Lab Homework #2: The Student Collection Object

In this homework assignment, we are going to define a student class and use a collection of student objects. By using a vector collection variable (also implemented as a class), we are going to implement a number of operations. You are given a template program file that you can extend to complete the requirements of the assignment. Basically, we are going to make use of the following class in order to keep a record of a list of students.

```
// student class
class Student
{
  private:
    unsigned id;  // positive value
    string name;  // name with space allowed
  public:
    void setName(string n)
    { name = n;
    }

    string getName()
    { return name;
    }

    void setID(int id_param)
    { id = id_param;
    }

    int getID()
    { return id;
    }

    // creating a single string value out of
    // all the fields of the student object
    string string_format()
    { string result = to_string(id) +" -- " + name;
      return result;
    }

}; // student class ends.
```

We want to incorporate the student class to create new student objects and store them in a student collection variable. The student collection is also a class that is comprised of a set of functions. A vector of student objects in the member variable of the StudentCollection class. Further, by using the vector variable we want to be able to do a set of tasks. These tasks will be implemented inside the StudentCollection class.

The declaration of the StudentCollection class is shown in the following:

```
class StudentCollection
{
  private:
    vector<Student> student_records; // holds all student records

  public:
  // read the file and load the values to create student objects
  // the objects are added in the student records vector
  void load_all_records();

  // this function will input values for a new movie record
  // and will return that
  void add_new_record();

  // search the student records based on title
  int find_student_record(unsigned search_id);

  // if there is a student with delete_id in the record then
  // that record would be delete. Otherwise, nothing happens.
  void delete_record(int delete_id);

  // view all student records
  void view_all_records();

  // save all the record in the vector to a file
  void save_all_records();

};
```

Use the given template file and complete the functions of the StudentCollection class in order to achieve the following tasks (inside the code, we have provided written comments to describe the steps you should do to complete the functions):

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program

In the main function, we have created an object of the StudentCollection class. Depending on the menu choices, we are going to call appropriate functions from this class. The menu choices are displayed inside a do-while loop. As long as the user is not using the exit option, the loop will continue to provide the user options to interact with the StudentCollection object.

When the user selects the option 0, the function add_new_record() of the object StudentCollection will be called. In this function, first we should input the id and name values. Further, we should create an object of the student class and use the setID and setName functions to initialize the values for that object. Lastly, the object should be added in the student_records vector variable.

The option 2 is self explanatory and the code for this option has been provided. The find_student_record function takes an unsigned ID number and returns the index value of the record in the student_records vector variable. If no such records exist then a negative index value is returned. In the menu option 4, the index value is further used to delete the record from the student_records vector variable by using the erase method. However, we have to complete the code based on the return value of the find_student_record function.

In order to display a student record, the string_format function of the student class has been provided in the template file for your reference. This function will combine all of the fields of a student class and create a single string value and return it to the calling function. Please study the overall architecture of the program and the menu based interaction system. You can use this system in solving your future assignments.

This program also employs file based record save option. When the program first starts, it should read the content of the file, create student objects based on the read values, and add the student objects in the student_records. You can implement the load_all_records function in order to complete this task (we should use the ifstream object to read values from the file). When the main function starts, this function is called in the main.

Similarly, in the menu option we can choose to store all the current records in the file. For this menu option we can write all the codes in the save_all_records function. In this function, the program will loop through all the objects in the student_records variable and then write them in the file by using ofstream object.

For this lab, you are allowed to work with one partner in a team of two if you wish. If so, you must agree to do all of the work on this lab together, and you must agree that you will contribute equally to the submission. If you are working as a group then email your professor beforehand and get the confirmation. Further, add a comment at the top of the source program file and list both of your full names. Each student should individually submit their work on the Blackboard.

When you are satisfied with your program, by the due date of 5 pm Friday, 5 February, submit it on the Blackboard. Thank you.

NB: If your implementation is complete then it should conform to the sample input/output interaction scenario shown in the following. Please note that the purple texts in the following are user input to the program.

Sample Interaction Scenario

```
0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 1


We have the following student records:
200 -- Tony Sanders

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 0

Student ID: 300
Student Name: Kafi Rahman


0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 0

Student ID: 500
Student Name: Sabina Richard


0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 1
```

```
We have the following student records:
200 -- Tony Sanders
300 -- Kafi Rahman
500 -- Sabina Richard

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 2

Enter the ID that you want to search: 500

The record has been found: 500 -- Sabina Richard

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 4

Success! all records have been saved.

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 3

Enter the ID that you want to delete: 500

The record has been found: 500 -- Sabina Richard
the record has been deleted

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 1
```

```
We have the following student records:
200 -- Tony Sanders
300 -- Kafi Rahman

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 3

Enter the ID that you want to delete: 200

The record has been found: 200 -- Tony Sanders
the record has been deleted

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 1


We have the following student records:
300 -- Kafi Rahman

0. Add a new student record
1. View all student records
2. Find a record by its ID number
3. Delete a student record
4. Save all student records in file
5. Exit the program
Your choice: 5

Thanks for using the program
```