Final Project

## Description

You will create a class that follows the instructions specified below. When completed, you will submit a tar.gz file (tar -zcf) named “FinalProject.tar.gz” to your group’s Dropbox. It must include a Makefile and all source code and header files used to build your final binary.

## Details

The Dropbox contains 1 file called inputFiles.tar.gz. In this archive are 4 files: smallOutput.txt, mediumOutput.txt, and largeOutput.txt. Each file is comprised of records representing an employee of a fictitious company, and follows these rules:

* Each line represents the record of one employee.
* Each record is comprised of department number, employee number, and employee name.
* There are 5 possible departments: Accounting(0), Business(1), Human Resources(2), Sales(3), and Production(4).
* The maximum allowable employee name is 30 characters.

Each file has an expected running time. See the examples section for more details.

## Instructions

You will create a C++ class called ‘binaryFile’ that will implement methods that are described below. You must use a header file for the class which will contain the class definition, all structure and/or enumerator definitions, and header files necessary for execution.

* Read data from a file (described in the “Details” section above) and writes the data out to a binary file. (5 pts)
* Sort the binary file by department, and within each department sort in ascending order (0…100) by employee number. (10 pts)
* Search the binary file for an employee based on the department and employee number. The public method must return a bool (true/false) that represents if the employee was found or not. The public method must only take the department number and employee number as arguments. The private method should return the offset within the file where the employee is located. (10 pts)
  + In main, use the return value to print “Employee found” or “Employee not found”. This should not be part of the ‘search’ method’s functionality.
* Retrieve from the binary file an employee based on the department and employee number. This public method must return the employee details (department, employee number, and name.) The public method must only take the department number and employee number as arguments. (10 pts)
  + In main, use the return value to print the employee details to the screen. This should not be part of the ‘retrieve’ method’s functionality.
* Update an employee’s name based on the department and employee number. The public method must return a bool (true/false) that represents if the employee was updated or not. The public method must only take a structure containing the employee’s department, number, and string. (10 pts)
  + In main, use the return value to print “Employee updated” or “Employee not updated”. This should not be part of the ‘search’ method’s functionality.
* Use the provided exception handling class to create exceptions as appropriate to account for, at a minimum: inability to open files, data does not conform to expectations, error occurred during normal operations. Examples situations that should result in an exception handling message. (10 pts)
  + File does not exist
  + Employee name exceeds 30 characters
  + Could not convert employee record from text to binary.

Your class will follow the principle of accessing private variables from the private scope and using getters and setters to retrieve and store or update public and private variables.

You will also utilize a Makefile which will at a minimum do the following (5 pts):

* Create the executable binary “excBinaryFile”.
* Create a clean rule which will remove all files created from the build and execution processes to include \*.o files, excBinaryFile, and any temporary files that may have been created during the execution of excBinaryFile.
* Create object files for linking into the excBinaryFile.

## Examples

Use the values below to establish your program executes as expected.

|  |  |  |
| --- | --- | --- |
| **smallOutput.txt** | | |
| **Entries do exist** | **Entries do not exist** | **Expected Execution Time: ~1 sec** |
| 0,37503,Brylee | 0,18738,Hunter |  |
| 1,58510,Madisyn | 1,63006,Annabel |
| 2,44258,Jackson | 2,757,Layton |
| 3,45731,Sylvia | 3,68263,Maxim |
| 4,58508,Janiyah | 4,65898,Rex |

|  |  |  |
| --- | --- | --- |
| **mediumOutput.txt** | | |
| **Entries do exist** | **Entries do not exist** | **Expected Execution Time: ~18 sec** |
| 0,135016,Cyrus | 0,139708,Benicio |  |
| 1,243012,Lucia | 1,250506,Amari |
| 2,15009,Batteo | 2,206998,Cory |
| 3,105010,Patricia | 3,6011,Zariyah |
| 4,297008,Raphael | 4,243114,Mariana |

|  |  |  |
| --- | --- | --- |
| **largeOutput.txt** | | |
| **Entries do exist** | **Entries do not exist** | **Expected Execution Time: ~54 sec** |
| 0,205003,Evelyn | 0,205508,Meadow |  |
| 1,25016,Corinne | 1,335014,Crosby |
| 2,189708,Zaiden | 2,430011,Clara |
| 3,155013,Antonella | 3,5014,Watson |
| 4,430010,Clara | 4,433415,Kyler |