Data Structures

Information and Communications Technology

Course Number: CST8130

Contribution to Program: Vocational

Normative Hours: 75

Applicable Program(s):

0006X01FWO Computer Eng. Technology - Comp. Science

AAL: 3
Core/Elective: Core
Approval Date: 19/08/2013

0006X03FWO Computer Eng. Technology - Comp. Science

AAL: 3
Core/Elective: Core

Prepared by:
Rex Woollard
Course Professor

Pre-Requirements
CST8132

Course Professor

Approved by:
Andrew Pridham
Academic Chair, ICT

Approved for Academic Year:
2013-2014

COURSE DESCRIPTION

Learn to use a variety of data structures and associated processing algorithms to manage massive amounts of data. Explore concepts using Object-Oriented Programming (in Java). Implement some solutions using first principle coding and others using predefined Collection and Map classes. Master techniques to assess the relative performance of alternative solutions (Big-O analysis) and make well-reasoned choices of selected data structures.

RELATIONSHIP TO VOCATIONAL LEARNING OUTCOMES

This course contributes to your program by helping you achieve the following Vocational Learning Outcomes:

Computer Eng. Technology - Comp. Science 0006X01FWO

1 Diagnose, solve, troubleshoot, and document technical problems involving computing devices using appropriate methodologies. (T,A)
3 Participate in analyzing, planning, designing, and developing the architecture of computing devices and systems. (T,A)
6 Analyze, build, test, implement, and maintain applications. (T,A)
8 Articulate, defend, and conform to workplace expectations found in technology environments. (T,A)

Computer Eng. Technology - Comp. Science 0006X03FWO

1 Diagnose, solve, troubleshoot, and document technical problems involving computing devices using appropriate methodologies. (T,A)
6 Analyze, build, test, implement, and maintain applications. (T,A)
8 Articulate, defend, and conform to workplace expectations found in technology environments. (T,A)

T: Teach A: Assess CP: Culminating Performance

ESSENTIAL EMPLOYABILITY SKILLS

The course contributes to your program by helping you achieve the following Essential Employability Skills:

1 Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience. (A)
3 Execute mathematical operations accurately. (T,A)
4 Apply a systematic approach to solve problems. (T,A)
5 Use a variety of thinking skills to anticipate and solve problems. (T,A)
6 Locate, select, organize and document information using appropriate technology and information systems. (T,A)
7 Analyze, evaluate and apply relevant information from a variety of sources. (T,A)
10 Manage the use of time and other resources to complete projects. (A)

© 2013 Algonquin College of Applied Arts and Technology
### COURSE LEARNING REQUIREMENTS

**When you have earned credit for this course, you will have demonstrated the ability to:**

1. Use Big-O analysis to assess algorithm performance. Select optimum data structures to implement problem solutions, based on Big-O analysis.

2. Explain concepts about commonly used data structures and implementation issues associated with the Java programming language. Implement representative Java code to support the explanation.

