CPS222 - DATA STRUCTURES AND ALGORITHMS

Professor:	Russell C. Bjork	Spring Semester, 2015
Office:	KOSC 242 x4377 russell.bjork@gordon.edu	MWF 3:20-4:20 pm KOSC 125
Hours:	MWF 2:10-3:10 pm R 9-12 am	
Lab Instructor:	Joshua Noseworthy joshua.noseworthy@gordon.edu	Lab: Mondays 6-9 PM KOSC 244

PREREQUISITE: Computer Science 122

CATALOG DESCRIPTION:

Introduces analysis of algorithms; implementation of data structures; advanced methods for organizing data in primary and secondary storage; problem-solving strategies; recursion; parallel algorithms; continued development of algorithm analysis skills. Weekly laboratories will introduce C++, including use of templates and the Standard Template Library. Prerequisite: CPS122.

COURSE OBJECTIVES:

In general, the primary intention of this course is to familiarize you with a broad range of "standard" data structures and algorithms for both internal and external storage. A secondary intention is to introduce C++. Specifically, upon completion of this course:

1. You should be thoroughly familiar with each of the data structures listed in the schedule of topics below. This means that you should be able to:
   a) Describe each of the structures covered, using words and/or diagrams.
   b) Describe one or more typical applications for which the particular structure would be an appropriate data representation.
   c) Implement algorithms for creating and maintaining the structure. This would include such operations as insertion, deletion, traversal, lookup, and reorganization as appropriate.
   d) Derive time and space complexity figures for the various operations as a function of the number of elements in the structure.
   e) Select appropriate data structure(s) for various applications, drawing upon the repertoire you have developed in this course.
2. You should be familiar with the major algorithm-discovery strategies.
3. You should be able to write programs using C++.
4. Your programming skills should be strengthened through the lab and project work.

TEXT: Goodrich, Michael, Roberto Tamassia and David Mount. *Data Structures & Algorithms in C++* 2nd ed. (Hoboken, NJ: Wiley, 2011)

ON RESERVE: Stroustrup, Bjarne. *The C++ Programming Language*. (Special Edition) (Addison Wesley Longman, 2000)

COURSE TECHNIQUES AND PROCEDURES

As in previous CS courses, regular practice with evaluation will be the heart of the course. For each unit of material, you will be asked to read a portion of the text book and to do assigned homework problems, and to apply the material you have learned through laboratory assignments and/or a programming project.

Class sessions will include a discussion and amplification of the material in the text and the presentation of further examples and supplementary material. You should not expect to grasp everything presented in the text when you first read it; however, you should note areas that are unclear to you and be prepared to raise questions about them in class.

In contrast to previous courses that have used Java as the programming language, this course will use C++. C++ is a hybrid language, embodying an object-oriented extension to C, a procedural language, but can be used either as an object-oriented language or as a "better C". It is also the first purely-compiled language you will have studied in the CS curriculum. You will notice that very little lecture time has been allocated to coverage of C++; rather, laboratory sessions and programming projects will be used for this purpose. It is important that you bear in mind that learning this language is secondary to the objective of gaining familiarity with algorithms and data structures. If you leave the course having simply learned to program in C++, you will have failed to achieve the major goal of the course, regardless of what final grade you receive.

COURSE REQUIREMENTS AND EVALUATION:

1. You will be expected to read material from the textbook, as assigned in the schedule below. Reading assignments should be completed BEFORE the class hour in which the topic is discussed. Lecture presentations will assume that you have read the text, and it is expected that your participation in the class will reflect that fact. However, our classroom discussion will not rigidly follow the order of material in the text, nor will it be confined to material covered there.

2. Seven problem sets will be distributed during the semester, and will be due as shown in the course schedule. Note that these will be fairly substantial assignments; you would do well to work on the problems as the material is covered in class, rather than waiting until just before the set is due to tackle the whole assignment. Solutions to each problem set will be discussed in class on the due date and/or posted Blackboard after the set is graded. **Late homework will be accepted up until the time the graded homework is returned, for a 50% grade penalty, but <u>NOT</u> after that.** Problem sets will be worth 25% of the final grade.

   The tentative emphases of the various problem sets are as follows *(subject to change)*:

Set	**Emphases**
1	Preliminaries
2	Sequential Structures
3	Trees
4	Search Structures
5	B-Trees; Sorting
6	Graphs
7	Algorithms

3. Weekly laboratories will focus on gaining practical experience with the material covered in the book and/or in lecture. Lab assignments will be given out the Friday before lab, and **must** be read over carefully **before** coming to lab.

   For each laboratory, there will be a writeup to turn in. There may also be a quiz given at the start of the lab hour (based on your reading of the lab assignment and/or material referred to in it) and/or a quiz based on the work done in lab given at the start of class on the due date. Each laboratory writeup and quiz(zes) will be worth 2% of the final course grade (16% total for the 8 labs with writeups due).

   The following are the tentative emphases for the lab sessions *(subject to change)*:

Lab	Emphasis
1	Using C++
2	Pointers and Linked Lists
3	(Lab time used for work on Project 1 - no writeup or quiz)
4	Stacks and Queues
5	(Lab time used for work on Project 2 - no writeup or quiz)
6	Expression Trees
7	Binary Search Trees
8	(Lab time used for work on Project 3 - no writeup or quiz)
9	Hashing; Performance of Searching Algorithms
10	(Lab time used to work on Project 4 - no writeup or quiz)
11	Graph Representation
12	Huffman Encoding
13	(Lab time used to work on Project 5 - no writeup or quiz)

4. Five programming projects will be assigned, to be done in teams of two. Especially for the early projects, you must partner with someone whose C++ experience is similar to yours to ensure maximum learning (i.e. a neophyte should partner with a neophyte, not a guru!) These projects must be done in accordance with the handout "Guidelines for Computer Science Projects", which will be distributed with the first project. You are expected to read these carefully and comply with them exactly. The following are the tentative emphases for the projects, along with their relative value in the final grade computation.

