CISC 181 Project 1 – Computing Wages[1]

Due: At 11:00am (class time) on Tuesday, March 8, 2005.

A company pays its employees as managers (who receive a fixed weekly salary), hourly workers (who receive a fixed hourly wage for up to the first 40 hours they work and "time-and-a-half," i.e., 1.5 times their hourly wage, for overtime hours worked), commission workers (who receive $250 plus 5.7% of their gross weekly sales), or pieceworkers who receive a fixed amount of money per item for each of the items they produce – each pieceworker in this company works on only 1 type of item).

Write a program to compute the weekly pay for a number of employees whose information will be input one at a time. You do not know the number of employees in advance. Each type of employee has its own paycode (and this will be the first piece of input for each employee whose pay you are to calculate): Managers have paycode 1, hourly workers have paycode 2, commission workers have paycode 3, and pieceworkers have paycode 4.

Use a `switch` to compute each employee's pay based on that employee's paycode. Within the `switch`, prompt the user (i.e., the payroll clerk) to enter the appropriate facts your program needs to calculate each employee's pay based on that employee's paycode. After each employee's information is entered, print out the weekly pay for that employee.

After all employees have been processed, your program should print out a table summarizing the processing. For each type of employee, it should print out the employee-type (e.g., manager), the number of employees of that type processed, the total salary paid to that type of employee, and the average salary for that type of employee. Finally, the program should print the total amount paid out to employees that week.

Hand-in a single file containing the following: (The file can be created using the UNIX command `script` as done previously in lab.)

- A listing of your program with appropriate comments and formatting. Use the UNIX command `cat` to list the file containing your program, or use one of the methods discussed in email if you are working on your PC.

- A script file of enough runs of test data to verify that the program is correct. Remember that the program may be run multiple times after

[1] This project is adapted from exercise 2.56 on p. 166 of Deitel and Deitel, *C++ How to Program*.

it is compiled once. Be sure to check boundary conditions such as no employees of a given type being entered.

Rules of the Game

For programming projects, unlike labs, you are expected to do all the work on your own. You may, however, ask another student for help *debugging* your program. Please also see the TA and/or instructor for any conceptual help you need and for debugging help. Many students are tempted, especially later in the semester when they have much to do, to improperly obtain help from other students. This is the most common form of academic dishonesty in computer science courses and is monitored carefully. Unfortunately, quite a few students have been referred to the Student Judicial System for such problems. So please do not resort to using the solutions of others as your own. Examples of actions that are specifically prohibited include:

- submitting another person's work as your own.
- using another person's solution as a model for your own work.
- except as explicitly directed, working together on an assignment, sharing the computer files or programs involved.
- knowingly allowing another student to look at, copy, or use one of your computer files.

Late projects will be penalized 5% a day for each 24 hours beyond the due date. No assignments will be accepted more than 7 days late.