School of Electrical and Information Engineering
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
ELEN2004 – Software Development I

ELEN2004: Software Development I

Laboratory Assignment No 6

Due Date: Monday May 3rd, 2015, By 07:59Hrs

Outcome

After the assignment is completed we should understand the following:

1. Independent development of functions.
2. Concepts of struct and record processing .
3. Packaging a set of object routines as a static library
4. The use of a simple Makefile to manage libraries, programs, etc
5. How to profile the performance and execution time of an algorithm.

Scheduled Tasks

Note: You do not have all the tools to complete the exercise today, but you have enough to plan and start it.
The rest of the tools you'll need will be covered in the next lectures.

The exercise of the laboratory assignment-5 involves reading an input file of records and constructing a *struct*{ } of information record for each line of input. The records are then stored in an *array/vector* of the same *struct*{ } data type. The array of record created will be subjected to various operations of sorting by a specific field, searching and printing the contents of a record.

Detailed Description of the Tasks

You are required to define a struct data type **struct Record{ }** that has the following fields;

1. StudId: a student Id of an integer type.
2. SurName: a student's surname of a string type.
3. FirstName: a student's surname of a string type.
4. Mark: a mark obtained by the student for a course as an integer type;

You are required to implement a library named **libStudRec.a** that maintains the following functions.

- **void readRecs(ifstream& inPutFile, Record recList[], int& numRead);**
  This routine reads the records from an ifstream *inPutFile* and populates an array of records *recList[]*. Note you may use < *vector* > type in place of an array. The number of records in the file is initially unknown. The actual number read is returned in the parameter *numRead*.

- **int compareRec(const int A, const int B);**
  This returns 1, 0 and -1 according to whether $(A > B), (A = B)$, or $A < B$ respectively.

- **int compareRec(const String& A, const String& B);**
  Same as the previous except that the items being compared are of string data type.

- **void swapRec(Record& rec1, Record& rec2);** The function swaps the contents of *rec1* with *rec2*.

- **void sortRecById(Record recList[], const int& numRead);** Returns the array or records sorted by *StudId* in ascending order. The sorting algorithm to use should be the bubble-sort.

- **int seqSearchById(Record recList[], const int& numRead, const Record& rec);**
  Searches the list of records in recList[], using sequential search algorithm, for a StudId given in *rec* and returns the position (index into recList[]) of the record if the record exists. Otherwise it returns "-1".

- **int binSearchById(Record recList[], const int& numRead, const Record& rec);**
  Same as in seqSearchById() except that it uses the binary-search algorithm.

- **void printRec(Record recList[], const int& start, const int& last);** This routine prints the records in Reclist[] starting from the index position *start* and ending at the index position *last* or the end of the recList[] which ever comes first.

Using the library **libStudRec.a**, carry out the following functions

1. Construct a program that reads the records contained in a file *Lab5InputData.txt* into an array, call it *ArrayOfRecords[],* in memory.

2. Sort the contents of *ArrayOfRecords[],* by the StudId

3. Print the first 10 records of the contents of *ArrayOfRecords[].*

4. Using the contents of your file (You will need to close and open your input file multiple times).

   - determine the average time it takes to search for students using your sequential search algorithm,
   - determine the average time it takes to search for students using your binary search algorithm.
   - Manage the development of the entire exercise with a *makefile*

The Deliverables

- Submit a zipped or a tarred file that includes the files for the .h file, your function files, the file of your your main program and the output text of your result. Please do not send a set of independent .cpp files.

- You DO NOT need to include the text of the pseudo-code developed for your functions.

Submission should be under the name of the **first student** of the pair of students that worked together.