



## Staff Management System

Design a Staff Management System using C++ programming. The interface of the project prompts the user to choose from a selection of options. The system must be able to store employee id, name, age, address, role and salary in a class called Staff.

The menu must ask the user if they wish to:

1. insert an employee into the system,
2. search for an employee via their id number,
3. search for all employees with a wage higher than 20k.
4. exit the program

Use a while loop and a switch statements to perform the menu part of the project. The user must choose one out of the 4 options listed in the menu [1 or 2 or 3 or 4], if the user entered invalid choice, an error message should be printed, and the screen should be held until the user press enter or any character. After that, the menu should be printed again asking the user to choose from the menu. Every time, the user is done performing the task of his choice, the menu must be printed again. For example, if the user decided to insert a new employee, by the end of this process, the menu must be printed again.

The staff class should contain the following methods:

- a default constructor where the user will set values to the data members (id, name, age, address, role and salary).
- Argument constructor.
- A print method.
- Destructor method.
- Setters and getters.
- Static search method (call it searchID), in this method the user will enter the ID
- Static search method (call it searchWage)

The project executes as following:

1. When the user chooses 1, the default constructor should be called. The data members of the object must be stored in a text file in a table format, before storing the data into the text file, make sure that the id value is not duplicated, use the searchID method for this purpose. if it's duplicated, take a proper action. In the default constructor, make sure to use the setters to assign the user entered values to the data members. Call the destructor method before going back the menu.
2. When the user chooses 2, the project must ask the user to enter the id value, the searchID method should be called, if a match was found, retrieve the data and store them in local variables. Use the variables to create a staff object using the argument constructor. After that call the print method. In the argument constructor, use the MIL. After the print method is executed, Call the destructor method.
3. When the user chooses 3, the searchWage method will be called, you will search the file for wages grater than 20K, whenever a match is found, retrieve the data and store them in local variables. Use the variables to create a staff object using the same argument constructor used in 2. After that call the print method. Make sure to call the destructor method after calling the print method.
4. When the user chooses 4, the program should be terminated.

**Question:** why do you need to call the destructor method in 1, 2 and 3?

Execute the project and test all 4 choices. Provide your code, the text file, and the output of the exaction.