

# University of Wollongong

School Computing and Information Systems

CSCI251/851

## Advanced Programming

Autumn 2017

### Assignment 2

( Due: Week 6 - 11:59pm Tuesday 4 April )

8 marks

#### Aim:

Developing programs via functional decomposition and pointers.

#### On completion you should know how to:

- design a program by decomposing the problem into small functional units,
- implement a program using pointers and dynamic memory,
- produce a database program for maintaining records.

#### Requirements:

You are provided with an incomplete database program for accessing an electronic phone book. Your task is to firstly convert memory management from static to dynamic memory and implement the main search function. Each record in the database contains the following information:

Phone Number	// 8 digit number
Family Name;	// up to 20 chars
Given Name;	// up to 20 chars
Street Number;	// number 1..2500
Street Name	// up to 30 chars
Suburb	// up to 30 chars
PostCode	// 4 digit number

The program is implemented in 3 files: **main.cpp** contains the text-menu based user interface. **ass2.h** contains public function prototypes. **ass2.cpp** contains the function definitions of the database program. Test data is provided in **phone.txt**.

**Step 1:** involves converting the program from static to dynamic memory. To do this, first convert the DB array into an array of pointers to PhoneRecords:

```
PhoneRecord *gPhoneRecs[cMaxRecs];
```

Then modify the ReadFile() function so that as records are read, dynamic memory is allocated to array elements using the new operator:

```
gPhoneRecs[i] = new PhoneRecord;
```

This statement should go just after the break statement within in the read file loop. Now convert all references to the PhoneRecord fields to '->'. e.g.

```
fin >> gPhoneRecs[i]->PostCode;
```

Also, complete the DisplayRecord() function for displaying records in the DB array five records at a time. (Make sure the format complies with that shown below.) Example:

```
Command: d
56391746   John Johns, 520 Park Lane, Oak Flats, 2532
14053398   Jill Jackson, 153 Seaview Dr, Mt Keira, 2211
70399355   Jack Hendly, 134 Parker St, Blacktown, 5933
83438529   Chris Smith, 788 Crest St, Albion Park, 2544
41878844   Joe Jones, 89 Keira Lane, Kanahooka, 2322
Display more records? y/n: <y>

92392252   Jack Bell, 707 Cherry Lane, Waterfall, 2233
93037935   Lauren Hendly, 557 Cope St, Kanahooka, 2322
50981291   Mary Hendly, 94 Crest St, Blacktown, 5344
93220958   Jenifer Parker, 415 Park Lane, Oak Flats, 2456
23339704   Mavis Johnstone, 904 Peace Dr, Mt Keira, 2211
Display more records? y/n: <n>
```

**Step 2:** involves implementing an AddRecord() function for adding a new record to the DB:

- a adds a new record to the file

Example:

```
Command > a
Adding a new Record to Database...
Enter phone number: 42219999
Enter family name:  Gates
Enter given name:   Bill
Enter street number: 123
Enter street name:  Silicon Dr
Enter suburb name:  Silicon Valley
Enter postcode:     9999
A new record has been added to the database
...there are 245 records in the database
```

**Note:** The new record should be both added to the gPhoneRecs[] array and the data file. To append the record data to the end of the data file use:

```
fout.open(filename, ios::app); // opens file in append mode.
```

**Step 3:** involves implementing a Search() function for searching the database. The search function should request the Phone number from the user and display the matching phone number on the screen. Example:

```
Command > s
Enter phone number: 85332209
Jill, Dell, 779, Parker St, Smithville, 2958

Command > s
Enter phone number: 88884444
Record not found!
```

Also, implement a CleanUp() function for deleting all memory in the DB array when the program ends to avoid a memory leak.

**Step 4:** (CSCI851 students only) Add an erase record menu item to main.cpp e.g.:

“(e) Erase record”

and implement this function in ass2.cpp. e.g.

```
Command > e
Enter phone number: 88884444
88884444 has been erased from the database
...there are 244 records in the database
```

**Marking:** Step 1 is worth 4 marks and Steps 2 and 3 are worth 2 marks each. CSCI851 students who fail to implement Step-4 will receive a 1 mark deduction. Further information on the marking criteria will be provided on moodle before submission. **You must demonstrate your program during week 6 or 7. Failure to demo on time will result in a 1 mark deduction for each week late.**

**Note:** Make sure your code continues to work after each step before proceeding onto the next step. If you are unsure on exactly what to do in each step do not hesitate to ask your tutor. (The onus is on you to resolve anything you find unclear or ambiguous.)

## Submit:

Before submitting your files, test your program with the input provided below and save the output to a file named “output.txt”. Submit your files using the submit facility on UNIX as shown below:

```
$ submit -u login -c CSCI251 -a2 main.cpp ass2.h ass2.cpp output.txt
```

where 'login' is your UNIX login ID (Note: CSCI851 should also submit to -c CSCI251).

Deductions will be made for untidy work or for failing to comply with the submission instructions. Requests for alternative submission arrangements will only be considered before the due date. An extension of time for the assignment submission may be granted in certain circumstances. Any request for an extension of the submission deadline must be made to the Subject Coordinator before the submission deadline. Supporting documentation must accompany the request for any extension. Late assignment submissions without granted extension will be marked but the marks awarded will be reduced by 1 mark for each day late. Assignments will not be accepted if more than three days late.

**Before you submit your code, run your program and enter the following input in response to the user prompts. Marks will be deducted if your program fails to run properly on the test input below when tested.**

```
r
d
y
n
s
79521743
a
42219999
Gates
Bill
123
Silicon Dr
Silicon Valley
9999
s
42219999
q
```