

**FACULTY OF INFORMATICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT’S INFORMATION:** | | | |
| Subject: | CSCI124 Applied Programming | | |
| Session: | July 2014 | | |
| Programme / Section: | J766SENG (SE) / J766CS53 (MGD) / J766CS42 (DSS) | | |
| 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: | 2% |
| **ASSESSMENT CRITERIA:** | | | |
| Correctness | All programs should produce the correct result as stated in the specification. | | |
| Coding | Programs should use appropriate control structures and data structures correctly based on what have been covered in the class and stated in the specification. Necessary input validations should be done. | | |
| Readability | Appropriate comments are included. Meaningful identifiers used. Proper indentation and line spacing used. | | |
| Well formatted output | Output should be well formatted with appropriate messages displayed. Numbers are shown with appropriate precision. | | |
| **SUBMISSION:** | | | |
| All completed work should be submitted online through Moodle before or on the due date provided.  **SUBMIT AS EARLY AS POSSIBLE. YOU CAN RE-SUBMIT LATER IF NECESSARY. ONLY THE LATEST SUBMISSION WILL BE MARKED.**  **IF YOU SUBMIT YOUR ASSIGNMENT TWICE, ONE SUBMMISSION BEFORE THE DUE DATE AND ANOTHER AFTER THE DUE DATE, THEN YOU WILL BE PENALIZED FOR LATE SUBMISSON.** | | | |
| DUE DATE: | **WEEK 5** | | |
| **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:**

The aim of this lab is to provide you with some experience in writing programs using binary files. You will also gain some experience working with makefile.

**TASK 1:**

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, or display the details of all the chemical elements with the atomic number in between two given values. 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

Function 1 be written in a file named fileOperation.cpp and functions 2, 3 & 4 should be written in a file named chemical.cpp. Create suitable interface file for each of the implementation file stated earlier. All functions should be called from main(). The main() function should be written in a separate file named main.cpp.

**TASK 2:**

Your second task is to write a suitable makefile for the application done in Task 1. The makefile should be written to create an executable named prog. Include also a clean target in your makefile to remove all the targets created earlier.

Test your makefile with no targets created at the beginning and observe the compilation process done. Then make some changes in one of the implementation file above. You can do this by simply adding a cout statement. Then run the makefile again and observe the difference between this compilation process and the previous one. Write down your findings in a text file by including the screen shot of the terminal screen when you run your makefile, and submit this together with your makefile.