

Data Structures and Algorithms

Semester 4 (2018/19)

## SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Information Technology

Diploma in Information Security and Forensics

Diploma in Financial Informatics

**Assignment**

**Duration: 14** to **28** January 2019 (Week 14 & 15)

**Weightage:** 20% of Module

**Individual/Team/Both:** Team of 2 Students

**Penalty for late submission: 10 marks per day**

**NO late submission shall be entertained after week 16.**

|  |
| --- |
| ***WARNING***  ***If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action will also be taken.***  ***Similar action will be taken for the student who allows other student(s) to copy his/her work.*** |

There are a total of 5 pages (including this page) in this specification.

1. **OBJECTIVE**

This assignment is to test your understanding of the key concepts in the Data Structures and Algorithms module and to evaluate your ability in the selection, design and implementation of appropriate data structures and algorithms in a C++ application.

1. **INTRODUCTION**

An application is to be developed to create a spell checker. Given words are checked against a dictionary which is set of words in a file. This dictionary must be loaded into an appropriate data structure at the start of the program.

You are to form a team of TWO (2) and implement the application using C++.

The application should allow a user to:

* spell check a word entered by the user
* spell check a file (in .txt or .csv format) specified by the user
* add a new word to the dictionary
* save the dictionary (with new words added)
* display all words in the dictionary that starts with certain letter(s) specified by the user

For a given word, the spell checker should specify whether the word is correct (i.e. found in the dictionary) or incorrect.

For a given text file, the words that are incorrect (i.e. not found in the dictionary) are displayed.

For both cases above, the display of the number of searches/comparisons done is optional.

A text driven user interface is adequate for the application, however, it must be user friendly.

**Note 1:** You are **NOT** to make use of the Standard Template Libraries (STL) from the C++ standard library for the implementation of your data structures and operations.

1. **BASIC REQUIREMENTS**

The application must:

* Make use of a **data structure**
* Demonstrate the **application of algorithms** learnt (e.g. searching)

The data structures and algorithms chosen should be appropriate for your application.

The suitability of your chosen data structure and algorithm, the complexity and user friendliness of the application will determine how well you score in this section.

1. **ADDITIONAL FEATURES**

In addition to the basic requirements, outlined above, you may implement **additional** feature that checks the type of error in the word.

Examples of errors:

* Deletion
* Insertion
* Substitution
* Transposition

For example, consider the following words:

COMPUTER COMPTER COMPUTXER COMPUDER COPMUTER

The first word is correct.

The second word has a deletion error (one letter is removed).

The third word has an insertion error (an extra letter is added).

The fourth word has a substitution error (one letter is incorrect).

The fifth word has a transposition error (2 adjacent letters are swapped).

**Only 2 features are necessary for maximum marks.**

1. **DELIVERABLES**
   1. A **report** in Microsoft word document format that clearly indicates:
2. Team members’ names, student IDs and group name
3. A brief description of the application
4. Roles and contributions of each member in the group
5. **Instructions** on how to run your application
6. Class Diagram
7. Description of data structures and algorithms implemented
8. Detailed explanation of why the data structures and algorithms are selected.
9. All relevant appendices (diagrams, screenshots, user guides, etc.) wherever appropriate
10. References for any non-DSA materials used in the report and/or application
    1. Folder containing all the necessary files (solution, .h, .cpp, .dat) to run your application.

**Note 2:** You are required to upload a zipped file to the DSA network folder [\\ictspace.ict.np.edu.sg\dsa](file:///\\ictspace.ict.np.edu.sg\dsa). Leaders are to upload the file to their respective folder. E.g. Name1\_Name2.zip

**Note 3:** A penalty of 10 marks per day after **28 January 2019, 9 am** will be applied for late submission.

* 1. A 20-minute presentation that has the following:
* Introduction
* Description of application
* Demo of Application
* Explanation of selection, design and implementation of data structures and algorithms used
* Q&A session

1. **MARKING SCHEME**

Assignment (20% of module):

**Report (20 marks)**

* Clarity and comprehensiveness of the document
* Explain and analyse clearly and comprehensively the selection and implementation of the data structures and algorithms

**Program (60 marks)**

* Basic Requirements (70%)
  + Section 3 of this document
  + Proper documentation and programming style
    - A blocked comment at the top of the class stating Team members’ name, group, student ID and any features that you would like to highlight specifically.
    - A blocked comment at the top of the each function, giving brief description of what the function does, including input parameters and return value, if any.
    - Good programming practices (indentation, good function/variable/attribute naming, etc.).
* Additional Features (30%)
  + Section 4 of this document
  + Documentation and programming style

**Note 5:** Unfamiliarity with any part of the program will cause penalties that will compromise the overall assignment grade for a student.

**Presentation (20 marks)**

* Give good introduction of the application.
* Give detailed description of the various functionalities of the application
* Explain clearly and comprehensively the choice of design and implementation of the data structures and algorithms.
* Able to provide good answers in response to tutor’s questions.

The student must be able to demonstrate understanding of the concepts of data structures and algorithms learnt, apply the understanding in the implementation of the application.

-- End of Document --