Curtin College-Bentley
DSA1002 Data Structure and Algorithm
Trimester 2, 2022

Assignment 1

Weight: 30% of the unit

Assignment Location: The assignment 1 is uploaded under **Assessments section (Assessment 2 – Assignment 1)** on unit Moodle page**.**

Answer Format. When you write an answer, clearly indicate the relevant question number/letter. Include your name and student ID at the start. Also add appropriate comments to code files to indicate author name and student ID. Detailed submission guidelines can be found below in section 3.

Timeframe. You have 14 days (336 hours) to complete and submit your answers, from **08:00am on 17th Aug 2022** until **08:00am on 31th Aug 2022** (UTC+8). You may schedule your work within this period. **However, late submissions are not allowed**.

Submission. Submit your answer document(s) to the "**Assessment 2 – Assignment 1**" area on Moodle under assessments section. You must verify that your submission was successful. Correctly submitting is entirely your responsibility.

Reference Material. This is an OPEN BOOK and OPEN COMPUTER assignment. You may refer to any written material, including your notes, course materials, books, websites, Unit Moodle page recordings etc. However:

- You must complete this assignment entirely on your own.
- You should answer all questions in your own words and code.
- You can use pseudo code and algorithms provide in the unit slides (Moodle page) for your implementation.
- During the assignment, you may not communicate with any other student regarding the test.
- During the assignment, you may not communicate with any other person in order to seek or receive an answer to any test question.
- Your answer document will be checked by text matching software for signs of cheating,collusion and/or plagiarism.
- The assignment questions have been designed such that any two students, working independently,should not produce the same answers.
- The coding part of this assignment can be submitted in either python/java.

1. Overall Assignment Description

In this assessment you will apply a detailed knowledge of data structure and algorithms to real-life application to have a better understanding of the details covered in the unit. Detailed description on each question and steps required to be performed can be found in the question description. You can use pre-built ADTs from your tutorial and workshops (should be appropriately cited). Do not use build-in programming language (Python/Java) ADTs, e.g. using built-in Python stack.

Question 1: (Mini Project)

This assignment is an extension of the code you implemented to Double Ended Double Linked List (**DEDLL**). You will be implementing a personal journal of multiple pages (say 25 maximum pages, check for this maximum size in your code). The writing code should be able to read an already stored (i.e., seralised and stored in file) a journal or make a new journal or edit an existing journal. You code should start (named as My Journal Organiser.py) in an interactive mode with following options:

1. Read journal (this will open further sub menu)
2. Start/edit a journal (this will open into further sub menu)

Sub-menu for option 1 (in the main menu)

1. Load a journal (load seralised journal stored as file), this will start loaded journal on page 1
2. Go to next page
3. Go to previous page
4. Close the journal (may be say details about journal and a good bye message)

Sub-menu for option 2 (in the main menu)

1. Make a new journal (ask for the name of journal and use that name while you store it in file)
2. Add new page (add page at end as last page)
3. Add page at star (add as 1st page)
4. Add in after a page number (also should be able to edit pages number all proceeding pages accordingly)
5. Edit exiting pages (update text written on particular page number, this also might need load the stored journal first.
6. Store new/updated journal (store new journal with the name taken as input in option 1 and edited journal with the same name as before)

*You can store some random story on each page of a journal, just to test functionality.

Helping Hint:

The edge node in the DEDLL can store a string type text and a page number along with necessary pointers to connect the DEDLL.

2 Project Report

A project report of minimum 8-10 pages should be submitted (pdf format) including following details:
Usage information:

- Introduction: describing basic introduction of your program (software).
- Dependencies: any libraries required to use the program (software).
- Terminologies and abbreviations used in the code
- Future directions: suggested future improvements

Class UML Diagrams:

- Complete UML class diagrams of the classes used for implementation

Comments on Code: it is suggested to add detailed comments in your code.

References: (if any) all materials should be referenced Chicago referencing style.

3 Submission

Submit electronically through Moodle unit page under assessments section ("**Assessment 2 – Assignment 1**").
You should submit a single file, which should be zipped (.zip) or tarred (.tar.gz). Check that you can decompress it on the lab/personal computers. Your work will be tested on lab/computer other than your PC so try to check your code on other PCs too. The file must be named DSA_Assignment_1_<student id>, use underscores instead of the spaces in the file name.
The file should contain following deliverables:

- Your code. This means all.java/.py files needed to run your program. Do include code provided to you as part of the assignment if that is required to run your program. Do not include .class files or anything else that is not required to recompile .java files.

- README file including short descriptions of all files and dependencies, and information on how to run the program.

- Your program (software) test harnesses. One of the easiest ways for us to be sure that your code works is to make sure that you've tested it properly.

- Documentation and Report for your code (Project Report)

Please verify that your submission is correct and not corrupted. You may make multiple submissions, only your last one will be marked. **However, late submissions are strictly not allowed.**

4 Marking Criteria

The assignment will be marked based on the following breakdown of the submission:

Code Demo: (40 Marks) Code should be demonstrated during the tutorial to achieve this requirement. The code should be appropriately written, as ADTs with comments.

Project Report: (30 Marks) A minimum 8-10-page report based on information describe in section 2.

Code Testing: (30 Marks) Code should be implementable, and testable with the test harness.

5 Academic Integrity

Please see the Coding and Academic Integrity Guidelines on unit Moodle page.

In summary, this is an assessable task. If you use someone else's work or assistance to help complete part of the assignment, where it's intended that you complete it yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your own original work. Further, if you do not reference any external sources that you use, you are committing plagiarism and/or collusion, and penalties for academic misconduct may apply.

Curtin college also provides general advice on academic integrity at https://www.curtincollege.edu.au/content/dam/navitas/upa/curtin/pdfs/academic-integrity-policy.pdf

The unit coordinator may require you to provide an oral justification of, or to answer questions about, any piece of written work submitted in this unit. Your response(s) may be referred to as evidence in an academic misconduct inquiry.