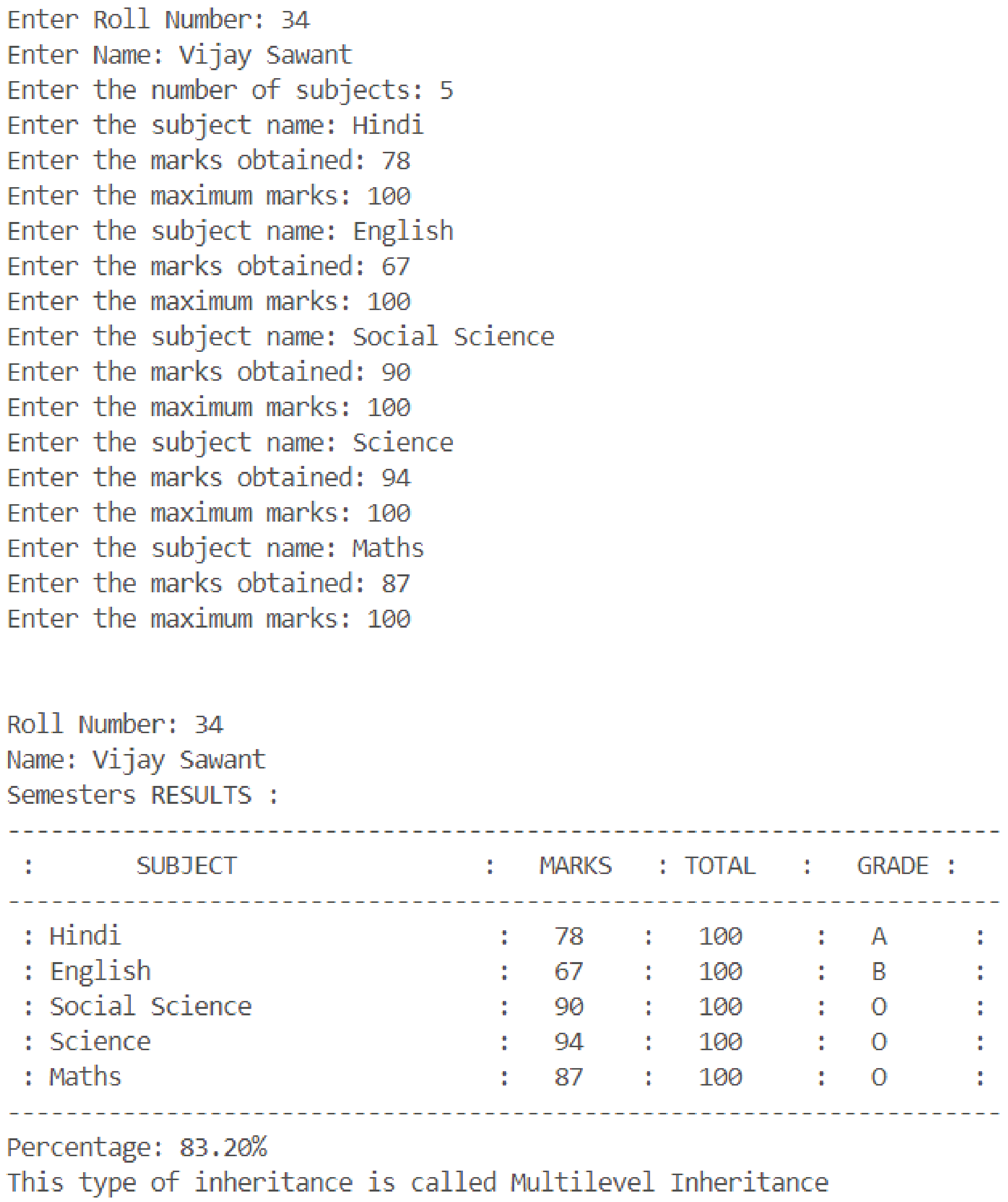
OUTPUT



```
Enter Roll Number: 34
Enter Name: Vijay Sawant
Enter the number of subjects: 5
Enter the subject name: Hindi
Enter the marks obtained: 78
Enter the maximum marks: 100
Enter the subject name: English
Enter the marks obtained: 67
Enter the maximum marks: 100
Enter the subject name: Social Science
Enter the marks obtained: 90
Enter the maximum marks: 100
Enter the subject name: Science
Enter the marks obtained: 94
Enter the maximum marks: 100
Enter the subject name: Maths
Enter the marks obtained: 87
Enter the maximum marks: 100


Roll Number: 34
Name: Vijay Sawant
Semesters RESULTS :
---------------------------------------------------------------------
 :       SUBJECT                 :   MARKS   : TOTAL   :   GRADE :
---------------------------------------------------------------------
 : Hindi                          :   78    :   100     :   A      :
 : English                        :   67    :   100     :   B      :
 : Social Science                 :   90    :   100     :   O      :
 : Science                        :   94    :   100     :   O      :
 : Maths                          :   87    :   100     :   O      :
---------------------------------------------------------------------
Percentage: 83.20%
This type of inheritance is called Multilevel Inheritance
```

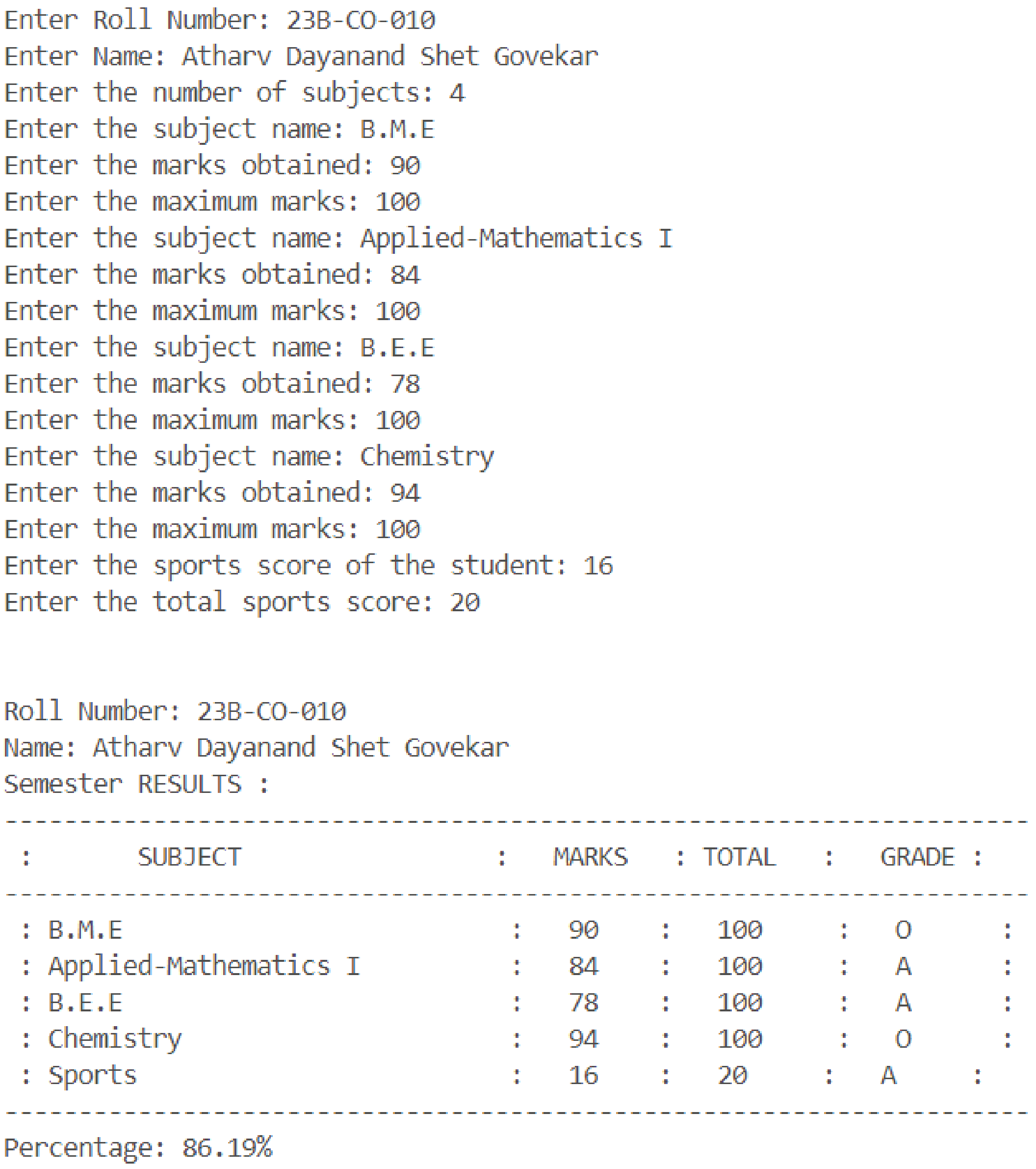
D]b]"A new scheme for evaluation of students' performance is formulated that gives also weightage for sports. Extend the inheritance relation discussed in the above program (D) such that the Result class also inherits properties of the Sports class. Note that the Sports class is a derived class of the Student class. Write a program to model this relationship such that members of the Student class are not inherited twice. What type of inheritance does this model belong to?"

PROGRAM –

```
#include <iostream>
#include <iomanip>
#include <cstring>
using namespace std;

class Student {
protected:
    char *roll_number;
    char name[50];
public:
    Student() {
        roll_number = nullptr;
        name[0] = '\0';
