FACULTY OF ENGINEERING AND INFORMATION SCIENCES

SUBJECT'S INFORMATION:			
Subject:	CSCI251 Advanced Programming		
Session:	July 2019		
Programme / Section:	Computer Science		
Lecturer:	Ms. Siti Hawa		
Coursework Type *(tick appropriate box)*	❑ Individual Assignment ❑ Group Assignment ❑ Project ✓ Lab Task ❑ Seminar / Tutorial Paper ❑ Others		
Coursework Title:	**Lab Task 3**	Coursework Percentage:	1%

ASSESSMENT CRITERIA:

All programs should produce the correct result as stated in the specification. Programs should be written only using the programming structures and concepts already covered during lectures. Meaningful identifiers used. Proper indentation and line spacing. Suitable comments are recommended. Output should be well formatted with appropriate messages displayed. Numbers are shown with appropriate precision. Programs with syntax error and are unable to execute will not be awarded any mark.

SUBMISSION:

All completed work should be submitted online through Moodle before the due date provided.

SUBMIT AS EARLY AS POSSIBLE. ONLY ONE SUBMISSION IS ALLOWED. IF RE-SUBMISSION IS NECESSARY, YOU ARE REQUIRED TO REMOVE THE EARLIER SUBMISSION AND THIS MUST BE DONE BEFORE THE DUE DATE. OTHERWISE YOU WILL BE PENALIZED FOR LATE SUBMISSON.

DUE DATE:	**WEEK 6**

PENALTIES FOR LATE SUBMISSION:

Penalties apply to all late work, except if student academic consideration has been granted. Late submissions will attract a penalty of 25% of the assessment mark per day including the weekend. Work more than (3) days late will be awarded a mark of zero.

PLAGIARISM:

When you submit an assessment task, you are declaring the following

1. It is your own work and you did not collaborate with or copy from others.
2. You have read and understand your responsibilities under the University of Wollongong's policy on plagiarism.
3. You have not plagiarised from published work (including the internet). Where you have used the work from others, you have referenced it in the text and provided a reference list at the end ot the assignment.

Plagiarism will not be tolerated. Students are responsible for submitting original work for assessment, without plagiarising or cheating, abiding by the University's policies on Plagiarism as set out in the University Handbook under University Policy Directory and in Faculty handbooks and subject guides.

COURSEWORK SPECIFICATION

OBJECTIVES:

Following completion of this task, students should be able to:

- Write C++ programs text files and binary files.

Question 1 (Formatted I/O)

You are responsible to maintain a simple sales record for the sales of a company. The company keeps a text file to store the records of the monthly sales made by the salespersons that contains the employee number, name, and sales made for each salesperson. The file may be formatted in the following way:

12345	Ali Baba	1040.00
12233	Sinbad	35.98
23781	Spongebob	199.00
67543	Cinderella	500.50
23781	Spongebob	1000.00
12345	Ali Baba	80.00
67543	Cinderella	67.50

Your task is to write a program that will allow new sales records to be written to the file and also to read the file and print out the total sales made by each salesperson together with the overall total sales made so far. Filenames should be entered by the user.

Declare suitable data structure to keep each record. Use individual functions to perform the calculations and display. Include necessary error checking in your program and provide a well-formatted output.

Question 2 (Formatted I/O and Unformatted I/O)

You are provided with a text file (chemicalElements.txt) containing a list of chemical elements. Each line on the text file will contain the atomic mass, followed by the element name, chemical symbol, and an integer atomic number. Example of a few lines from the file is as follows:

```
1.0079  Hydrogen    H    1
4.0026  Helium      He   2
6.941   Lithium     Li   3
```

Your task is to read the information for each of the element in this file and produce a binary file containing records of these chemical elements. Each record should be stored in a `struct` variable declared as:

```
struct ChemicalElement
{
      int atomicNumber;
      char name[100];
      char symbol[3];
      float mass;
};
```

Once the binary file has been created, allow the user to perform a search on the binary file by providing the element name or symbol (display all the details if found), display all the chemical elements' details that have atomic mass greater than a given value, display the details of all the chemical elements with the atomic number in between two given values, or display the details of a chemical element by giving the record number. The details of the chemical element will then be displayed on standard output.

Separate function should be written to:

1. Read records from the text file and write to binary file.
2. Search by name or symbol
3. Display data with atomic mass greater than the value passed to it
4. Display data between two given atomic number
5. Display data given a record number

All functions should be called from `main()`.
