endl<<endl;

    string regNum, color;

    cout << "Choose Vehicle Type:\n"<<endl;

    cout << "1. Car:   "<<endl;

    cout << "2. Bike:  "<<endl;

    cout << "Enter your choice: ";

    cin >> choice;
    cin.ignore();

    cout << "Enter Registration Number: ";
    getline(cin, regNum);
```

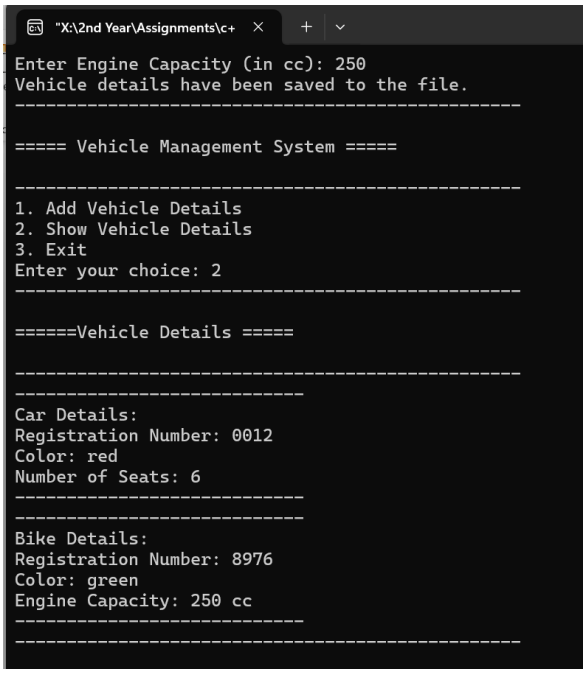
```
    cout << "Enter Color of vehicle: ";
    getline(cin, color);

    if (choice == 1)

    {
        int seats;

        cout << "Enter Number of Seats of Vehicle: ";

        cin >> seats;

        Car newCar(regNum, color, seats);

        newCar.writeToFile();

    }
    else if (choice == 2)
    {
        float engineCapacity;

        cout << "Enter Engine Capacity (in cc): ";

        cin >> engineCapacity;

        Bike newBike(regNum, color, engineCapacity);

        newBike.writeToFile();
    }

    else

    {

        cout << "incorrect  choice! Please enter 1 or 2." << endl;

    }

    cout << "Vehicle details have been saved to the file." << endl;
}

void showVehicleDetails()
{
    ifstream file("vehicle_details.txt");
```

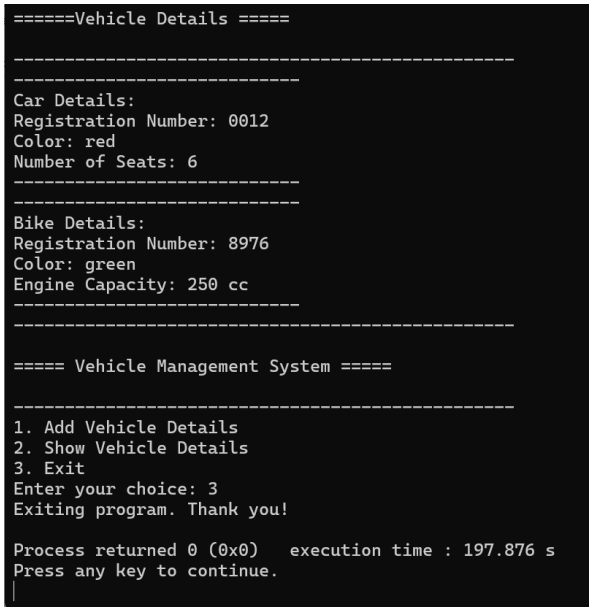
```
    if (file.is_open())
    {
        string line;
        cout << "-------------------------------------------------" << endl;
        cout << "\n======Vehicle Details =====\n" << endl;
        cout << "-------------------------------------------------" << endl;
        while (getline(file, line))
        {
            cout << line << endl;
        }
        file.close();
    }
    else
    {
        cout << "No vehicle records found!" << endl;
    }
}
int main()
{
    int choice;
    while (true)
    {
        cout << "-------------------------------------------------" << endl;
        cout << "\n===== Vehicle Management System =====\n" << endl;
```

```
        cout << "-----------------------------------------------" << endl;

        cout << "1. Add Vehicle Details\n";

        cout << "2. Show Vehicle Details\n";

        cout << "3. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        cin.ignore();

        switch (choice)

