COMP5511: Principles of Data Structures

(Data Structures and Algorithms)

Instructor: Bipin C. DESAI, Section DD, Wed 17.50-20:10

BipinC.Desai@concordia.ca,

Lab Instructors

LAB DD DI	15:15 -- 17:05	Maryam Valipour	maryam.valipour@mail.concordia.ca	H927
AB DD DJ	20:30 22:20	Kevin Hareshbhai Jivani	kevinhareshbhai.jivani@mail.concordia.ca	H831

COURSE OUTLINE

Course Calendar Description: COMP5511

.Definition, use, and application of fundamental data structures and associated algorithms. Asymptotic analysis of algorithms. Storage management: arrays, strings, lists and trees. Data abstraction: stacks, queues, priority queues, sets, and tables. Searching and sorting. Programming techniques: designing classes for data structures.

Prerequisites: *COMP 5481*

COMP 5481 must be completed previously:or equivalent training or experience in Java programming

Specific Knowledge and Skills Needed for this Course:

Students taking this course are expected to have sufficient knowledge of the following topics. Should you have difficulties in any of these topics, you are strongly encouraged to review them before the DNE deadline.

Java, IDE, Unix based OS Use of Web technology

This course will emphasize data structures, their design, implementation and performance issues.

1. Introduction, lists, linked lists, stack, queue, and circular linked list and their implementation; ADT, Binary search; Trade offs
2. Recursion and its implementation; Call trees and traces
3. Time complexity: notations, significance, bounds, worst-case, average-case; time and space trade-offs.
4. Lists, Stacks, Postfix expression, Queue
5. Trees: binary tree, tree traversals, heaps and priority queues;
6. Sorting: Bubble sort, selection sort, heap-sort, replacement sort, merge-sort, quick-sort, radix sort; time complexity lower-bound for comparison-based sorting;
7. Hashing: load factor, clustering, hash functions, open hashing, closed hashing;
8. Memory management: first-fit, best-fit, fragmentation, coalescing, compacting garbage collection;
9. Graph and its representations, depth-first search, width-first search, topological sorting;
10. Review

Grade Distribution

1. Assignments: 40%

   There will be four to five assignments which will include both theoretical and programming parts..

2. Quizzes: 60%

   There will be an on-line quiz at the end of each class. These would be taken using the Course Manager System . Half of the class will take it in H823 and the other half in H827. The quizzes are open book open internet but no communications of any type. The time allocated to complete the quiz requires that one is prepared for it – one does not have a time to do a search – with or without chat-bots.Anyway the chat-bots are not the smart, yet!!

3. Mid-term or Final Examination: **none**

To pass the course, a student must obtain passing mark in each of the grading components: these being the assignments, projects and the quizzes,. Following the practice in the Dept. of CS, there is no hard and fast rule to convert a numerical percentage and the final letter grades in this course. Note: The traditional passing mark is 50%

Textbook

Data Structures and Algorithms in Java by Michael Goodrich, Roberto Tamassia, Michael H. Goldwasser Wiley. (any edition)

References

Any text book on Java: among others, the following is free;

Introduction to Programming Using Java, https://open.umn.edu/opentextbooks/textbooks/419

There are lot of sample code sand demos for data structures on line.

Class notes and Recorded lectures from previous offering would be posted a week ahead of the class to allow a blended flavour to the offering and make up for missed classes

Assignments

Students should be aware of the University's code of conduct (academic), especially the parts concerning cheating, plagiarism and the possible consequence of violating this code. Besides, you learn little from copying the work of somebody else and it will not get you ready to answer questions on exams.

Criteria used in evaluations

It is expected that each student have accessed, read and understood and would abide the code of conduct located at:

https://www.concordia.ca/students/academic-integrity.html

The Faculty of ENCS also expects all students to sign and abide by the spirit of the document on originality (the URL of it is given below). Please download the from, read it and understand it. We assume that you have downloaded, carefully read and agreed and virtually signing the form in question for each submission for grading.

https://www.concordia.ca/ginacody/students/academic-services/expectation-of-originality.html

- Design: the program should be constructed from coherent, independent, and decoupled functions. A function should usually access only its own parameters and local variables.
- Style: the program should be general-purpose and well-organized.
- Documentation and Layout: The documentation should consist of a well-annotated program and clearly formatted output. Helpful identifiers and a clear layout are part of documentation. The documentation should include the description of your design and the algorithm implemented.
- Correctness and Testing: the program should conform to the specification given in the assignment. This includes the proper handling of special cases and error conditions and the providing of correct results. The submitted test cases take into consideration special cases and error conditions.
- Efficiency: The program must implement the most appropriate method.
- Program-User Interface: The program should be easy to use.

