The following source code provides *one* solution for the *programming Exercise 4*.

```cpp
/*******************************
* Program: e4.cpp - Exercise 4 - Coding Solution *
*******************************/
#include <iostream> // Input/Output Stream Objects
#include <iomanip> // Input/Output Stream Manipulators
using namespace std; // Specifies default namespace for objects

#define THRESHOLD  40.0  // Overtime hours threshold
#define MULTIPLIER  1.5  // Overtime Pay Multiplier
#define TAXRATE  0.03625 // Employee Tax Rate

int main ()
{

float // floating point data with 6 or less significant digits
    PAYRATE,  // Hourly Pay Rate
    WKHRS,   // Overtime Pay Multiplier
    OTHRS,   // Overtime Hours
    REGHRS;  // Regular Hours

double    // floating point data with more than 6 significant digits
    OTPAY,  // Overtime Pay
    REGPAY, // Regular Pay
    GROSS,  // Gross Pay
    TAX,    // Payroll Tax
    NET;    // Net Pay

system ("cls"); // Clear the Screen
cout << "PAYROLL CALCULATOR\n\n";
cout << "Written by Randy Gibson - 1/1/2012\n";
cout << "Coded by Joe Student - 1/2/2012\n\n";
```
cout << "Employee's hourly pay rate? ";
cin >> PAYRATE;
cout << "Total hours worked this week? ";
cin >> WKHRS;

cout << setw(4) << setprecision(1) << fixed;
cout << "\nOvertime threshold = " << THRESHOLD << " hours";
if (WKHRS <= THRESHOLD) { cout << " (not exceeded)"; OTHRS=0.0; }
else { cout << "\nOvertime multiplier = " << setw(3) << MULTIPLIER; OTHRS=WKHRS-THRESHOLD; }

REGHRS = static_cast<int> ((WKHRS-OTHRS) * 10 + 0.5) / 10.0;
REGPAY = static_cast<long int> (REGHRS*PAYRATE * 100 + 0.5) / 100.0;
OTPAY  = static_cast<long int> (OTHRS*MULTIPLIER*PAYRATE * 100 + 0.5) / 100.0;
GROSS  = REGPAY+OTPAY;
TAX    = static_cast<long int> (GROSS*TAXRATE * 100 + 0.5) / 100.0;
NET    = GROSS-TAX;

cout << setw(4) << "\n\nRegular hours = " << REGHRS;
cout << setw(6) << setprecision(2) << " Regular pay = $" << REGPAY;
if (WKHRS > THRESHOLD)
{
    cout << setw(4) << setprecision(1) << "Overtime hours = " << OTHRS;
    cout << setw(6) << setprecision(2) << " Overtime pay = $" << OTPAY << endl;
}
cout << "\n\nGross pay before taxes = $" << GROSS << endl;
cout << setw(7) << setprecision(3) << TAXRATE*100 << "% payroll taxes = ";
cout << setw(6) << setprecision(2) << TAX << endl;
cout << "\nNet pay after deducting taxes = $" << NET << endl;

system ("pause");  // Debugging support statement to be removed
return 0;  // Send a null error code to the parent process