## 60-141
## Intro to Algorithms & Programming II  Winter 2014

# Lab #10: File Processing
## (Due at the end of the lab period or beginning of the next)

**Objective:** In this Lab you will practice using sequential writing and reading to and from a text (character based) file.

Background:.
To process sequential text files in C, we use the header file <stdio.h> where various functions for file processing are defined. A few of them are:

1. fopen (): to open a file,
2. fclose (): to close a file,
3. fscanf (): to read from a file, and
4. fprintf (): to write into a file.

### Work to do:
**PART – A**
**Writing into a Sequential File:**
In this lab you will be using the structure (as defined below) and implement an interactive program capable of prompting the user and storing 3 employee records into a file called "employee.dat".  Write a program called “Lab10b.c” that creates the file called “employee.dat”.  This file will also be used in the second part of this laboratory.

The format of the file would look like this sample (excluding the first line):

```
ID FIRSTNAME LASTNAME
10 john doe
20 mary jane
30 jim smith
```

Instructions: Write a C program called "Lab10a.c" to accomplish the following: (The skeleton code is given below for your convenience. Use the 3 functions specified to perform the input, printing and saving the records to the file.)

```
#include <stdio.h>
struct employee {
     char firstname[40];
     char lastname[40];
     int id;
};
typedef struct employee Employee;

/* Input the employee data interactively from the keyboard */
void InputEmpRecord(Employee *EmpList);

/* Display the contents of Employee records from the list */
void PrintEmpList(const Employee *EmpList);
```

```
/* Save the employee records from the list to the newly created text
file specified by FileName */
void SaveEmpList(const Employee *EmpList, const char *FileName);

int main() {
     Employee EmpList[3];
     InputEmpRecord(EmpList);
     PrintEmpList(EmpList);
     SaveEmpList(EmpList, "employee.dat");
     return 0;
}
```

## PART – B

**Reading from a Sequential File:**

Write a different C program called "Lab10b.c" that reads the file "employee.dat" containing the 3 records created in part A into an array called EmployeeList (of type employee, size 3). Then, use the WordCap(char *word) function to capitalize only the first letter of each of the first and last names of the 3 employee records. Then save the new records back to the "employee.dat" file, overwriting the old records.

Note: After you run program "Lab10b.c" you can use the command "cat employee.dat" to view the contents of the file and verify the results. The file now should look like this sample (excluding the first line):

```
ID FIRSTNAME LASTNAME
10 John Doe
20 Mary Jane
30 Jim Smith
```

## EVALUATION:

You need to show your instructor the complete programs at the end of this lab, or at the beginning of your next lab. The marks you will receive for this lab are made of two parts: Lab work marks 8 and **attendance marks 2**. **Total 10 marks**.

**Lab Work Mark**: You will be evaluated based on your solutions for the problems based on the following scheme:

**0 mark = No work done.**
**2 mark = Incomplete code / does not compile, with no/invalid documentation**
**4 marks = Complete running program with no/invalid documentation**
**6 marks = Incomplete code / does not compile, with proper documentation**
**8 marks = Complete running program with proper documentation**

## IMPORTANT:

ASK QUESTIONS IF YOU GET STUCK, BUT DO YOUR OWN CODE. ANY CODE SUSPECTED TO BE SIMILAR TO ANOTHER SUBMISSION WILL CAUSE BOTH SUBMISSIONS TO RECEIVE A ZERO MARK ON ALL LABS AND BE REPORTED FOR PLAGIARISM