Nicholas Grogg

Dr. Edwards

CS 151

**Program 4 Part 2: The Plan**

**General Program Idea**

My whole program will run in while loop with a bool variable. It will use a numeric entry to determine what to do, in a manner similar to the first program. For example entering 6 will set the bool to false and close the program, entering 3 will display an employee's information.

To make the program more dynamic I'll be using pointers to create a linked list within the classes. This will allows users to easily create as many of whatever type or employee as they like as long as the memory doesn't run out. The linked list will not be circular. Upon activating the program it'll create the classes with empty linked lists in each class. Adding to the classes will add an employee to the linked list. Accessors and mutators will function by finding the employee in the list and then altering whatever was supposed to be altered. This program is essentially for me a mixture of Lab 10, The Linked List lab and Program 1.

**Classes**

All classes from HRM.cpp, including the expanded salaried employee class

Executive sub-class, based on salaried employee

temp slave ~~intern~~, based on hourly employee

**Functions used**

All functions from HRM.cpp program

struct for the linked lists

A function to add an item to the linked list

A function to search for an item in the linked list

A function to remove an item from the linked list

A function to output the linked list

Accessors and Mutators to change the data in the linked lists

Overloaded >> and << operators (assuming they aren't already in HRM.cpp)

**Variables used**

All variables from HRM.cpp class

Int variables for the menu program, each number is used for something

1- Create an employee

2- Modify an employee's information

3- Display an employee's information

4- Remove an employee

5- Monthly Payroll, print checks for employees

6- Used to output employee report

7- Quit program

Bool for while loop, run while true

variables exclusive to executive sub-class

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

string for area of responsibility.

string for employee’s supervisor.

A protected: member variable of type double that holds the executive’s annual salary.

A protected: member variable of type double that holds the bailout bonus for the executive.

Member functions to set the annual salary and bonus variables.

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

A member function called print, which outputs the administrator’s data to the screen.

An redefined member function for print\_check( ) that gives a swiss bank account number.

variables exclusive to temp-slave sub-class

A termination date, string