    }
    void input(char *r, const char *n) {
        delete[] roll_number;
        roll_number = new char[strlen(r) + 1];
        strcpy(roll_number, r);
        strcpy(name, n);
    }
};

class test : virtual public Student {
protected:
    int n;
    char **subject;
    int *marks;
    int *max_marks;
public:
    test() {
        n = 0;
        subject = NULL;
        marks = NULL;
        max_marks = NULL;
    }
    ~test() {
        if (subject != NULL) {
            for (int i = 0; i < n; i++) {
                delete[] subject[i];
            }
            delete[] subject;
        }
        delete[] marks;
        delete[] max_marks;
    }
    void getinfo() {
        cout << "Enter the number of subjects: ";
        cin >> n;

        delete[] subject;
        delete[] marks;
        delete[] max_marks;

        subject = new char*[n];
        marks = new int[n];
        max_marks = new int[n];

        cin.ignore();

        for (int i = 0; i < n; i++) {
          subject[i] = new char[50];
          cout << "Enter the subject name: ";
          cin.getline(subject[i], 50);
          cout << "Enter the marks obtained: ";
          cin >> marks[i];
          cout << "Enter the maximum marks: ";
          cin >> max_marks[i];
          cin.ignore();
        }
    }
};

class sports : virtual public Student {
protected:
    string eval = "Sports";
    int score;
    int total;
public:
    sports() {
        score = 0;
        total = 0;
    }
    void getdata() {
        cout << "Enter the sports score of the student: ";
        cin >> score;
        cout << "Enter the total sports score: ";
        cin >> total;
    }
    void display() {
        cout << "The score of the student in " << eval << " is: " << score << endl;
        cout << "The total score in " << eval << " is: " << total << endl;
    }
};

class result : public test, public sports {
protected:
    int total_marks = 0;
    int total_max_marks = 0;
    float percentage;
    int sports_grade;
public:
    result() : Student() {
        total_marks = 0;
        total_max_marks = 0;
