Project #3 Hints

Example Code for Project #3

```
     // Open input file
     ifstream infile;
     infile.open("empinfo.data");

     //Check if input file exists
     if ( !infile ) {
          cout << "\nFile does not exist. Use Add and Save options to
create file\n";
          return;
     } //if

     char Str[80]; // string to be read from the file
     char d[] = ";"; //delimiter

     int i = 0;
     pHeadPtr = 0;

     while (infile.getline(Str, 80)) {
          pCurrPtr = new CEmployee;
          pCurrPtr->m_pLink = pHeadPtr;
          pHeadPtr = pCurrPtr;

          strcpy(pCurrPtr ->m_Name,strtok(Str, d));
          pCurrPtr ->m_Age = atoi(strtok(NULL, d));
          pCurrPtr ->m_Salary = atoi(strtok(NULL, d));
          i++;
     } //while

     infile.close();
     cout << endl << i << " employee records are loaded from the data
file into the link list" << endl;
```


Recommended Project #3 Code Design

```
/*************************************************************************
*   FILE NAME: Prj3                                                      *
*************************************************************************/
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
#include <string.h>

/*************************************************************************
*   CLASS NAME: CEmployee                                                *
*   PURPOSE:                                                             *
*                                                                        *
*   MEMBER FUNCTIONS:                                                    *
*     void Open (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr );           *
*     void AddEmp (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr );         *
*     void DeleteEmp (CEmployee *&pHeadPtr);                             *
*     void SearchEmp (CEmployee *pHeadPtr);                              *
*     void ListAllEmp (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr);      *
*     void SaveToFile (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr);      *
*     void ExitProg (CEmployee *pHeadPtr);                               *
*************************************************************************/
class CEmployee {
   public:
      void Open (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr );
      void AddEmp (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr );
      void DeleteEmp (CEmployee *&pHeadPtr);
      void SearchEmp (CEmployee *pHeadPtr);
      void ListAllEmp (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr);
      void SaveToFile (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr);
      void ExitProg (CEmployee *pHeadPtr);

   private:
     char m_Name [20];
     unsigned int m_Age;
     unsigned int m_Salary;
     CEmployee *m_pLink;
};

/*************************************************************************
*   FUNCTION: Open                                                       *
*   PURPOSE:                                                             *
*                                                                        *
*   PARAMETERS: pHeadPtr, pCurrPtr                                       *
*   LOCAL VARIABLES: temp_name                                           *
*************************************************************************/
void CEmployee::Open (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr )
{

}
```



```
/*****************************************************************
*   FUNCTION:  DeleteEmp                                         *
*   PURPOSE:  Deletes employees from link list.                  *
*   PARAMETERS:  pHeadPtr                                        *
*   LOCAL VARIABLES: CurrPtr, LastPtr, Del_name                  *
*****************************************************************/
void CEmployee::DeleteEmp (CEmployee *&pHeadPtr)
{

}


/*****************************************************************
*   FUNCTION: SaveToFile                                         *
*   PURPOSE: Saves data in link list to data file empinfo.data   *
*   PARAMETERS: pHeadPtr, pCurrPtr                               *
*   LOCAL VARIABLES:  None                                       *
*****************************************************************/
void CEmployee::SaveToFile (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr)
{

}

/*****************************************************************
*   FUNCTION: AddEmp                                             *
*   PURPOSE:  Adds emmployees to link list.                      *
*   PARAMETERS:  None                                            *
*   LOCAL VARIABLES:  None                                       *
*****************************************************************/
void CEmployee::AddEmp(CEmployee *&pHeadPtr, CEmployee *&pCurrPtr )
{
   pCurrPtr = new CEmployee;
   pCurrPtr->m_pLink = pHeadPtr;
   pHeadPtr = pCurrPtr;
   cout << "\nEnter Employee Name: ";
   cin >> pCurrPtr->m_Name;
   cout << "\nEnter Employee's Age: ";
   cin >> pCurrPtr->m_Age;
   cout << "\nEnter Employee's Salary: ";
   cin >> pCurrPtr->m_Salary;
}
```

```
/*****************************************************************
*  FUNCTION: ExitProg                                            *
* PURPOSE: Exits program and deletes memory spaces that were being   *
*        used.                                                   *
*  PARAMETERS: pHeadNode                                         *
*   LOCAL VARIABLES:  pCurr, pDeleteThisNode                     *
*****************************************************************/
void CEmployee::ExitProg (CEmployee *pHeadNode)
{
   CEmployee *pCurr, *pDeleteThisNode;

   pCurr = pHeadNode;
   while ( pCurr != 0)
      {
      pDeleteThisNode = pCurr;
      pCurr = pCurr->m_pLink;
      delete pDeleteThisNode;
      }
}



/*****************************************************************
*   FUNCTION:  ListAllEmp                                        *
*   PURPOSE:  Lists all employees in link list to the screen.    *
*   PARAMETER:  pHeadPtr, pCurrPtr                               *
*   LOCAL VARIABLES: count                                       *
*****************************************************************/
void CEmployee::ListAllEmp (CEmployee *&pHeadPtr, CEmployee *&pCurrPtr)
{
}

/*****************************************************************
*   FUNCTION:  SearchEmp                                         *
*   PURPOSE:  Searches link list for specified employee.         *
*   PARAMTERS:  pHeadPtr                                         *
*   LOCAL VARIABLES: CurrPtr, Search_name, flag                  *
*****************************************************************/
void CEmployee::SearchEmp (CEmployee *pHeadPtr)
{
}
```