Project documentation:

1. Main file: This main code represents a program that manages student and course information. It begins by prompting the user to enter the number of students, courses, and CPG (Computer Programming) students. It then dynamically allocates memory for arrays of `CStudent`, `CCourse`, and `CPG\_Student` objects based on the input.

Next, it enters a loop to register each student by calling the `registerStudent()` function and calculate their GPA using the `claculateGPA()` function. Another loop adds courses using the `AddCourse()` function. Afterward, it displays the information of each student using the `getStudentInfo()` function and prints the details of each course using the `getCourseInfo()` function.

In the bonus section, it manages additional information for CPG students. It registers CPG students, prompts the user to enter their job title, and sets the job title using the `setPGStudentJobTitle()` function. It then calculates the GPA of each CPG student, displays their information, and prints their job title using the `getPGStudentJobTitle()` function.

Finally, it deallocates the memory used by the dynamic arrays using the `delete[]` operator and returns 0 to end the program.

1. Student.cpp: The student.cpp code defines the implementation of the `CStudent` class and includes a bonus class called `CPG\_Student`. The `CStudent` class represents a student and has member functions to set and get various student attributes such as name, ID, email, major, grades, score, and password. It also includes functions to register a student by prompting the user to enter their information and to calculate the student's GPA based on their grades.

The `CPG\_Student` class is a bonus class that extends the `CStudent` class. It adds an additional member variable called `pg\_student\_job\_title` and provides functions to set and get the job title for a CPG student. The `setPGStudentJobTitle()` function uses `strncpy()` to safely copy the provided job title into the `pg\_student\_job\_title` member variable.

this code provides a basic implementation of student management, allowing users to register students, calculate their GPA, and handle additional information for CPG students such as job titles.

1. Student.hpp: The student.hpp code defines a header file named "student.hpp" that contains the class declarations for `CStudent` and `CPG\_Student`. The `CStudent` class represents a student and has private member variables for attributes such as name, ID, email, major, grades, score, and password. It includes public member functions to set and get these attributes, as well as functions for registering a student, displaying their information, and calculating their GPA.

The `CPG\_Student` class is a derived class that inherits from `CStudent`. It adds an additional private member variable called `pg\_student\_job\_title` and provides functions to set and get the job title for a CPG (Computer Programming) student.

The header file also includes necessary standard library headers and uses the `using namespace std` directive for convenience.

this code provides the class declarations for student management, allowing users to define and manipulate student objects, as well as handle additional information specific to CPG students.

1. Ccourse.cpp: The ccourse code defines the implementation of the `CCourse` class, which represents a course. It includes member functions to set and get the course name, course code, and course cost. The `setCourseName()`, `setCourseCode()`, and `setCourseCost()` functions set the corresponding member variables to the provided values, while the `getCourseName()`, `getCourseCode()`, and `getCourseCost()` functions return the values of these member variables.

Additionally, the code includes functions `AddCourse()` and `getCourseInfo()` to register a course and display its information, respectively. The `AddCourse()` function prompts the user to enter the course name, code, and cost, and stores these values in the corresponding member variables. The `getCourseInfo()` function retrieves the course information using the getter functions and prints it to the console.

The code includes necessary standard library headers and uses the `using namespace std` directive for convenience. this code provides a basic implementation of a course management system, allowing users to create course objects, store their information, and retrieve and display course details.

1. Ccourse.hpp: The Ccourse.hpp code defines a header file named "CCourse.hpp" that contains the class declaration for the `CCourse` class. The `CCourse` class represents a course and has private member variables for attributes such as course name, course code, and course cost. It includes public member functions to set and get these attributes, as well as functions to add a course and retrieve its information.

The header file also includes necessary standard library headers and uses the `using namespace std` directive for convenience.this code provides the class declaration for managing courses, allowing users to define and manipulate course objects, set their attributes, and retrieve course information.