Grading is meant to give students a feedback about the quality of their code. Student should expect a full mark only if the submission is excellent in all grading categories. Typically, only a minority of submissions is excellent in all these categories and students should see, from the lower mark, what to improve. The above categories are used in industry for code evaluation as well and we hope you want to be ready for industry. Your marker is being asked to feedback on the criteria which are unsatisfactory.

Each assignment/project may contain a theory part and a programming part. If there is a theory part, only one of the questions from it may be chosen for marking. The choice of the question to be marked is done randomly.

1. All assignments are to be submitted on-line using the Course Manager System. There is a penalty for late submissions and the system doesn't accept any late submission once the penalty becomes 100%.
2. You are advised to retain a copy of **all** your term work until you receive a final grade for the course.
3. Please backup all your computing work.
4. Most of the handouts will be in electronic file form from the course home page. Point your browser to:

   Course Manager System

   These files would be in one of the following formats: text format (*.txt) or secure PDF.

Assignments submission

There will be four-five assignments including a number of programming assignments. For **these assignments**, discussion is allowed, but each group of student must **independently and separately** prepare and hand in their programming solution. **Programming Languages** Students should Java (use *C* ++ may be tolerated if there is a compelling reason)

1. Source code (mandatory)
2. Executable file (mandatory)
3. Header files (mandatory)
4. Makefiles (if applicable)
5. Input file used for data (if applicable)
6. Output file (if applicable)
7. A READ-ME file indicating any special instructions for compiling and running your implementation: 14 ptit must also include your Student I.D, your name, course, section and the name of your instructor. Use only your ``official'' name: please **do not** use abbreviations or nick names; capitalize the usual ``last'' name. **Any** deviation from these labeling guidelines will be penalized (at least 20%).
8. Documentation is to be included in electronic form (PDF is preferred format, HTML would be tolerated). The program, its successful compilation and its correct execution and resulting outputs may be verified.

   Make sure you have your User ID and PW for the Course Manager System . Consult the Lab Instructor.

   In the first week please form groups of 4, select a group leader among the group members and fill in the details in CrsMgr: There is a feature to upload all members of a team to CrsMgr - this requires the forming of such a team off-line!. Alternatively you may use the CrsMgr to choose any group and join it and then vote for the group leader before the deadline. Once you have formed a group of the size you want, ‘lock’ it. An unlocked group may be used to add students who have no joined any group.

Please note that all course work is done in groups(teams). Each team consists of usually 3-4 students, from the same section. Select a group leader among the group members and fill in the details in CrsMgr. A team could decide to have a smaller number (including one) such team may 'lock' it in CrsMgr once the desired number is reached. It is likely that there would be drop-outs and the number of teams members could become smaller. If possible, the instructor may be able to re-align smaller team if requested.

Each member is responsible for the solutions for each assignment and all the projects. Each member of the team may agree to take on the responsibility of the completion of a well defined portion of the work, to be agreed by the team members.

Anyone who has not joined a group will be placed in an group with less than 4 members or assigned to a new group.

Note: It is likely that one or more members of your team may drop the course and or not contribute equally to the work. For the former, redistribute the tasks among the remaining team members; for the later, use the peer review to reflect the contribution of such members.

To improve the class participation and avoid disturbing the class, please turn off your mike and camera during the lectures. All other electronic devices during the classes and tutorials should also be turned off.

