Structures

```
using namespace std;
#include <iostream>



   int main ()
  {   int id,i;
      double gross, hours, rat
      fedtax, statetax;

      //---------------------------------
      for (i=0;i<3;i++)
      {cout << " Enter id ";
         cin >> id;
         cout << " Enter Hours  ==> ";
         cin >> hours;
         cout << " Enter Rate ==> ";
         cin >> rate;

      //---------------------------------

         gross = hours*rate;
         net = gross*0.7;
      //---------------------------------
         cout.setf(ios::fixed);
         cout.setf(ios::showpoint);
         cout.precision(2);
         cout << "Gross : $ " <<  gross <<  "\t";
         cout << "Hours : " <<  hours <<  endl;
         cout << "Rate  : " <<  rate <<  endl;
         cout << "net   : " <<  net <<  endl;

      }
      return (0);
   }
```

The variables in the old version that are related to the employee are now in the structure emprecord and are considered to be fields in the structure.

The variables in the old version that are related to the employee now are now fields of the variable name (in this case ***employee***) and must be referred to with the variable name, followed by a period and the field name.

```
using namespace std;
#include <iostream>
struct emprecord
{ int id;
      double gross, hours, rate, net, fed, state, fica, fedtax,
statetax;
};

 int main ()
   { int i;
     emprecord employee;

      //----------------------------------------------------
      for (i=0;i<3;i++)
      {cout << " Enter id ";
         cin >> employee.id;
         cout << " Enter Hours  ==> ";
         cin >> employee.hours;
         cout << " Enter Rate ==> ";
         cin >> employee.rate;

      //----------------------------------------------------

         employee.gross = employee.hours* employee.rate;
         employee.net = employee.gross*0.7;
      //----------------------------------------------------
         cout.setf(ios::fixed);
         cout.setf(ios::showpoint);
         cout.precision(2);
         cout << "Gross : $ " <<  employee.gross <<  "\t";
         cout << "Hours : " <<  employee.hours <<  endl;
         cout << "Rate  : " <<  employee.rate <<  endl;
         cout << "net   : " <<  employee.net <<  endl;
      }
      return (0);
    }
```

The variable employee is declared to be of type emprecord.