### EMBEDDED KNOWLEDGE AND SKILLS

- For a given algorithm, analyze the pattern of loops and method calls to determine the Big-O.
- For a data structure, identify the Big-O associated with various access methods (best-case, worst-case and typical case).
- List the standard Big-O types in sequence and graph curves representing the progression of their performance.
- For a given problem, select suitable API Collection and Map classes (both SE and JavaFX) and associated API algorithms to achieve optimum performance, based on Big-O analysis.
- Through written explanations and diagrams, clarify the role of reference variables in managing independent objects; arrays of primitives; arrays of references.
- Explain how heap and stack memory allocation operates, with particular attention to issues of scope (visibility), life-time (persistence), available space, and implicit memory release (garbage collection versus clearing a stack frame).
- Explain Big-O issues associated with sorting and searching algorithms when applied to specific Collection and Map data structures.
- Explain the role of inheritance, interfaces, named inner classes and static inner classes in implementing abstract data types (ADT).
- Interpret generic syntax and describe the implementation details, in particular, the role of reference-to-object as the underlying mechanism.
- Use file I/O to load large amounts of data from secondary storage.
- Use parallel arrays to provide multiple sorted views of a common collection of objects.
- Explain the callback mechanism in Java. Write Java code that implements callback functionality using Comparator and Comparable interfaces.
- Linked List: Using diagrams and Java code, demonstrate processing used to maintain this fundamental data structure.
- For any given Java Collection or Map concrete subclass, explain or diagram its implementation structure; explain the implementation mechanism for iteration; identify algorithm limitations and associated Big-O implications.
- Explain how Stack is an example of an abstract data type (ADT) that can be implemented using a fundamental underlying data structure: either linked list or array.
- Explain how Queue is an example of an abstract data type (ADT) that is implemented using a fundamental underlying data structure: either linked list or array.
- Explain how binary trees and binary search trees can be organized using references to discrete nodes, or organized using an array.
- Explain the rationale for the use of B-trees, and diagram basic processing steps required to complete add operations.
- Explain the role of the Set interface, and how concrete subclasses are implemented, with particular attention to the data structure and performance implications.
Table 1: Learning Activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Research data structure topics of relevance on the internet, and in</td>
<td>Locate relevant topics using increasingly refined search criteria.</td>
</tr>
<tr>
<td>programming language-specific documentation.</td>
<td>Evaluate the quality and utility of the topics.</td>
</tr>
<tr>
<td></td>
<td>Navigate the vast Java API repository of documentation to find</td>
</tr>
<tr>
<td></td>
<td>relevant information.</td>
</tr>
<tr>
<td>4. Trace recursive program flow and write recursive algorithms to</td>
<td>Demonstrate fluent use of Eclipse and / or NetBeans.</td>
</tr>
<tr>
<td>implement selected data structures.</td>
<td></td>
</tr>
<tr>
<td>5. Write Java programs that organize data using data structures selected</td>
<td>Demonstrate fluent use of Eclipse and / or NetBeans.</td>
</tr>
<tr>
<td>from those listed above according to the standards and time frame</td>
<td>Embed JavaDoc comments in Java code and generate the resulting JavaDoc</td>
</tr>
<tr>
<td>published in problem descriptions.</td>
<td>documentation.</td>
</tr>
<tr>
<td>6. Use a debugger to trace program execution and correct program errors.</td>
<td>Demonstrate fluent use of Eclipse and / or NetBeans.</td>
</tr>
<tr>
<td>7. Use a debugger to probe the organization of data at run-time and</td>
<td>Demonstrate fluent use of Eclipse and / or NetBeans.</td>
</tr>
<tr>
<td>prove the correctness of a specific data structure.</td>
<td></td>
</tr>
</tbody>
</table>

**LEARNING RESOURCES**

No textbook required. Previous semester textbooks can be used as a partial resource (for example, Deitel, *Java: How to Program 9th Edition*, ISBN: 0-13-257566-3). Students will research topics using internet-based resources.

This course is part of the mobile (laptop) program initiative at Algonquin. Students are required to have a functioning laptop at all lecture and lab classes. The specifications for the required laptop and additional information about the mobile program initiative can be found at [http://www2.algonquincollege.com/mlearning](http://www2.algonquincollege.com/mlearning)

**Required Software**

- NetBeans IDE, freely available from [https://netbeans.org/downloads/](https://netbeans.org/downloads/)

**LEARNING ACTIVITIES**

During this course, you are likely to experience the following learning activities:

- classroom lectures
- live, computer-based activities during lectures, using your personal mobile computer which is correctly configured for the learning tasks
- software demonstrations
- online lectures
- online software demonstrations
- individual and paired laboratory work
- practical and reading assignments
- research of course-related material

The course consists of 3 hours of lectures (2 hours in-class and 1 hour online) and 2 hours of in-class labs per week. You will probably need to spend an additional 6 hours per week, on average, of your own time for assignments and study.

**Lectures**

Theoretical course material will be presented through online lectures and in-class lectures. The online lectures consist of multimedia material, some of which can be downloaded in a batch process; some of which can be viewed as structured web pages. The in-class lectures will be aided by use of computer projections, videos, demonstrations and brief lecture notes.

- Students are expected to work through the online lecture material and attend all of the in-class lectures. Attendance will be recorded.
- Students will be expected to find and read applicable material on the internet to be prepared to answer oral, written or online questions in
Students can seek additional help through the use of BlackBoard discussion groups and e-mail. Students are encouraged to ask questions during lectures and to consult with the professor on topics which they do not clearly understand. Professors will inform students, at the beginning of the course, of suitable times for consultations.