TENTATIVE SCHEDULE

- **Class 1,2:** Introduction, Time complexity: big-O (big-Theta) notation, time complexity vs. space complexity;
- **Class 2**, **3:** Recursion Lists, Pointers, Linked list, ADT, Binary search, Tradeoffs of implementations.
- **Class 3,4:** Lists, Stack, Postfix expression, Queue,
- **Class 5**, **6, 7:** Sorting: Bubble sort, selection sort, (heap-sort), merge-sort, (quick-sort); Heaps and priority queue, heap-sort Time complexity lower-bound for comparison-based sorting; Radix sort
- **Class 7, 8:** Tres, B-Tree, B+-tree, Indexing
- **Class 9, 10**: External Sorting, Pattern Matching
- **Class 11:** Graph and its representations, depth-first search, width-first search, topological sorting

- **Class 12:** *Memory management,* Misc.

Important Notes

- **In the event of circumstances beyond the instructor's or the University's control, the content and/or evaluation scheme in this course is subject to change**.
- You are advised to retain a copy of all your term work until you receive your final grade for the course.
- Please backup all your work.
- All programs must have adequate internal and external documentations.
- You should be aware of the University's code of conduct (academic) as specified in appropriate section of the Calendar, especially the parts concerning cheating, plagiarism, and the possible consequence of violating this code. No need to mention that you will learn little from copying others work and it will not help you prepare for the quizzes. It is expected that each student has accessed, read and understood and would abide by the code of conduct.

http://www.concordia.ca/info/currentstudents/academicintegrity/

- The Faculty of ENCS also expect all students to sign and abide by the spirit of the document on originality (the URL of it is given above). Each student must print it and submit it to the tutor once to cover all work for the course:

All submissions in this course must confirm to the spirit of "Expectations of Originality".

Submission format

A single submission is required for each group marked entity (assignment/project - except Assg. 0). Each submission would be in the form of a SINGLE file uploaded to CrsMgr.

Assignment upload

CrsMgr does not allow anyone to download a file submitted for group work by clicking on the link for the uploaded file on the Assignment Upload page. The group leader can delete/change the uploaded file before the deadline. It is expected that the group leader communicates this information to the team. The team can always ask the group leader to be replaced.

For any submission, when the file is uploaded, the user is displayed a message to acknowledge the upload of the file and the md5 digest of the uploaded file. CrsMgr also sends this md5 digest by email(look in the Spam/Trash folder). The user can compute the md5 digest to see if the upload was correct. In UNIX one can use the following command!!

```
md5sum   name_of_the_file_submitted
```

Peer Evaluation

For group work, it is expected that all members of the team would have contributed for the submission. Each group marked entity would require submission (to be uploaded by the group leader or a designate) to CrsMgr and each member of the team would do a peer evaluation of the other members in the team. The effort of each member of the team is evaluated by the other members of the team; so each member of a team **must** evaluate the contribution of the other members. THIS IS AN INTEGRAL PART OF EACH GROUP MARKED ENTITY (ASSIGNMENT/PROJECT)

The peer evaluation is meant to encourage and ensure the participation of the members of a group to the group work and discourage freeloading. The deadline for a peer evaluation is usually shortly after the deadline for the corresponding marked entity, However, a peer evaluation could be done anytime a groups marked entity requiring peer evaluation is posted. An early peer evaluation could be updated up to the deadline for the marked entity. The peer evaluation for each team mate is entered and submitted one at a time. The entry is displayed and could be updated before the deadline.

Health and Safety Guidelines

General health and safety instructions and available health and safety trainings can be found at: Safety Programs - Concordia University (https://www.concordia.ca/campus-life/safety/general-safety.html)

If you have health issue such as a common cold, flu etc. please consider either not to attend the class or if you do attend wear a proper mask.

PLEASE DO NOT ASK THE INSTRUCTOR TO ENTER A MISSED/FORGOTTEN PEER EVALUATION. THE FUNCTION IS ACCESSIBLE ONLY TO YOU.

There would be a penalty for late submission of any marked entity. Your grade depends on three factors: the score for the marked entity for the group, your average peer evaluation, and whether you did make one for the others in your team.

If you do not do a peer evaluation for a submission, you will not be assigned any credit for that submission! What it means is that YOU will get a ZERO. There will be NO exception to this rule.

Recommendation: Start work on your assignments/projects as early as possible! Do the peer evaluation as soon as a group marked entity (assignment/project) is posted. *.db