        percentage = 0;
        sports_grade = 0;
    }

    void calculateResult() {
        for (int i = 0; i < n; i++) {
            total_marks += marks[i];
            total_max_marks += max_marks[i];
        }
        total_marks += score;
        total_max_marks += total;
        if (total_max_marks > 0) {
            percentage = ((float)total_marks / total_max_marks) * 100;
        } else {
            percentage = 0;
        }
    }
```

```
    int calc(int marks) {

        if (marks <= 100 && marks >= 85) return 10;

        else if (marks < 85 && marks >= 75) return 9;

        else if (marks < 75 && marks >= 65) return 8;

        else if (marks < 65 && marks >= 55) return 7;

        else if (marks < 55 && marks >= 50) return 6;

        else if (marks < 50 && marks >= 40) return 5;

        else return 0;

    }


    char grap(int grade) {

        if (grade == 10) return 'O';

        else if (grade == 9) return 'A';

        else if (grade == 8) return 'B';

        else if (grade == 7) return 'C';

        else if (grade == 6) return 'D';

        else if (grade == 5) return 'E';

        else return 'F';

    }

    void display() {

        cout << "\n\nRoll Number: " << roll_number <<
endl;

        cout << "Name: " << name << endl;

        cout << "Semester RESULTS :\n";

        cout << "--------------------------------------------------------
------------\n";

        cout << " :      SUBJECT              :  MARKS   :
TOTAL   :   GRADE :\n";

        cout << "--------------------------------------------------------
------------\n";


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

            int mar = (marks[i] * 100) / max_marks[i];

            int grade = calc(mar);

            char grade_letter = grap(grade);

            cout << " : " << left << setw(30) << subject[i] << "
:   "

                << fixed << setprecision(1) << marks[i] <<
"    :   "

                << max_marks[i] << "     :   "

                << setw(2) << grade_letter << "     :\n";

        }


        int sports_max = (score * 100) / total;