        {

        case 1:

            addVehicleDetails();
            break;

        case 2:

            showVehicleDetails();
            break;

        case 3:

            cout << "Exiting program. Thank you!\n";
            return 0;

        default:

            cout << "Invalid choice ! Please make number choice again.\n";
        }
    }
}
```









2. Create a program that:

1. Reads student records (roll, name, marks) from a text file
2. Throws an exception if marks are not between 0 and 100
3. Allows adding new records with proper validation

4. Saves modified records back to file

```
#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <sstream>

#include <iomanip>

using namespace std;

const string subjects[] = {"PYTHON", "C/C++", "JAVA", "FORENSICS", "Ethical"};


class Person

{

public:

    int roll_no;
    string student_name;

    void inputBasicInfo()

    {
        cout << "Enter Roll Number: ";
        cin >> roll_no;

        cin.ignore();

        cout << "Enter the Student Name: ";
        getline(cin, student_name);
    }
};


class Student : public Person

{
```

```
public:

    vector<int> marks;

    void getdata()

    {
        cout << "-----------------------------------------------" << endl;
        cout << "  ------- !!! ADD Student Details!!! -------" << endl;
        cout << "-----------------------------------------------" << endl;

        inputBasicInfo();

        marks.clear();
        cout << "Enter student marks for subjects:\n";
        for (int i = 0; i < 5; i++)

            {
                int student_mark;

                cout << subjects[i] << ": ";

                cin >> student_mark;

            if (student_mark < 0 || student_mark > 100)

                {
                    cout << " ---invalid!!! mark input---. --marks Must be
between 0 and 100.--\n";

                    i--;

                continue;
            }

            marks.push_back(student_mark);
        }
    }

    float average() const

    {
        float total = 0;
```

```
        for (int mark : marks)

            total += mark;

        return total / marks.size();
    }

    char grade() const

    {
        float avg = average();

        if (avg >= 90)

            return 'A';

        else if (avg >= 80)

            return 'B';

        else if (avg >= 70)

            return 'C';

        else if (avg >= 60)
            return 'D';

        else

            return 'F';


    }

    void display() const

    {
        cout << left << setw(20) << roll_no << setw(25) << student_name;

        for (int student_mark : marks)

            cout << setw(10) << student_mark;

        cout << setw(10) << fixed << setprecision(2) << average();
```

```
        cout << setw(6) << grade();
        cout << endl;
    }

    string toString() const

    {
        stringstream ss;

        ss << roll_no << "|" << student_name << "|";

        for (size_t i = 0; i < marks.size(); ++i)

            {
                ss << marks[i];

                if (i < marks.size() - 1) ss << ",";
        }

        return ss.str();
    }

    bool fromString(const string& line)

    {
        stringstream ss(line);

        string roll_noStr, student_nameStr, student_markStr;

        if (!getline(ss, roll_noStr, '|') || !getline(ss, student_nameStr, '|')
|| !getline(ss, student_markStr))

            return false;

        try
            {

                roll_no = stoi(roll_noStr);
            }

        catch (...)

            {
                return false;
            }
```

```
            student_name = student_nameStr;
            marks.clear();

            stringstream ms(student_markStr);
            string m;

        while (getline(ms, m, ',')) {

            try

                {
                    marks.push_back(stoi(m));
                }

            catch (...)

                {

                    return false;

                }
        }

        return marks.size() == 5;
    }
};

void displayAll()

{
    ifstream file("students.txt");

    if (!file)
        {
            cout << "No records found.\n";
            return;
        }

    cout << left << setw(20) << "Roll Number" << setw(25) << "Student Name";

    for (int i = 0; i < 5; ++i)

        cout << setw(10) << subjects[i];
```

```
    cout << setw(10) << "Average" << setw(6) << "Grade";

    cout << "\n" << string(115, '-') << endl;

    string line;
    Student s;

    while (getline(file, line))

        {
            if (s.fromString(line))
                {

                    s.display();
                }
            else

                {

                    cout << "Invalid line: " << line << endl;

                }
    }
    file.close();
}

void addStudent()

{
    Student s;
    s.getdata();

    ofstream file("students.txt", ios::app);

    if (!file)

        {
            cout << "Error writing to file.\n";
            return;
        }
    file << s.toString() << endl;
    file.close();

    cout << "-------------------------------------------------" << endl <<
endl;
```

```
    cout << "----- !!! CoNgRaTuLaTiOnS ReCoRd AdDeD !!! -----" << endl;
    cout << "Student information has been added successfully.\n";
    cout << "-------------------------------------------------------" << endl <<
endl;
}

void updateStudent()

{
    ifstream file("students.txt");

    if (!file)
        {
            cout << "No records found.\n";
            return;
        }

    vector<Student> all;
    string line;
    Student s;
    while (getline(file, line))
        {
            if (s.fromString(line))

