University of Wollongong

School of Computing and Information Technology

CSCI251/851 **Advanced Programming** **Autumn 2018**

Exercise 1 **Demo Required in your Week 2 Lab Class** **2 marks**

Submit your files by 11:59pm Friday Week-2

Intro to C++

Instructions: Complete the following exercises by writing the programs in appropriately named files (e.g. ex1-1.cpp, ex1-2,cpp, etc.). Please ensure you code complies with the C++ Guide Book available in the week-1 lecture notes on the subject's moodle website. Also, test that your files compile and run on banshee.uow.edu.au (UNIX). You may need to install ssh on your PC for this (see resource folder on moodle). You should demo your programs in your week-2 lab class to receive the 2 marks and submit via unix submit by 11.59pm Friday week-2 (see Resource section on moodle on how to submit via banshee). If you need more time, ask your tutor for an extension. (Failure to comply with these instructions may result in zero marks for this exercise.)

1. Loops, c-strings, Arrays and Functions

Write a program that reads a string comprised of a number of words and (1) prints the string in reverse order, (2) prints the words in reverse order. eg:

```
Enter words: This C++ stuff is cool
looc si ffuts ++C sihT
sihT ++C ffuts si looc
```

Your program must be comprised of a **`main()`**, a function called **`PrintReverseString()`** and **`PrintReverseWords()`** and must use a char array (c-string) for storing the words. Both functions are passed the string in a char array and should print the string or words in reverse order respectively.

2. Enums

Enumerated types are declared with the keyword **enum** thus:

```
enum typename {enumerator list};
```

. . . and are explained on page 22 of the C++ Guide Book in the week-1 lecture notes.

(i) Declare an **enum** type called **Month** for representing the months of the year:
(Jan, Feb, Mar, ... , Oct, Nov, Dec).

(ii) Write a function with the prototype:

```
void GetMonth(Month &Mth);
```

that asks the user to enter a month represented with an integer (1..12) and sets **`Mth`** appropriately. (Note: pass-by-reference is explained on page 14 of the C++ Guide book.)

(iii) Write a function with the prototype:

```
void PrintMonth(Month Mth);
```

that prints on the screen the name of the month passed to the function.

(iii) Write a **`main()`** for testing both functions.

3. Structs

Structs are explained on page 19 of the C++ Guide Book. To store a collection of related but unlike values use a **`struct`** eg

```
struct MyType                // "MyType" is a new type
{
     type field_name1;       // "type" can be any built-in
     type field_name2;       // or user defined type
     type field_name3;
     ...
};

MyType student;   // "student" is a variable of type "MyType"
```

Declare structs and enums according to the following specifications:

(i) **struct name:** **NameType**

fields:

```
Last name        // array of 20 chars
Initial          // char
First name // array of 20 chars
```

(ii) **enum name:** **GenderType**

enumerators:

```
eMale
eFemale
```

(iii) **struct name:** **AddressType**

fields:

```
Number           // int
Street           // array of 20 chars
Suburb           // array of 20 chars
PostCode         // int
```

(iv) **struct name:** **StudentType**

fields:

```
Name          // NameType
Gender        // GenderType
Address       // AddressType
```

(v) Write a function called **`GetStudentDetails()`** that is passed a **`StudentType`** by reference and gets student record data for it from the user. Note: Input checking is unnecessary.

(vi) Write a function called **`PrintStudentDetails()`** that is passed a **`StudentType`** and prints the student record data on the screen.

(vii) Write a main() function to test the above functions.

4. Files

Files and I-O streams are explained on page 24 of the C++ Guide. File I-O in C++ can be performed using filestream objects. Use:

```
ifstream fin;   // to read from a file
ofstream fout;  // to write to a file
fstream  fio;   // to read or write to a file (not used here)
```

Write two functions that can save or load a **StudentType** to or from a file.
Your functions should have the following prototypes:

```
void SaveStudentRecord(StudentType &StudentRec);
void LoadStudentRecord(StudentType &StudentRec);
```

Test these functions by writing an appropriate **`main()`** function.

Submit:

You are required to BOTH demo your work in the lab to your lab supervisor and submit your files via UNIX submit command by the due date. i.e.:

$ submit -u login -c CSC251 -a ex1 ex1-1.cpp ex1-2.cpp ex1-3.cpp ex1-4.cpp

where 'login' is your UNIX login ID.

Note: Both CSCI251 and CSCI815 should submit to "-c CSCI251". If your submission receipt has errors concerning main() being multiply defined, that's ok for this submission. Marks will be deducted for untidy work or for failing to comply with the submission/demo instructions. Requests for alternative submission or demonstration arrangements will only be considered before the due date. An extension of time for submission or demonstration may be granted in certain circumstances. Any request for an extension of the submission deadline must be made to the Subject Coordinator or your Lab Supervisor before the deadline. Late undemonstrated work without cause will receive zero marks.