        char sports_grade_letter = grap(calc(sports_max));

        cout << " : " << left << setw(30) << eval << " :    "

            << fixed << setprecision(1) << score << "    :   "

            << total << "     :   "<< sports_grade_letter <<
"    :\n";

        cout << "--------------------------------------------------------
------------\n";

        cout << "Percentage: " << fixed << setprecision(2) <<
percentage << "%" << endl;

    }

};

int main() {

//This type of inheritance is called HYBRID inheritance

    char roll[50];

    char name[50];

    cout << "Enter Roll Number: ";

    cin.getline(roll, 50);

    cout << "Enter Name: ";

    cin.getline(name, 50);

    result r;

    r.input(roll, name);

    r.getinfo();

    r.getdata();

    r.calculateResult();

    r.display();


    return 0;}
```

OUTPUT



```
Enter Roll Number: 23B-CO-010
Enter Name: Atharv Dayanand Shet Govekar
Enter the number of subjects: 4
Enter the subject name: B.M.E
Enter the marks obtained: 90
Enter the maximum marks: 100
Enter the subject name: Applied-Mathematics I
Enter the marks obtained: 84
Enter the maximum marks: 100
Enter the subject name: B.E.E
Enter the marks obtained: 78
Enter the maximum marks: 100
Enter the subject name: Chemistry
Enter the marks obtained: 94
Enter the maximum marks: 100
Enter the sports score of the student: 16
Enter the total sports score: 20


Roll Number: 23B-CO-010
Name: Atharv Dayanand Shet Govekar
Semester RESULTS :
---------------------------------------------------------------------
 :       SUBJECT                 :   MARKS   : TOTAL   :   GRADE :
---------------------------------------------------------------------
 : B.M.E                          :   90    :   100     :   O      :
 : Applied-Mathematics I          :   84    :   100     :   A      :
 : B.E.E                          :   78    :   100     :   A      :
 : Chemistry                      :   94    :   100     :   O      :
 : Sports                         :   16    :   20     :   A     :
---------------------------------------------------------------------
Percentage: 86.19%
```