Labs

You will apply the lecture material to a series of exercises and assignments which are closely integrated with the current lecture materials. They will be marked.

- **Exercise**: These are small, guided learning activities that can often be completed in less than a couple of hours (the length of a lab period). Exercises are to be completed by each student independently, though you can always ask questions of other students and the lab professor.
- **Assignment**: These are larger activities that build on the material explored in earlier exercises. Some will be completed independently; some can be completed in a partnership. Submissions for these activities will include selected documentation (such as problem statement, test plan, memory maps, Big-O analysis). These will likely require additional time outside of regularly scheduled lab periods.

Observations About Success Strategies

- Your ability to complete the exercises and assignments successfully will directly correlate with your level of success on tests and the final exam. If you do the work yourself, you should be able to pass tests and exams.
- Exercises and assignments build on earlier work and are increasingly complex as the course progresses. If you fall behind in your work, you will face increasing difficulty catching up.
- You should seek advice and help from the professor and peers during the lab period, and through the use of BlackBoard discussion groups.
- Attendance will be recorded and can directly affect your final course grade. (see evaluation section).
- Students should perform initial analysis and design on the full assignments before their scheduled lab period in order to take advantage of the limited lab time.

### EVALUATION/EARNING CREDIT

<table>
<thead>
<tr>
<th>The following will provide evidence of your learning achievements:</th>
<th>This activity validates the following Course Learning Requirements and/or Essential Employability Skills:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Final Examination 30%</strong></td>
<td>- Explain concepts about commonly used data structures and implementation issues associated with the Java programming language. Implement representative Java code to support the explanation. - [CLR 2]</td>
</tr>
<tr>
<td></td>
<td>- Trace recursive program flow and write recursive algorithms to implement selected data structures. - [CLR 4]</td>
</tr>
<tr>
<td></td>
<td>- Write Java programs that organize data using data structures selected from those listed above according to the standards and time frame published in problem descriptions. - [CLR 5]</td>
</tr>
<tr>
<td></td>
<td>- Use Big-O analysis to assess algorithm performance. Select optimum data structures to implement problem solutions, based on Big-O analysis. - [CLR 1]</td>
</tr>
<tr>
<td></td>
<td>- Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience. - [EES 1]</td>
</tr>
<tr>
<td></td>
<td>- Execute mathematical operations accurately. - [EES 3]</td>
</tr>
<tr>
<td></td>
<td>- Apply a systematic approach to solve problems. - [EES 4]</td>
</tr>
<tr>
<td></td>
<td>- Use a variety of thinking skills to anticipate and solve problems. - [EES 5]</td>
</tr>
<tr>
<td></td>
<td>- Analyze, evaluate and apply relevant information from a variety of sources. - [EES 7]</td>
</tr>
<tr>
<td></td>
<td>- Manage the use of time and other resources to complete projects. - [EES 10]</td>
</tr>
<tr>
<td><strong>Programming In-lab Exercises and Lab Assignments 40%</strong></td>
<td>- Use a debugger to probe the organization of data at run-time and prove the correctness of a specific data structure. - [CLR 7]</td>
</tr>
<tr>
<td></td>
<td>- Explain concepts about commonly used data structures and implementation issues associated with the Java programming language. Implement representative Java code to support the explanation. - [CLR 2]</td>
</tr>
<tr>
<td></td>
<td>- Trace recursive program flow and write recursive algorithms to implement selected data structures. - [CLR 4]</td>
</tr>
</tbody>
</table>
Research data structure topics of relevance on the internet, and in programming language-specific documentation. - [CLR 3]

Write Java programs that organize data using data structures selected from those listed above according to the standards and time frame published in problem descriptions. - [CLR 5]

Use a debugger to trace program execution and correct program errors. - [CLR 6]

Use Big-O analysis to assess algorithm performance. Select optimum data structures to implement problem solutions, based on Big-O analysis. - [CLR 1]

Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience. - [EES 1]

Execute mathematical operations accurately. - [EES 3]

Apply a systematic approach to solve problems. - [EES 4]

Use a variety of thinking skills to anticipate and solve problems. - [EES 5]

