**Assignment #8**

**Introduction to C Programming – COP 3223**

**Objective**

1. To give students practice in writing and calling their own functions.
2. To give students practice in planning and implementing complex programs.

**Introduction: Ultimate Computing Fun**

A new game store is opening in your area! Ultimate Computing Fun (UCF) Games is looking for a home for their wide variety of computer games and has decided to make you a part of their growing business.

As we get closer to summer, UCF Games would like to improve their personnel system in advance of any new hires.

**Problem: Personnel System (personnel.c)**

A personnel system should be able to track the current employees, support the hiring of new employees, record the number of hours each employee works and their hourly pay.

In this assignment, you will be filing in the functions of a program to support these tasks. There are five functions; four correspond to the available menu options and one is to print the menu.

The main function is already written for you and should not be modified.

Here is what the menu should look like:

1 - Hire a new employee

2 - Update an employee's hours

3 - Print personnel information

4 - Calculate payroll

5 – Quit

The program personnel\_skeleton.c contains the code for you to fill in. This includes the function prototypes, main function, preconditions and postconditions for each function body. Only fill in the function bodies.

As UCF Games is a small store, the maximum number of employees is 10.

**Program Details**

*Menu*

The menu function is slightly different from those we have completed in class. The purpose is to print the menu as shown above and then scan in the user’s choice. This choice should be returned.

*Hire a new employee*

The hiring function takes in the two arrays required to store employee information and a pointer called num. This pointer points to the location where the current number of employees is stored.

The hiring function should print which employee number we are hiring for. Then ask the user how many hours per week they will work and how much they will be paid for each hour of their time.

These values should be stored in the appropriate spots of the arrays. The number of employees should be increased by one.

*Update an employee's hours*

The update function takes in the employee hours arrays which stores each employee’s hours per week. The purpose of this function is to update one employees work hours.

You should ask the user which employee to modify. Recall that users will be using a 1-based numbering systems, which arrays use a 0-based numbering system. You are guaranteed that the user will enter a valid employee number.

Then, ask the user how many hours the employee will work each week. Update the employee’s information in the array.

*Print personnel information*

The print function takes in the two arrays required to store employee information and an integer called num. Num is the current number of employees.

The purpose of this function is print a table of employee information. You should first include this header:

Number Hours Pay Rate

Then, print the corresponding information for each current employee. See the program sample run below for more information.

*Calculate payroll*

The payroll function takes in the two arrays required to store employee information and an integer called num. Num is the current number of employees.

The purpose of this function is to calculate the weekly pay for each current employee. This is a simple calculation of the employee’s hours per week times their hourly pay. Print the result for each employee to two decimal places. See the program sample run below for more information.

We will not account for things like overtime, salary, or taxes.

**Input Specification**

The user will enter integers for their menu selection.

The user will enter valid employee numbers, hours, and pay amounts.

**Output Sample**

Below is a sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity’s sake.)

**Welcome to the UCF Games Personnel Records.**

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*1*

**This will be employee #1.**

**How many hours per week will this employee work?**

*40*

**What is this employee’s hourly pay?**

*18*

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*3*

**Number Hours Pay Rate**

**#1 40 18.00**

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*1*

**This will be employee #2.**

**How many hours per week will this employee work?**

*20*

**What is this employee’s hourly pay?**

*10*

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*2*

**Which employee do you want to update?**

*1*

**How many hours will employee #1 work?**

*35*

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*3*

**Number Hours Pay Rate**

**#1 35 18.00**

**#2 20 10.00**

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*4*

**Employee #1 will make $630.00 this week.**

**Employee #2 will make $200.00 this week.**

**What would you like to do?**

**1 - Hire a new employee**

**2 - Update an employee's hours**

**3 - Print personnel information**

**4 - Calculate payroll**

**5 – Quit**

*5*

**Thank you for using our system!**

**Deliverables**

One source files – *personnel.c* – is to be submitted over WebCourses.

**Restrictions**

Although you may use other compilers, your program must compile and run using Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

**Grading Details**

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.