

San José State University
Department of Computer Engineering

CMPE 180-92
Data Structures and Algorithms in C++
Spring 2017

Instructor: Ron Mak

Assignment #6

Assigned: Thursday, March 2
Due: Thursday, March 9 at 5:30 PM
CodeCheck: <http://codecheck.it/codecheck/files/1703040929alwnb99v81mqrg0uk694e8o8e>
Canvas: Assignment 6. Employee Records
Points: 100

Employee records

This assignment will give you practice manipulating a vector of objects and reading a text file of comma-separated values (CSV). CSV files are typically written from spreadsheets.

Implement a C++ class **Employee**. Create separate **.h** and **.cpp** files for the class, and another **.cpp** file for the main program.

Commands

Your program will read a text file containing commands, one per line. These commands create and manipulate a vector of employee records. There are three types of commands:

+ *id, last, first, salary*

Create a new employee record (object) with the employee ID *id*, last name *last*, and first name *first*, and salary *salary*. The record fields after the **+** command are separated by commas. Field *id* is an **int**, *last* and *first* are each a **string**, and *salary* is a **double**. Insert the new record into the vector while maintaining the sorted order (by employee ID) of the vector. It is an error if the new record has an employee ID that is already in the table. Do not insert an employee record with a duplicate ID.

- *id*

Remove the employee record that has the employee ID *id* from the vector. It is an error if *id* is not in the vector.

? id

Print the employee record that has the employee ID *id* in the vector. It is an error if *id* is not in the vector.

Any other command is an error.

Input file `commands.in`

Your program must read and execute these commands:

```
+ 123,MacDonald,Ron,80000
+ 100,Brown,Bob,150000
% 111,Smith,Alice,95000
? 111
+ 222,Kurtz,Karl,175000
- 333
+ 150,Robins,Roberta,200000
+ 123,Bristol,Buddy,50000
? 123
? 175
? 100
- 100
? 100
+ 175,Anderson,Amy,125000
```

You may assume there are no syntax errors in the input record fields.

Reading comma-separated values

Your program must overload the input stream extraction operator `>>` to read an employee record.

To read the comma-separated values, you can use the `getline` function. It has an optional parameter to specify a delimiter character, such as the comma. See <http://www.cplusplus.com/reference/string/string/getline/>

Vector insertion and deletion

Inserting a new element into a vector at a given position and removing an element at a given position from a vector are operations that require iterators. If `v` is a vector variable, `v.begin()` will return an iterator that points to the first element. Then `v.insert(v.begin + i, employee)` will insert an `Employee` object into position `i`, and `v.erase(v.begin + i)` will remove the `Employee` object at position `i`.

Expected output

Your program must overload the output stream insertion operator << to write an employee record.

After your program has read and executed all the commands, it should then print in sorted order all the employees that are in the vector.

Finally, your program should calculate and print the minimum, maximum, and average salary of the employees in the vector.

CodeCheck will match your output against:

```
+ 123: Employee{ID=123, last=MacDonald, first=Ron, salary=80000}
+ 100: Employee{ID=100, last=Brown, first=Bob, salary=150000}
% 111: *** Invalid command ***
? 111: *** ID not found ***
+ 222: Employee{ID=222, last=Kurtz, first=Karl, salary=175000}
- 333: *** ID not found ***
+ 150: Employee{ID=150, last=Robins, first=Roberta, salary=200000}
+ 123: *** Duplicate ID ***
? 123: Employee{ID=123, last=MacDonald, first=Ron, salary=80000}
? 175: *** ID not found ***
? 100: Employee{ID=100, last=Brown, first=Bob, salary=150000}
- 100: Employee{ID=100, last=Brown, first=Bob, salary=150000}
? 100: *** ID not found ***
+ 175: Employee{ID=175, last=Anderson, first=Amy, salary=125000}

All employees
-----
Employee{ID=123, last=MacDonald, first=Ron, salary=80000}
Employee{ID=150, last=Robins, first=Roberta, salary=200000}
Employee{ID=175, last=Anderson, first=Amy, salary=125000}
Employee{ID=222, last=Kurtz, first=Karl, salary=175000}

Statistics
-----
Minimum salary = $ 80000.00
Maximum salary = $200000.00
Average salary = $145000.00
```

Submission into Canvas

When you're satisfied with your program in CodeCheck, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution. Submit this zip file into Canvas. You can submit as many times as you want until the deadline, and the number of submissions will not affect your score. Only your last submission will be graded.

Submit the signed zip file from CodeCheck into Canvas:

Assignment #6. Employee Records

Note: You must submit the signed zip file that you download from CodeCheck, or your submission will not be graded. Do not rename the zip file.

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Good output (as determined by CodeCheck)	20
Good program design <ul style="list-style-type: none">Well-designed class <code>Employee</code> with public and private members.Vector element search done correctly.Vector element insertion done correctly.Vector element removal done correctly.Overloaded input stream <code>>></code> extraction operator.Overloaded output stream <code><<</code> insertion operator.	60 <ul style="list-style-type: none">101010101010
Good program style <ul style="list-style-type: none">Separate <code>.h</code> and <code>.cpp</code> source files.Descriptive variable names.Meaningful comments.Follow the coding style (formatting, braces, indentation, function declarations before the main, etc.) of the Savitch textbook.	20 <ul style="list-style-type: none">5555

Extra credit (up to 25 points)

Implement a **binary search algorithm** to find employee records in the vector. Add more records to the vector so that you can test your search algorithm with at least 25 records.

Run this outside of CodeCheck. Create a zip file of your C++ source files (`.cpp` and `.h`) and the command file that adds more employee records to the vector. Include a text file of your test output. Submit this zip file into Canvas: **Assignment #6. Extra credit**

The extra credit will be graded according to these criteria:

Criteria	Maximum points
Binary search algorithm <ul style="list-style-type: none">Well writtenEffectively used	25 <ul style="list-style-type: none">1510

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (<http://theory.stanford.edu/~aiken/moss/>). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.