A **class CStudent** is added to a project for an application called **ResultsApp** that is to be used to manage the exam results for a group of students for a particular subject.

A student will be awarded marks for both continuous assessment and for a final written exam. A maximum mark of 50 applies to each element. To pass the subject, a student must obtain a minimum total mark of 40 and must also obtain at least 15 marks in each element.

If a student passes the subject he is then awarded a grade based on the following table

|  |  |
| --- | --- |
| **Total Mark** | **Grade String** |
| >= 70 | H1 |
| 60 to < 70 | H2.1 |
| 50 to < 60 | H2.2 |
| 40 to < 50 | Pass |

The code below shows a definition of the class.

**class CStudent**

**{**

**private:**

**string m\_Name;**

**int m\_Assessment;**

**int m\_Exam;**

**public:**

**CStudent (void);**

**CStudent (string name);**

**CStudent (string name, int assessment, int exam);**

**void SetAssessment(int mark);**

**void SetExam(int mark);**

**string GetName(void);**

**int GetAssessment(void);**

**int GetExam(void);**

**bool HasName(string name);**

**bool IsAPass(void);**

**string GetGrade(void);**

**};**

Write definitions for all the member functions of the class

**a)** The first constructor

**CStudent (void)**

should initialize the name data member to an empty string and the

other data members to zero.

1. The second constructor

**CStudent (string name)**

should initialize the appropriate data member to the value passed as an argument and the other data members to zero.

1. The third constructor

**CStudent (string name, int assessment, int exam)**

should initialize the appropriate data members to the values passed as arguments.

1. The **Set** functions should insert the mark for the particular element.

1. The **Get** functions should return the appropriate data member.

1. The function **HasName** checksfor a matching name in the object.
2. The function **IsAPass** should return true only if each element is greater than or equal to 15 and the total mark is greater than or equal to 40.
3. The function **GetGrade** will return the appropriate grade string if the student has passed the subject. If a student has failed it will return an empty string.

The application **ResultsApp** mentioned in above is designed for a group of up to 40 students and will instantiate a global collection of **class CStudent** objects and a global variable to contain the actual number of students as per the following code

**CStudent list[40];**

**int numStudents = 0;**

The global data will be made persistent by reading from and writing to a file each time the application is opened and closed.

Each time a new student is added to the system the **numStudents** variable will be incremented.

The application will display a menu driven interface that will allow the user to access this global collection. Menu options and associated callback functions will allow a user to :

1. initialize the group of students by entering the name of each student. This should cause a **class CStudent** object to be instantiated for each student and for the object to be inserted in the collection.

The callback function has a prototype **void DoInitializeGroup(void).**

Write a definition for this function.

1. select an element (either assessment or exam) and then insert the marks for each student in the group.

The callback function has a prototype **void DoEnterElementMarks(void)**

Write a definition for this function.

1. display a list of the names and grades of all the students who have passed the exam.

The callback function has a prototype **void DoDisplayPassList(void)**

Write a definition for this function.