# Lab Week 11

Skills Required

* Polymorphism, Create and use classes, Exception Handling, Read and write Files, Work with arrays and vectors, Create Functions, Include Headers and other files, Loops (while, for), Conditional (if, switch), Datatypes, assignment, etc.

**Assignment**

You will write a program that keeps track of employees and their pay using the techniques learned in the second half of lab.

**Part 1**

You are provided the basic frame for an **Employee class** in **Employee.h and Employee.cpp**. It will contain their name, their pay rate in dollars, and the amount of money they have made at the company (the balance). The Employee can get paid, get a raise, or be fired. There are functions provided to do these things.

* giveRaise: It takes in an integer, which is the *percentage* of a pay raise. (For instance, if you make $10.00 per hour and you get a 5% raise, you will then make $10.50 per hour.) This function should set their new pay rate.
* pay: This should pay the employee once.
* fire: This should remove the employee from the payroll. It should set their pay rate to 0, and isEmployed to false. (Note: Retain the records for the employee, just make sure they aren’t paid anymore and are marked as not employed here.)
* isEmployed: This is a Boolean function that should return if they are employed or not.

Fill in the function definitions for the Employee. Employees should start off with a base pay of $10.00.

Create a file called **Lab11.cpp** and add your main() function inside of that. Create some Employee objects, and test out their pay rates and pay. Make sure you get correct results. If not, now is when you want to debug your code before moving on. You will need to include the relevant libraries.

**Part 2**

You will use a vector to store the employee’s information.

Create your employee vector, and test adding employees to it, changing their pay rates, and paying them. Ensure the vector is what you expect (either by setting breakpoints and checking the vector, or printing contents to an output source.)

Debug any issues you have, then move forward.

**Part 3**

Write the remainder of your main function to support the commands listed below in the File Input/Output section. It should use the newly created vector to alter the Employee data.

Debug this part to find any remaining issues in your program.

**File Input/Output**

You will need to read from a file called input.txt, and write out to a file called output.txt. The input file contains the following commands:

* NEW: This will add a new employee to the company. The number after the command is their employee ID number, and the next two words are their first name and last name.
* RAISE: This will give the employee a raise. The number after the command is their ID number, and the next number is the percentage their pay goes up (for example, “5” for a 5% raise.)
* PAY: This will pay all employees.
* FIRE: This will fire an employee. The next number is the ID number of the employee to fire.

The output file should contain:

* The employee’s name, followed by their ID number in this format:  
   Name, ID Number ###:
* If the employee is currently employed, it should write out their pay rate in this format:  
   Current pay rate: $##
* If they are not employed, you should write out that:  
   Not employed with the company.
* The employee’s balance to date:  
   Pay earned to date: $##
* There should be a blank line between employees.
* See the sample output for what it should look like when done.

Your program will be graded with a different file (still called input.txt) and will be graded against a different output.txt file. They should match exactly if your program works correctly.

**Hints**

* **WRITE A LITTLE AT A TIME AND TEST REGULARLY!!!** It will save you a lot of time debugging (and save you a lot of points) if you make sure your program will compile on a regular basis.
* Set breakpoints in your code and see what your variables are at any given moment. Use this to ensure you’re reading or writing correctly.
* Do not spend more than a few minutes on any bit of code. If you can’t figure it out after a while, try moving onto a different section of code.

**Submit your assignment**

Update your files on GitHub and submit your GitHub link to Canvas.