ENGINEERING COLLEGE

AIM:

To create a student class and store information and showcase the inheritance concept.

ALGORITHM:

· Create a base class Student with ID, name, and marks.

· Add a function to store student details.

· Add a function to search and display student details by ID.

· Create UnderGraduate and PostGraduate classes from Student.

· Add department details in derived classes.

· Override the search function to display program and department.

· Create student objects and store their addresses in a base class pointer array.

· Read student ID from the user.

· Search the student using polymorphism and display details.

· Print “Student not found” if ID does not match.

PROGRAM:

```
/*
*  Program to create a Student database using Class Inheritance
*  Author  : MUTHUGANESH S
*  Date    : 29/1/2026
*  Filename: EngineeringClg.cpp
*/

#include <iostream>
using namespace std;


class Student {

private:
    int StudentID;
    string Name;
    int Marks[5];

public:

    // Base method to add common student data
    virtual void AddData(int Id, string name, int mark[]) {
```

```
        StudentID = Id;
        Name = name;

        for (int i = 0; i < 5; i++) {
            Marks[i] = mark[i];
        }
    }

    // Virtual display/search method
    virtual bool Search(int Id) {

        if (StudentID == Id) {

            cout << "Student ID : " << StudentID << endl;
            cout << "Name       : " << Name << endl;
            cout << "Marks      : ";

            for (int i = 0; i < 5; i++) {
                cout << Marks[i] << " ";
            }

            cout << endl;
            return true;
        }

        return false;
    }
};


class UnderGraduate : public Student {

    string Department;

public:
    void AddData(int Id, string name, int mark[], string Dept) {
            Student::AddData(Id, name, mark);
        Department = Dept;
    }

    bool Search(int Id) override {

        if (Student::Search(Id)== true) {

            cout << "Program    : Under Graduate" << endl;
            cout << "Department : " << Department << endl;
            return true;
        }

        return false;
    }
};


class PostGraduate : public Student {
```

```
    string Department;

public:
    void AddData(int Id, string name, int mark[], string Dept) {

        Student::AddData(Id, name, mark);
        Department = Dept;
    }

    bool Search(int Id) override {

        if (Student::Search(Id)== true) {

            cout << "Program    : Post Graduate" << endl;
            cout << "Department : " << Department << endl;
            return true;
        }

        return false;
    }
};

int main() {

    UnderGraduate ug1, ug2;
    PostGraduate pg1, pg2;

    int marks1[5] = {85, 90, 78, 92, 88};
    int marks2[5] = {75, 80, 68, 82, 78};
    int marks3[5] = {88, 92, 84, 90, 86};
    int marks4[5] = {95, 98, 94, 96, 92};

    ug1.AddData(1, "Alice", marks1, "Computer Science");
    ug2.AddData(2, "Bob", marks2, "Mechanical Engineering");
    pg1.AddData(3, "Charlie", marks3, "Electrical Engineering");
    pg2.AddData(4, "David", marks4, "Civil Engineering");

    Student* students[4];
    students[0] = &ug1;
    students[1] = &ug2;
    students[2] = &pg1;
    students[3] = &pg2;

    int searchId;
    cout << "Enter Student ID to search: ";
    cin >> searchId;

    bool found = false;
    for (int i = 0; i < 4; i++) {
        if (students[i]->Search(searchId)) {
            found = true;
            break;
        }
```

```
    }

    if (!found) {
        cout << "Student not found." << endl;
    }

    return 0;
}
```

OUTPUT: