**Student Athlete**

**Note:** ALL input must be validated (except string input for name and sport) and all output must be formatted.

You will be writing one program as described below:

**Student** class with private data for first name, last name, grade (9-12), and total fees. Include a non-default constructor that requires first name, last name, and grade. Your constructor must also initialize total fees to 0. You may NOT use a default constructor!

Overload the following operators:

+= This will add a fee to a student’s total fee.

<< This will output a student’s grade, full name and total fees.

== This will compare a last name to a student’s last name and return a Boolean value.

== This will compare a grade to a student’s grade and return a Boolean value.

**Athlete** class with private data for first name, last name, name of sport played and a static field for athletic fee. Currently the athletic fee is $125.50. Include a non-default constructor that requires first name, last name, and name of sport. You may NOT use a default constructor!

Overload the following operators:

<< This will output a student’s sport and full name.

== This will compare a sport to a student’s sport and return a Boolean value.

**main():**

Your main() function will create two vectors, one for Student and one for Athlete. Your program will call the displayMenu() function and based on which option the user chooses, call the appropriate function. Allow the user to continue choosing various options until they decide to quit.

Your program will contain a series of functions (non-member) as listed below:

**displayMenu():**

Will display a menu and returns the users choice. Your menu will contain the following options:

Please choose from the following menu

1: Add student

2: Print a list of enrolled students

3: Print a list of students by last name

4: Print a list of students who play sports

5: Print a list of students by grade

6: Print a list of students by sport

7: Exit program

**addStudent():**

This function will accept both vectors as parameters. Your function will ask the user for the students first name, last name, and grade. Then it will ask whether or not the student plays a sport. If the student does play a sport, then it will ask for the name of the sport and add that student to the athletic vector. You will use your overloaded += sign to add the athletic fee (must use static variable) to your student. You will continue to ask for that students sports (meaning they could play more than one, so they may have multiple entries in the athlete vector). Once you are finished with that student, you will add that student to the student vector.

**displayStudents():**

This function will accept your student vector and display the name, grade and total fee for each student. The output will use your overloaded insertion operator.

**displayOneStudent():**

This function will accept your student vector and display the name, grade and total fee for one student. You will ask the user for the last name only (we will not be comparing first names and we will assume that there are no duplicates) and use your overloaded == operator to compare the last name to your student object. You will also use your overloaded insertion operator as you did in the displayStudents() function.

**displayAthletes():**

This function will accept your athlete vector and display the full name and sport played for each student. The output will use your overloaded insertion operator.

**displayStudentsByGrade():**

This function will accept your student vector and display the full name, grade, and total fee for all students in a given grade. You will have to ask the user which grade they would like to see displayed. Use your overloaded == operator and your overloaded insertion operator as you did in the displayStudents() function.

**displayStudentsBySport():**

This function will accept your athlete vector and display the full name and sport for all students playing a given sport. You will have to ask the user which sport they would like to see displayed. Use your overloaded == operator and your overloaded insertion operator as you did in the displayAthletes() function.

**Lastly: You’ve made it this far …. Why not try?**

Include an option to display all students in a particular grade for a particular sport. For example, I might want to see all 9th grade students who play football.