

Task 1: Write a C++ program that will determine the gross pay for an employee. The company pays 'straight time' for the first 40 hours worked by each employee, and pays 'time and a half' for all hours worked in excess of 40 hours. Your program should prompt the user to enter at the keyboard the number of hours worked and the hourly rate for an employee, and display in a neat way on the screen the gross pay for the employee. After each output on the screen, ask the user if they would like to enter details for another employee (Y/N). If the user types a 'Y' (in upper or lower case), then repeat the last step, otherwise the program ends.

Part B: (Read Hints below and Chapter 12 of the course text book (Absolute C++ 5th Ed))

Task 2: Workers at a particular company have won a 7.6% pay increase retroactive (i.e. back-paid) for 6 months. Write a program that will read from a file the previous annual pay of one employee, and will overwrite the file to provide the amount of back-pay pay due to the employee, the new annual salary, and the new monthly salary. Use a variable declaration with the modifier `const` to express the percentage pay increase. Hint: if the text file is in the same folder as your project files, there will be no need to provide a path to the file.

Task 3: Revisit task 2. Now suppose you have a file which records each employee's name and salary on a separate line. Your program should construct a new file which contains for each employee the name of the employee, back-pay pay due to the employee, the new annual salary, and the new monthly salary. The file should provide this information in neat columns. It should also provide a final line which records the number of employees, and the total of back pay which must be paid. Remember to close files which you have opened before your program ends.

Hints

- if you are asked to open a file to read from it, and later to overwrite it, you must open as an input-file stream (`ifstream`), and when you have finished reading from it, close it before opening as an output-file (`ofstream`)
close a file by using the close method of `ifstream` class
`myfile.close();`
- if you are using file input or output: `#include <fstream>`
- if you are storing the name of the employee, you can use a string data-type (note the string data-type has a lower-case 's') and: `#include <string>`
- to determine if the end of the text file has been reached, check after each read that more was found in the file (i.e. end not yet reached), so you might have code a bit like this:

```
string name;
double salary;
myFile >> name >> salary;
while (!myFile.fail() )
{
    //do something with the name and salary read in

    // then try another read
    myFile >> name >> salary;
}
//now wrap up!
```