ASSIGNMENT 2

```
// University System
#include <iostream>
#include <string>
#include <map>
#include <vector>
using namespace std;
class Person {
private:
  string name, id, contact;
  int age;
public:
  Person(string n = "", int a = 0, string i = "", string c = "") {
    setName(n);
    setAge(a);
    id = i;
    contact = c;
  }
  virtual ~Person() {}
  void setName(string n) { if (!n.empty()) name = n; }
  void setAge(int a) { if (a > 0 && a < 120) age = a; }
  void setID(string i) { id = i; }
  void setContact(string c) { contact = c; }
  string getName() { return name; }
  int getAge() { return age; }
  string getID() { return id; }
  string getContact() { return contact; }
  virtual void displayDetails() {
    cout << "Name: " << name << ", Age: " << age << ", ID: " << id << ", Contact: " <<
contact << endl;
  }
  virtual float calculatePayment() { return 0.0; }
};
class Student : public Person {
private:
  string enrollmentDate, program;
  float gpa;
public:
  Student(string n, int a, string i, string c, string e, string p, float g) : Person(n, a, i, c) {
    enrollmentDate = e;
    program = p;
    setGPA(g);
  }
  void setGPA(float g) { if (g >= 0.0 && g <= 4.0) gpa = g; }
  float getGPA() { return gpa; }
  void displayDetails() override {
    Person::displayDetails();
    cout << "Program: " << program << ", GPA: " << gpa << endl;
  }
  float calculatePayment() override { return 18000.0; }
};
class Professor : public Person {
private:
  string department, specialization, hireDate;
public:
  Professor(string n, int a, string i, string c, string d, string s, string h) : Person(n, a, i, c) {
    department = d;
    specialization = s;
    hireDate = h;
  }
  void displayDetails() override {
    Person::displayDetails();
    cout << "Department: " << department << ", Specialization: " << specialization <<
endl;
  }
  float calculatePayment() override { return 45000.0; }
};
class Course {
private:
  string code, title, description;
  int credits;
public:
  Course(string c, string t, string d, int cr) {
    code = c; title = t; description = d;
    setCredits(cr);
  }
  void setCredits(int cr) { if (cr > 0) credits = cr; }
};
class Department {
private:
  string name, location;
  double budget;
public:
  Department(string n, string l, double b) {
    name = n; location = l; budget = b;
  }
};
class GradeBook {
private:
  map<string, float> grades;
public:
    void addGrade(string studentID, float grade) {
        grades[studentID] = grade;
    }
    float calculateAverageGrade() {
        float total = 0;
        for (auto g : grades) total += g.second;
        return grades.empty() ? 0 : total / grades.size();
    }
    vector<string> getFailingStudents() {
        vector<string> fail;
        for (auto g : grades)
            if (g.second < 40) fail.push_back(g.first);
        return fail;
    }
};
class EnrollmentManager {
private:
    map<string, vector<string>> enrollments;
public:
    void enrollStudent(string course, string studentID) {
        enrollments[course].push_back(studentID);
    }
    void dropStudent(string course, string studentID) {
        auto &list = enrollments[course];
        list.erase(remove(list.begin(), list.end(), studentID), list.end());
    }
    int getEnrollmentCount(string course) {
        return enrollments[course].size();
    }
};
void showDetails(Person* p) {
    p->displayDetails();
    cout << "Payment: " << p->calculatePayment() << endl;
}
int main() {
    Student stu1("kamal", 19, "ST101", "99999", "2024-01-01", "IT", 3.5);
    Student stu2("Dhruv", 20, "ST102", "88888", "2024-01-01", "ME", 3.0);
    Professor prof1(" Karan", 40, "PF201", "44444", "EE", "Signals", "2012-03-10");
    Professor prof2("Garv", 48, "PF202", "77777", "CE", "Structures", "2008-06-01");
    showDetails(&stu1);
    showDetails(&stu2);
    showDetails(&prof1);
    showDetails(&prof2);
    return 0;
}
```