Lab #7: Department Profit Calculator Joseph Lash   
Date Submitted: 4/22/2023  
   
Algorithm:

The purpose of this lab was to create a program that can calculate and output various operations for a department.

Setup

1. Create 5 header files: Employee, Manager, Department, Person, and Date.
2. Create 4 Implementation files: Employee, Manager, Department, and Person.
3. \* Make sure to have proper error checking throughout the program.

Date header file

1. Used given code.

Person class

1. Create private variables firstName, lastName, birthday, ID, and hiringDate.
2. Create set and get functions for each variable in the Person implementation file.

Manager class

1. Connect Person class using inheritance.
2. Create private variables promotionDate, salary, jobTitle.
3. Create an overloaded << for outputting the manager information.
4. Create set and get functions for each variable.

Employee Class

1. Connect Person class using inheritance.
2. Create private variables miles, employeeRating, and totalJobs.
3. Create private vector of floats (vecJobs).
4. Create set and get functions for the variables miles and employeeRating.
5. Overload a < operator for comparing employee names.
6. Overload a << operator for outputting the employee information.
7. Created a function for sorting employees (couldn’t figure this out).
8. Create a function for getting the incomes of all employee jobs.
9. This same function uses a for loop to calculate the total income of all employee jobs.

Department

1. Include Date, Employee, Manager, Person header files.
2. Create private variables capacity, deptName, productCost.
3. Create a private member object of the Manager class.
4. Create a private vector of Employees using composition.
5. Create a tokenization function for dates.
6. Create functions for adding a manager and hiring and firing employees.
7. These functions will prompt the user for information then send them to their respective set and get functions.
8. Create functions for calculating the average employee rating and the cost of travel.
9. Use for loops to calculate these numbers.
10. Create functions for printing off department information and the final profit.
11. Within the department information printing function, use a for loop to print off all the employee information.
12. Call the overloaded << operators to print off the employee and manager information.

Main

1. Create a class object of department.
2. Prompt for information from the user and send it to the department class.
3. Create a Main Menu with different options which will use a switch statement.
4. Create a counter variable which will prevent certain options from being chosen if the department is at max capacity, or if there is already a manager.
5. Option M adds a manager.
6. Option A adds an employee.
7. Option R removes an employee.
8. Option L lists the information of the department.
9. Option P prints the final profit.
10. The prompting for these options is done in the department class.

Screen Shots of Running Program:

Calculation Output:

A screenshot of a computer

Description automatically generated with medium confidence

Main Menu:

A screenshot of a computer

Description automatically generated with medium confidence

Integrity Statements:

* I have not shared the source code in my program with anyone other than the pre-approved human sources.
* I have not used source code obtained from another student, or any other unauthorized source, either modified or unmodified.
* If any source code or documentation used in my program was obtained from another source, such as the course textbook or course notes, that has been clearly noted with a proper citation in the comments of my program.

I have not knowingly designed this program in such a way as to defeat or interfere with the normal operation of any machine it is graded on or to produce apparently correct results when in fact it does not.