

**FACULTY OF INFORMATICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT’S INFORMATION:** | | | |
| Subject: | CSCI204 Object and Generic 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 10** | Coursework Percentage: | 1% |
| **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 16** | | |
| **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:**

In this lab task, you will experience how to write programs using exceptions. You are also exposed to drawing use case diagram and class diagram with associations and multiplicities.

**TASK 1:**

You are required to create an application that performs the following tasks:

* Write a function that opens a text file and reads its contents into a stack container. Then extract the characters from the stack and save them in a second text file. The order of the characters saved in the second file should be the reverse of their order in the first file.

* Write another function that opens a text file and reads its contents into the STL deque container. Then it should convert it to uppercase, and store it in a second file in the same order as the first file.
* Lastly, write one more function that opens two text files and reads their contents into two separate STL vector containers. Then it should determine whether the files are identical by comparing the characters in the vectors. When two nonidentical characters are encountered, the program should display a message indicating that the files are not same otherwise it should indicate that the files are identical.
* Create a suitable driver program to test all the functions above.

**TASK 2:**

The Australian Tax Office (ATO) is currently rewriting a piece of software. As part of the process the ATO needs to migrate a text file, which contains an integer as unique identifier (also known as a tax file number), and a string (a person’s full name which can contain white space characters) per line. Each tax file number is 'normally' associated with one name.

The file may look something like this:

654342 Fred E Smith

213233 Joe Blow

212221 Sarah T Brown

Assuming the appropriate headers are included; the file stream ins is opened for input and a map is created using the declaration,

map<int, string> taxrecord;

Assume the following typedefs exist.

typedef map<int, string>::iterator mapit;

typedef map<int, string> maptype;

(a) Write a function with the prototype

void populatemap(maptype& taxrecords, ifstream& ins);

Where the map taxrecords, is passed by reference and key/ data pairs are read from the stream ins one by one and inserted into the map. You should report any duplicate insertions into the map with an error message.

(b) Write a function with the prototype

mapit findrecord(maptype& taxrecords, int key);

where the map taxrecords is passed by reference and an integer key, key, is also passed. The function should attempt to find the key value in the map. If such a value is found, the function should return an iterator pointing to the matched pair, otherwise the value of the return value should be taxrecords.end().

Write a main() function to define and populate the taxrecord map and test the two

functions defined in (a) and (b).