Nicholas Grogg

Dr. Edwards

CS151

**Program 4: Part 1 and 2**

**Program 4: Part 1: Program output**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pay to the order of Todd Davis

The sum of 54.62 Dollars \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Check Stub NOT NEGOTIABLE

Employee Number: 457-55-5462

Salaried Employee. Regular Pay: 54.62 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Hourly Employee Name: Sir\_Beavis

Hourly Employee SSN: 1701

Number of Hours Worked: 69

Hourly Rate: 0.01

Check for Sir\_Beavis for 69 hours. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pay to the order of Sir\_Beavis

The sum of 0.69 Dollars \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Check Stub: NOT NEGOTIABLE

Employee Number: 1701

Hourly Employee.

Hours worked: 69 Rate: 0.01 Pay: 0.69 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Press any key to continue . . .

**Program 4: Part 2: The Plan**

**Program idea**

This program is going to be based on the already existing code, in addition to the employee, hourly employee, and salaried employee classes I need to create a salaried employee class, an executive employee class and a ~~intern~~ temp slave class. These will all be separate files with their own .cpp and .h files to keep the driver file simple.

The general idea is to allow maximum flexibility is to create the classes with a linked list. This allows the user to enter just one employee if they prefer or as many as the memory allows. This linked list will NOT be circular. I'll be executing the program in a while loop with a bool, similar to the first program. In a way this program is a mixture of the first program, the linked list lab, and the inheritance lab. This helps because then the only new concept to get snarled on is the inheritance.

Launching the program will create all the classes with an empty linked list, then enter the loop. From there the user can enter commands to interact with the menu. I will also include a minor help function in main to output the possible commands.

**Variables used**

In addition to all the variables in the HRM.cpp program

string for last name and first name

A bool for the while loop

Int variable for menu input

Int variable for employee type

int variable for employee ID, used to find correct employee

Variables for the temp slave class such as

the termination date

Variables for the executive class such as

string title (e.g. Director of Product Enhancements).

string for area of responsibility.

string for employee’s supervisor.

protected double that holds the executive’s annual salary.

protected double that holds the bailout bonus for the executive.

New variable for salaried employee salary grade clarification

**Functions used**

In addition to all the functions in the HRM.cpp program.

The temp slave and executive classes.

Temp accessor and mutator for termination date

executive functions

accessors to set the annual salary and bonus variables.

input function for reading in the administrator’s data from the keyboard.

print function which outputs the administrator’s data to the screen.

A redefined print\_check( ) that gives a swiss bank account number.

Add item to linked list

Remove item from linked list

Check if list is empty

Output list

Search list for item.

Overloaded >> & << operators to allow single line input.

**Test plans**

Creating an hourly employee

First I'd enter 1 to create an employee

Then I'd enter 1 to select an “hourly employee” type

Then I'd enter “1071 Picard-better than-Kirk 123-45-6789 40 42.42” for the EID, first name, last name, SSN, hours worked, hourly rate using the overloaded >> operator

The EID is the primary key used to modify or delete an employee. I won't have a system in place to check for doubles.

Creating a salaried employee

First I'd enter 1 to create a new employee

Then I'd enter 2 to select “Salaried Employee” type

Then using the overloaded >> operator I'd enter “1001001 Moron Bros 987-65-4321 000.00 1” to enter the EID, first name, last name, SSN weekly wage and salary grade.

Creating an executive

Enter 1 to create a new employee

Enter 3 to select “Executive employee” type

Then enter “10 Sir Derpington 454-18-3511 21.00 2 Boss Everything none 100.00 1,000,000,000.00” to enter the EID, first name, last name, SSN, weekly wage, salary grade, title, area of responsibility, supervisor, annual salary, bailout bonus

Creating a temp slave employee

Enter 1 to create a new employee

Enter 4 to select “Temp Slave” type

Then enter “21 Coffee Les 111-22-3333 80 1.25 12-12-2017” to enter EID, first name, last name, SSN, hours worked, hourly rate, and termination date.