Locate, select, organize and document information using appropriate technology and information systems. - [EES 6]

Analyze, evaluate and apply relevant information from a variety of sources. - [EES 7]

Manage the use of time and other resources to complete projects. - [EES 10]

Classroom Tests 30%

<table>
<thead>
<tr>
<th>Final Grade</th>
<th>Mark Equivalent</th>
<th>Numeric Value</th>
<th>Final Grade</th>
<th>Mark Equivalent</th>
<th>Numeric Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>90-100%</td>
<td>4.0</td>
<td>C+</td>
<td>67-69%</td>
<td>2.3</td>
</tr>
<tr>
<td>A</td>
<td>85-89%</td>
<td>3.8</td>
<td>C</td>
<td>63-66%</td>
<td>2.0</td>
</tr>
<tr>
<td>A-</td>
<td>80-84%</td>
<td>3.6</td>
<td>C-</td>
<td>60-62%</td>
<td>1.7</td>
</tr>
<tr>
<td>B+</td>
<td>77-79%</td>
<td>3.3</td>
<td>D+</td>
<td>57-59%</td>
<td>1.4</td>
</tr>
<tr>
<td>B</td>
<td>73-76%</td>
<td>3.0</td>
<td>D</td>
<td>53-56%</td>
<td>1.2</td>
</tr>
<tr>
<td>B-</td>
<td>70-72%</td>
<td>2.7</td>
<td>D-</td>
<td>50-52%</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td>0-49%</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FSP</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
PRIOR LEARNING ASSESSMENT AND RECOGNITION

Students who wish to apply for prior learning assessment and recognition (PLAR) need to demonstrate competency at a post-secondary level in all of the course learning requirements outlined above. Evidence of learning achievement for PLAR candidates includes:

- Portfolio
- Other

The nature of a PLAR assessment will depend on the prior experiences of the candidate. Some form of examination will be required, though this could take the form of an oral exam based on a portfolio. Alternatively, the examination could take the form of a formal written exam, comparable to the course final exam.

RELATED INFORMATION

The following information is course-specific:

Some of the learning in this course can occur in partnered activities. To be a participating member of a partnership, you need to be present at all classes. A participation bonus of up to 5% may be added to a student’s grade if they have achieved a high threshold of attendance, have participated in bonus activities, and have been credited by their partner with contributing to partnered lab work. The details of this bonus will be defined by the course professor in class.

The final exam is designed to be a comprehensive, summative evaluation instrument. The final exam mark may replace the earlier test and quiz marks when it is to the advantage of the student. If a student’s mark on the final exam (as a percentage) is better than the test/quiz summary mark (as a percentage), then the test/quiz mark summary mark will be eliminated, and be substituted with a prorated exam mark. This mark adjustment process applies only when a student’s attendance has reached the participation bonus threshold.

At least 50% or “D-” must be achieved in the combined grade for final exams for the exercise, assignment and bonus marks to be added into the final grade. (Students who have a failing grade on the combined tests and the exam will receive a grade of “F”)

The Information and Communication Technology (ITC) Department requires that all course assignments (homework exercises, laboratory work, projects, etc) be submitted using a standard which could be specific to one or more courses. Professors will ensure, that students are advised of the exact details of these course specific submission requirements. Professors will also post them online. Student submissions that do not meet the published submission standards may not be marked, and may incur a penalty of up to 100% of the submission mark.

All students are required to write the final exam. There are no provisions for “making up” a missed final exam. If, as a result of being off-track in your program or some unforeseen circumstance, you note that there is a scheduling conflict in your final exam schedule, it is your responsibility to alert your course professor no later than one week before final exams start, to allow for any special arrangements.

In the case of a documented emergency, the professor, in consultation with the Chair, will determine how the marks will be made up and/or final grade adjusted.

The following information is school/department-specific:

STUDENT ACADEMIC RESPONSIBILITIES

Each student is responsible for:

- Knowing the due dates for marked out-of-class assignments.
- Attending all classes and knowing the dates of in-class marked assignments and exercises.
- Maintaining a folder of all work done in the course during the semester for validation claims in cases of disagreement with faculty.
- Keeping both paper and electronic copies of all assignments, marked and unmarked, in case papers are lost or go missing.
- Regularly checking Blackboard announcements as well as one’s Algonquin e-mail account for important messages from both professors and college administration.
- Participating in on-line and classroom exercises and activities as required.
- Retaining course outlines for possible future use to support applications for transfer of credit to other educational institutions.

