CSCI 2170 Lecture Notes on Array of struct

const int MAX_EMPLOYEE = 100;
employee allEmployee[MAX_EMPLOYEE];
what will the internal representation of allEmployee look like?

- **member access** for one array element: read in name of the 2nd employee
  getline(cin, allEmployee[1].name);

- print the 1st two characters of the 3rd employee
  cout << allEmployee[2].name[0] << allEmployee[2].name[1] << endl;

- **array iteration** : count the total number of dependents of all employee
  int sum=0;
  for (int i=0; numOfEmployee; i++)
  sum += allEmployee[i].numOfDependents;

- **pass array of struct to function**
  declare : find FindHighestRate(int numOfEmployee,
  employeeType allEmployee []);
  activation: FindHighestRate(numOfEmployee, allEmployee);
  definition:

```
float  FindHighestRate(employee allEmployee[])
{
    float highest = allEmployee[0].rate;
    int     employeeId = allEmployee[0].id;

    for (int i=1; i<numOfEmployee; i++)
    {
            if (allEmployee[i].rate > highest)
            {
                    highest = allEmployee[i].rate;
                    employeeId = allEmployee[i].id;
            }
    }

    cout << "Employee: " << employeeId << " has the highest pay rate : "
            << highest << endl;
}
```

Example:

```
// User enters Title, Author, and DateBorrowed for a list of books.
//   When no more books are to be entered:
//     Program displays all books sorted by author.
//     Program prompts user to enter an author's name and then
```

```
//       displays all data for each book written by this author.
// NUMBER_BOOKS is the maximum number of books in program.
//   MAX_STRING is the max characters in any string.  Library is
//   an array containing data for all the books.

#include <iostream>   // Header file for input/output
#include <string>     // Header file for strcmp.
Using namespace std;

const int MAX_STRING = 25;
const int NUMBER_BOOKS = 10;

struct date
{
  int month;
  int day;
  int year;
};
typedef struct date  dateType;

struct book
{
  char    title[MAX_STRING];
  char    author[MAX_STRING];
  dateType dateBorrowed;
};
typedef struct book bookType;

void EnterBooksInLibrary(int &numberBooksInLibrary, book library[]);
void SortByAuthor(int numberBooksInLibrary, book library[]);
void Swap (book& book1, book& book2);
void Display (int numberBooksInLibrary, book library[]);
void ListDataForThisAuthor (int numberBooksInLibrary, char thisAuthor[], book library[]);

int main ()
{
  int      numberBooksInLibrary;
  book   library[MAX_BOOKS];
  char    thisAuthor[MAX_STRING];

  EnterBooksInLibrary(numberBooksInLibrary, library);

  // Sort all books by author's name and display
  SortByAuthor (numberBooksInLibrary, library);

  DisplayBooks (numberBooksInLibrary, library);

  // Read author from user & display all data for each book
  //   written by this author.
  cout << "Please enter the name of the author you are interested in:  ";
  cin >> thisAuthor;
```

```
  ListDataForThisAuthor(numberBooksInLibrary, thisAuthor, library);

  return 0;
} // end main

void EnterBooksInLibrary(int &numberBooksInLibrary, book library[])
{
  int    i;
  char continueFlag;

  i = 0;
  continueFlag = 'y';
  while (i < MAX_BOOKS && (continueFlag == 'y' || continueFlag == 'Y'))
  {
          cout << "Enter book title: ";
          getline(cin, library[i].title);

          cout << "Enter book author: ";
          getline(cin, library[i].author);

          cout << "Enter date borrowed (month day year): ";
          cin >> library[i].dateBorrowed.month
                  >> library[i].dateBorrowed.day
                  >> library[i].dateBorrowed.year;
          i++;

          cout << endl << "Enter another book title (y/n): ";
          cin >> continueFlag;
  }
  numberBooksInLibrary = i;

  return;
}

void SortByAuthor(int numberBooksInLibrary, bookType library[])
// Description: Displays all books in Library sorted by Author.
// Precondition: None.
// Postcondition: Books are displayed, sorted by author.
{
          bool change = true;
          int     i;

          // Do bubble sort to put books in order by author.
          while (change)
          {
                  change = false;
                  for (i = 0; i < numberBooksInLibrary-1; i++)
                  if (strcmp (library[i].author, library[i+1].author) > 0)
                  {
                          change = true;
                          Swap (library[i], library[i+1]);
```

```
                }
        }
} // end Sort

void Swap (bookType& book1, bookType& book2)
// Purpose: Swap book1 and book2.
// Preconditions: None
// Postconditions: book1 and book2 have been swapped.
{
        bookType temp;
        temp = book1;
        book1 = book2;
        book2 = temp;
} // end Swap

void Display(int numBooksInLibrary, bookType library[])
// Description: print information of all books in the library
// pre-condition: numberBooksInLibarary is given, library contained book records sorted by
author
// post-condition: Information about all books is printed
{

  // Display all books
  for (int i = 0; i < numberBooksInLibrary; i++)
    cout << endl << "Title: " << library[i].title
        << "\t" << "Author: " << library[i].author
        << "\t" << "Date borrowed: "
        << library[i].dateBorrowed.month << "/"
        << library[i].dateBorrowed.day << "/"
        << library[i].dateBorrowed.year
        << endl;
}  // end Display

void ListDataForThisAuthor (int numberBooksInLibrary,
                                    char thisAuthor[],
                                    bookType library[])
// Purpose: Function lists all data for each book written by thisAuthor.
// Preconditions: library contains book info sorted by author name,
//                    thisAuthor's name has been entered by user
// PostConditions: All data for each book written by thisAuthor is displayed.
{
        bool   found=false;

        for (int i = 0; i < numberBooksInLibrary; i++)
        {
                if (!strcmp (library[i].author, thisAuthor))
                {
                    found = true;
                    cout << endl << "Title: " << library[i].title
                        << "\t" << "Author: " << library[i].author
                        << "\t" << "Date borrowed: "
```

```
                    << library[i].dateBorrowed.month << "/"
                    << library[i].dateBorrowed.day << "/"
                    << library[i].dateBorrowed.year<< endl;
                }
            }

            if (!found)
                cout << “Sorry, we do not have book from this author.” << endl;

} // end ListDataForThisAuthor
```