            {

                all.push_back(s);
            }
    }
    file.close();

    int roll;


    cout << "-----------------------------------------------" << endl;
    cout << "               !!! STUDENT ReCoRd !!!       " << endl;
    cout << "    ------ !!! UPDATE STUDENT DATA!!! ------" << endl;
    cout << "-----------------------------------------------" << endl;
    cout << "Enter Roll Number to update student data: ";
    cin >> roll;

    bool found = false;

    for (auto& stu : all)
```

```
        {
            if (stu.roll_no == roll)

            {
                cout << "Enter new marks for student " << stu.student_name <<
":\n";
                stu.marks.clear();
            for (int i = 0; i < 5; ++i)
                {
                    int mark;

                    cout << subjects[i] << ": ";
                    cin >> mark;

                    stu.marks.push_back(mark);
                }

            found = true;
            break;
        }
    }

    if (!found)

        {
            cout << "Student not found.\n";
            return;
        }

    ofstream out("students.txt");

    for (const auto& stu : all)
        {
            out << stu.toString() << endl;
        }

    out.close();

    cout << "----------------------------------------" << endl << endl;
    cout << "------Congratulations------" << endl << endl;
    cout << "-----Record updated.-----\n";
    cout << "----------------------------------------" << endl << endl;
}

int main()
```

```
{
    int choice;

    cout << "-----------------------------------------------------------------" <<
endl;
    cout << "     ------ !!! WELCOME TO Management System !!! -----" << endl;
    cout << "-----------------------------------------------------------------" <<
endl << endl;

    while (true)

        {
            cout << "1. View Students\n";
            cout << "2. Add Student\n";
            cout << "3. Update Student\n";
            cout << "4. Exit\n";
            cout << "----------------------------------------" << endl ;
            cout << "Choice: ";

            cin >> choice;
            cin.ignore();

        switch (choice)

        {
            case 1:

                displayAll();
                break;

            case 2:

                addStudent();
                break;

            case 3:

                updateStudent();
                break;

            case 4:

                cout << "____ --- !!! Thank you For visiting !!!---___!\n";
                return 0;
```

```
            default:

                cout << "please !!! Enter correct choice.\n";
        }
    }
}
```

```
1. View Students
2. Add Student
3. Update Student
4. Exit
----------------------------------------
Choice: 1
Roll Number         Student Name            PYTHON    C/C++     JAVA      FORENSICS Ethical   Average   Grade
-------------------------------------------------------------------------------------------------------------
12                  ram                     78        89        98        77        67        81.80     B
9087                Hari Kumar              88        77        98        87        66        83.20     B
0                                           17        17        17        17        17        17.00     F
987                 Mohan Prasad            98        78        88        87        66        83.40     B
1. View Students
2. Add Student
3. Update Student
4. Exit
```

```
"X:\2nd Year\Assignments\c+
------------------------------------------------------
     ------ !!! WELCOME TO Management System !!! -----
------------------------------------------------------

1. View Students
2. Add Student
3. Update Student
4. Exit
--------------------------------------
Choice: 2
------------------------------------------
  ------- !!! ADD Student DEtails!!! -------
------------------------------------------
Enter Roll Number: 00987
Enter the Student Name: Mohan Prasad
Enter student marks for  subjects:
PYTHON: 98
C/C++: 78
JAVA: 88
FORENSICS: 87
Ethical: 66
--------------------------------------------------

----- !!! CoNgRaTuLaTiOnS ReCoRd AdDeD !!! -----
Student information has been added successfully.
--------------------------------------------------
```

```
987                 Mohan Prasad            98
1. View Students
2. Add Student
3. Update Student
4. Exit
----------------------------------------
Choice: 3
------------------------------------------------
               !!! STUDENT ReCoRd !!!
    ------ !!! UPDATE STUDENT DATA!!! ------
------------------------------------------------
Enter Roll Number to update student data: 0
Enter new marks for student :
PYTHON: 98
C/C++: 88
JAVA: 78
FORENSICS: 66
Ethical: 98
----------------------------------------

------Congratulations------

-----Record updated.-----
----------------------------------------
```

```
1. View Students
2. Add Student
3. Update Student
4. Exit
---------------------------------------
Choice: 1
Roll Number         Student Name             PYTHON    C/C++     JAVA      FORENSICS Ethical   Average   Grade
-------------------------------------------------------------------------------------------------------------------
12                  ram                      78        89        98        77        67        81.80     B
9087                Hari Kumar               88        77        98        87        66        83.20     B
0                                            98        88        78        66        98        85.60     B
987                 Mohan Prasad             98        78        88        87        66        83.40     B
1. View Students
2. Add Student
3. Update Student
4. Exit
---------------------------------------
Choice:
```