Harassment/Discrimination/Violence will not be tolerated. Any form of harassment (sexual, racial, gender or disability-related), discrimination (direct or indirect), or violence, whether involving a professor and a student or amongst students, will not be tolerated on the college premises. Action taken will start with a formal warning and proceed to the full disciplinary actions as outlined in Algonquin College Policies - HR22 and SA07. Harassment means one or a series of vexatious comment(s) (whether done verbally or through electronic means), or conduct related to one or more of the prohibited grounds that is known or ought reasonably to be known to be unwelcome/unwanted, offensive, intimidating, derogatory or hostile. This may include, but is not limited to: gestures, remarks, jokes, taunting, innuendo, display of offensive materials, offensive graffiti, threats, verbal or physical assault, stalking, slurs, shunning or exclusion related to the prohibited grounds.

For further information, a copy of the official policy statement can be obtained from the Student Association.

Violation of the Copyright Act

General – The Copyright Act makes it an offence to reproduce or distribute, in whatever format, any part of a publication without the prior written permission of the publisher. For complete details, see the Government of Canada website at http://laws.justice.gc.ca/en/C-42. Make sure you give it due consideration, before deciding not to purchase a textbook or material required for your course.

Software Piracy - The Copyright Act has been updated to include software products. Be sure to carefully read the licensing agreement of any product you purchase or download, and understand the terms and conditions covering its use, installation and distribution (where applicable). Any infringement of licensing agreement makes you liable under the law.
Disruptive Behaviour is any conduct, or threatened conduct, that is disruptive to the learning process or that interferes with the well being of other members of the College community. It will not be tolerated. Members of the College community, both students and staff, have the right to learn and work in a secure and productive environment. The College will make every effort to protect that right. Incidents of disruptive behaviour must be reported in writing to the departmental Chair as quickly as possible. The Chair will hold a hearing to review available information and determine any sanctions that will be imposed. Disciplinary hearings can result in penalties ranging from a written warning to expulsion.

For further details, consult the Algonquin College Policies AA32, SA07 and IT01 in your Instaguide.

The following information is College-wide:

Email
Algonquin College provides all full-time students with an e-mail account. This is the address that will be used when the College, your professors, or your fellow students communicate important information about your program or course events. It is your responsibility to ensure that you know how to send and receive e-mail using your Algonquin account and to check it regularly.

Centre for Students with Disabilities (CSD)
If you are a student with a disability, it is strongly recommended that you identify your needs to the professor and the Centre for Students with Disabilities (CSD) by the end of the first month of the semester in order that any necessary support services can be arranged for you.

Academic Integrity* & Plagiarism*
Adherence to acceptable standards of academic honesty is an important aspect of the learning process at Algonquin College. Academic work submitted by a student is evaluated on the assumption that the work presented by the student is his or her own, unless designated otherwise. For further details consult Algonquin College Policies AA18 [http://www3.algonquincollege.com/directives/policy/academic-discipline/] and AA20 [http://www3.algonquincollege.com/directives/policy/plagiarism/]

Student Course Feedback*
It is Algonquin College’s policy to give students the opportunity to complete a course assessment survey in each course that they take which solicits their views regarding the curriculum, the professor and the facilities. For further details consult Algonquin College Policy AA25 [http://www3.algonquincollege.com/directives/policy/course-assessment/]

Use of Electronic Devices in Class*
With the proliferation of small, personal electronic devices used for communications and data storage, Algonquin College believes there is a need to address their use during classes and examinations. During classes, the use of such devices is disruptive and disrespectful to others. During examinations, the use of such devices may facilitate cheating. For further details consult Algonquin College Policy AA32 [http://www3.algonquincollege.com/directives/policy/use-of-electronic-devices-in-the-academic-environment/]

Transfer of Credit
Students, it is your responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.

Note: College policies (previously called directives) are under review and redesign. The term directives is being retired. Students, it is your responsibility to refer to the Algonquin College Directives/Policies website for the most current information available at [http://www3.algonquincollege.com/directives/].