

Simple Payroll Program (P3)

```
#include <fstream>      // for file streams for input/output
#include <iomanip>      // for formatting manipulators
#include <string>       // for string variables
using namespace std;

// Reads the employee's pay data from the input file.
void GetData(ifstream& In, string& EmpID, double& HWorked, double& HRate,
             double& FITrate, double& SITrate, double& Insrate);

// Calculates employee's gross pay amount:
double Gross(double HWorked, double HRate);

// Calculate employee's tax withholding amounts:
double FIT(double Gross, double FITRate);
double FICA(double Gross);
double SIT(double Gross, double SITRate);
double FMED(double Gross);

// Prints out the employee's pay stub.
void WriteResults(string EmpID, double Gross, double FIT, double FICA,
                 double FMED, double SIT, double Ins, double Net);

// . . . continues . . .
```

```
void main() {

    ifstream In("PayData.txt");    // Open the input file.

    string EmployeeID;            // employee's ID number
    double HoursWorked,           // hours employee worked this pay period
           HourlyRate,           // employee's hourly pay rate
           FITrate,              // FIT rate
           SITrate,              // SIT rate
           InsPremium;           // insurance premium

    double GrossPay,              // employee's gross pay amount (hours * rate)
           FITWithheld,           // FIT withholding amount
           FICAWithheld,         // FICA withholding amount
           FMEDWithheld,         // FMED withholding amount
           SITWithheld,          // SIT withholding amount
           NetPay;                // net pay amount (gross - all
    deductions)

    GetData(In, EmployeeID, HoursWorked, HourlyRate, FITrate,
            SITrate, InsPremium);
    In.close();

    // . . . continues . . .
```

```
// Calculate the gross pay for this period:
GrossPay = Gross(HoursWorked, HourlyRate);

// Calculate the deductions for this period:
FITWithheld = FIT(GrossPay, FITrate);
FICAWithheld = FICA(GrossPay);
SITWithheld = SIT(GrossPay, SITrate);
FMEDWithheld = FMED(GrossPay);

// Calculate the net pay for this period:
NetPay = GrossPay - FITWithheld - FICAWithheld - SITWithheld -
        FMEDWithheld - InsPremium;

WriteResults(EmployeeID, GrossPay, FITWithheld, FICAWithheld,
            FMEDWithheld, SITWithheld, InsPremium, NetPay);
}

// . . . continues . . .
```

```
////////////////////////////////////// GetData()  
//  
// GetData() reads the employee's payroll data from the input file.  
//  
// Input/Output parameter:  
//   In      input stream  
//  
// Output parameters:  
//   EmpID   employee's ID string  
//   HWorked # of hours employee has worked in this period  
//   HRate   hourly pay rate for employee  
//   FITrate rate for computing FIT withholding  
//   SITrate rate for computing SIT withholding  
//   Insrate amount of insurance premium  
//  
// Preconditions:  
//   In has been opened on a properly formatted input file.  
//  
// Postconditions:  
//   The specified data has been read and stored in the corresponding  
//   parameters.  
//  
// Called by:  main()  
// Calls:     none
```

```
void GetData(istream& In, string& EmpID, double& HWorked, double& HRate,
             double& FITrate, double& SITrate, double& Insrate) {

    const char Delimiter = '|';    // label separator in pay data file

    In.ignore(255, '\n');          // ignore the header line
    In >> EmpID;                   // read employee ID
    In.ignore(255, Delimiter);     // ignore everything through hours worked
                                   // label

    // Read employee's pay data:
    In >> HWorked;                  // Read the # of hours worked this period.
    In.ignore(255, Delimiter);     // Advance to the next value.
    In >> HRate;                    // Read the hourly pay rate.
    In.ignore(255, Delimiter);     // Advance to the next value.
    In >> FITrate;                  // Read the FIT rate.
    In.ignore(255, Delimiter);     // Advance to the next value.
    In >> SITrate;                  // Read the SIT rate.
    In.ignore(255, Delimiter);     // Advance to the next value.
    In >> Insrate;                  // Read the insurance premium amount.

    return;
}
```

Simple Payroll Program (P3)

```
////////////////////////////////////// NOTE!!  
// Function header comments omitted to save space in notes.  
//  
double Gross(double HWorked, double HRate) {  
  
    return (HWorked * HRate);  
}  
  
double FIT(double Gross, double FITRate) {  
  
    return (Gross * FITRate / 100.0);  
}  
  
double FICA(double Gross) {  
  
    const double FICArate = 1.45;  
  
    return (Gross * FICArate / 100.0);  
}
```

```
////////////////////////////////////// WriteResults()  
//  
// WriteResults() writes the employee's payroll stub to the output file.  
//  
// Input parameters:  
//   EmpID      employee's ID string  
//   Gross      employee's gross pay amount for this period  
//   FIT        FIT withholding for this period  
//   FICA       FICA withholding for this period  
//   FMED       FMED withholding for this period  
//   SIT        SIT withholding for this period  
//   Ins        insurance premium for this period  
//   Net        employee's net pay amount for this period  
//  
// Preconditions:  
//   Each of the function parameters has been assigned the correct value.  
//  
// Postconditions:  
//   A pay stub, formatted as specified, has been printed to the output file  
//   "PayStub.txt".  
//  
// Called by:   main()  
// Calls:      none  
//
```

```
void WriteResults(string EmpID, double Gross, double FIT, double FICA,
                 double FMED, double SIT, double Ins, double Net) {

    ofstream Out("PayStub.txt"); // Open the output file.
                                // Enable floating-point output formatting.
    Out << fixed << showpoint;

    // Write the header information to the output file:
    Out << "Programmer: Bill McQuain" << endl;
    Out << "CS 1044 Project 3 Fall 2000" << endl;
    Out << endl;

    // Mark the beginning of the pay stub:
    Out << "-----" << endl;

    // Write out the pay stub:
    Out << "Pay data for: " << EmpID << endl;
    Out << endl;

    // . . . continues . . .
```

```
Out << "Gross pay: " << setw(8) << setprecision(2) << Gross << endl;
Out << endl;

Out << "Tax deductions:   FIT       FMED       FICA       SIT" << endl;
Out << endl;
Out << setw(21) << setprecision(2) << FIT
    << setw( 9) << setprecision(2) << FICA
    << setw(10) << setprecision(2) << FMED
    << setw( 9) << setprecision(2) << SIT
    << endl;
Out << endl;
Out << "Other deductions: Health Ins" << endl;
Out << setw(28) << setprecision(2) << Ins << endl;
Out << endl;

Out << "Net pay: " << setw(8) << setprecision(2) << Net << endl;

// Mark the end of the pay stub:
Out << "-----" << endl;

// Close the output file:
Out.close();
return;
}
```