Study.com

# Programming in C++ - Assignment: Writing an Employee Management System

If you have a Study.com College Saver membership and are seeking college credit for this course, you must submit an assignment and pass the proctored final exam. You must submit your assignment before registering for the final. Below you will find prompts and instructions for submitting your assignment.

## About this Assignment

C++ is a general-purpose object-oriented programming language. The best way to understand programming is to learn about flow, syntax, and debugging through practice. You will create a C++ program that incorporates elements you studied in this course.

## Prompt

For this assignment, you will write a program that keeps track of employees at a company. Three types of users can access the system: Human Resources, Management, and General Employees. A Human Resources user will be able to add, view, search, modify, and delete employees. A Management user can only search and view employees. A General Employee user can only view his or her own file. Employee files will have at least the employee's name and numeric user id.

1. Your program will implement a management information system.
2. The minimum requirements of the system are: implement a base employee class that will be defined for each employee type or function, implement a log-in feature, and the login will determine what functions are accessible to the employee.
3. The program will be interactive. At a minimum, a log-in screen and menu screen will be used. Input should be validated for correct login credentials.
4. Functions and classes should be used where appropriate to produce readable, reusable code. Library functions and user-defined functions should be used where appropriate.
5. The program will be well-commented/documented, including a comment block at the top of the document (document header comment) as well as line comments

Some simple rules to follow to make your code more readable:

- Use indentation or spacing consistently. (e.g., every line in a loop is indented to align)
- Add whitespace to help separate parts of the code. (e.g., empty line after a loop, space after a semicolon in a loop, etc.)
- Give meaningful names to variables. Variables named x, y, and z are not descriptive and give the reader no information about their purpose; instead use names like primary, max, and counter. Loop condition variables are an exception because they are contained to a single area of the code and have a universally known function, so naming with a single letter (i, j, etc.) is acceptable.
- Code should be organized into functions so that segments of code that are reused are placed within functions. Function definitions should be contained in a single section of the program. Functions, like variables, should have meaningful names.

## Grading Rubric

| Category | Unacceptable (0-2) | Needs Improvement (3-6) | Good (7-8) | Excellent (9-10) |
|---|---|---|---|---|
| Program Specifications (x5) | Program does not function consistently or at all. | Details of the assignment were violated and/or program exhibits inconsistent behavior. | Assignment is complete and produces correct results in typical cases. | No errors in any case and checks for exception cases. |
| Readability (x2) | Issues with naming, formatting, and organization. | One or two major problems with naming, formatting, or organization. | Code is understandable and well-organized. | Code is understandable and exceptionally well-organized and formatted. |
| Documentation (x2) | No comment header or comments in the file | File comments header missing or comments missing or not meaningful. | Code is appropriately commented but may have a few unnecessary comments. | No errors, code is commented exceptionally well. |
| Good Practice/ Efficiency (x1) | Tasks were not done efficiently, succinctly, or using best practices. | Code uses a bad approach in more than one place. | Code uses good practices and is overall efficient. | Code is exceptionally designed and efficient. |

## Before You Submit

When you complete your assignment, we suggest taking some time to check for any errors or to add any finishing touches. We also suggest that you use online plagiarism checkers such as PlagScan or DupliChecker to make sure that your assignment is not too similar to any existing materials. Plagiarized submissions will NOT be graded.

Thorough testing should be performed on your program to make sure that you can perform each individual function correctly. This should be tried with various input values or input files if any are provided for you. Consideration should also be given to the program behavior if a user inputs an incorrect value.

## How to Submit Your Assignment

When you are ready to submit your assignment, **please fill out the submission form and attach your assignment as a plain text or .C file.** After turning in your assignment, you may go ahead and take the proctored final exam. You do not need to wait for your written response to be graded. You should receive your assignment grade within one week.

If you are not satisfied with the score you receive on your assignment, you may revise or rewrite it, and resubmit it for grading using the same submission form above. Keep in mind that the grade you receive on your assignment is only a portion of your overall grade for the course, and you are free to retake the proctored final exam as well if you choose. Please see the course syllabus for a more detailed breakdown of the grading policy.

♡ Like this lesson f Share

## Explore our library of over 88,000 lessons

**Search** Search Courses & Lessons

**Browse** **Browse by subject** ▾