Project	Emphasis	Value
1	Using C++	5%
2	Stacks; Infix and Postfix Expressions	5%
3	Threaded Binary Search Trees	5%
4	B-Trees	5%
5	Graph Operations	5%

Total		25%

5. A mid-term examination (worth 16% of the final course grade) and a final examination (worth 18%) will be given as shown in the course schedule. Each exam will assume

familiarity with material in the text, covered in lecture, and/or used in homework problems or projects. Exams will be open book (course text only), open notes.

6. Your final grade will be computed on the basis of a weighted sum of the items listed above.

Summary:	Problem Sets	25%
	Labs	16%
	Programming Projects	25%
	Exams	34%

		100%

To receive a passing grade in the course, it is necessary to have a passing average in each of the four major categories above. Provided this is true, the following are minimum guaranteed grades for the overall percentages indicated.

	93% - 100%: A	90% - 92.9%: A-
87% - 89.9%: B+	83% - 86.9%: B	80% - 82.9%: B-
77% - 79.9%: C+	73% - 76.9%: C	70% - 72.9%: C-
67% - 69.9%: D+	63% - 66.9%: D	60% - 62.9%: D-

POLICY STATEMENT ON EXTENSIONS AND INCOMPLETES:

Extensions of the due dates for homework or projects will be given in the event of extenuating circumstances (such as illness, personal emergency) IF you submit a brief written request to the professor as soon as possible after the circumstances arise. This request will be initialed (if approved) and will be returned to you. You must attach it to the piece of work for which the extension was granted.

A grade of Incomplete will be given without penalty IF you are unable to complete the course work by the last day of the term due to major illness or other similar emergency. Again, a written request should be submitted. Such a request will only be granted if you are substantially up-to-date with your course work and were making good progress in the course up to the time that the difficulty arose. Of course, you must complete all work for the course by the midpoint of the next semester in accordance with College policy.

A grade of Incomplete with a penalty of one letter grade to be applied in the final grade computation MAY be given if you are unable to complete all the course work for reasons other than those noted above. You must make a written request, and your progress in the course, class attendance etc. will be taken into consideration in determining whether to grant it. Again, you must complete all work for the course by the midpoint of the next semester.

ATTENDANCE POLICY:

Regular class attendance is expected of all students, and class attendance will be recorded. Absences from class will be classified as “documented” or “undocumented”. A documented absence is one where written documentation is submitted supporting an absence from class due to circumstances beyond the student’s control. An undocumented absence is any other absence, including one which could qualify as documented if proper documentation were submitted.

Students who have more than three absences (of any kind) during the semester should expect to see their final grade reduced by 1% for the lesser of the number of undocumented absences and the total number of absences over 3, and students who have more than 12 undocumented absences will fail the course automatically. Note that it is not necessary to document absences unless there are more than three total absences; for most students, this will avoid the need to submit documentation. A student who anticipates the need to miss more than three classes due to athletic competitions or other student activities should review the college's attendance policy on page 36 of the catalog, and should then discuss alternatives to class attendance with the professor at the start of the semester.

A student who is habitually late will have late arrival for class counted as a half absence for that class, and a student who sleeps through most or all of a given class session will be counted as absent for that class.

You may ask the professor to waive this policy for you if you earned an A in the prerequisite course, or if you have an A average in this course as of the mid-term exam. If you wish to take advantage of this exemption, you must so inform the professor. However, the attendance policy will be reimposed if your subsequent work deteriorates.

STUDENTS WITH DISABILITIES:

Gordon College is committed to assisting students with documented disabilities (see Academic Catalog Appendix C, for documentation guidelines). A student with a disability who may need academic accommodations should follow this procedure:

1. Meet with a staff person from the Academic Support Center (Jenks 412 X4746) to:
    a. make sure documentation of your disability is on file in the ASC,
    b. discuss the accommodations for which you are eligible,
    c. discuss the procedures for obtaining the accommodations, and
    d. obtain a **Faculty Notification Form**.

2. Deliver a Faculty Notification Form to each course professor *within the first full week of the semester;* at that time make an appointment to discuss your needs with each professor.

Failure to register in time with your professor and the ASC may compromise our ability to provide the accommodations. Questions or disputes about accommodations should be immediately referred to the Academic Support Center. (See also Grievance Procedures in Student Handbook).

TENTATIVE COURSE SCHEDULE (SUBJECT TO CHANGE)

Date	Topic(s)	Reading	Written Work Due
	Preliminaries		
W 1/14	Course Introduction; Introduction to C++		
F 1/16	Introduction to C++ (continued)	Java/C++ §1-9	
M 1/19	*Martin Luther King Birthday - no class*		
W 1/21	Recursion	§3.5	**HOMEWORK 1 DISTRIBUTED**
F 1/23	Recursion (ctd)		
M 1/26	Algorithm Analysis	§4.1-4.2	
W 1/28	Algorithm Analysis (ctd)	§4.3	**LAB 1; START PROJECT 1**
	Sequential Structures		
F 1/30	List Structures: Arrays, Matrices, and Vectors	§3.1, §6.1; Java/C++ §10	
M 2/2	Linked Lists; Pointers in C++	§3.2; Java /C++ §11-13	**HOMEWORK 1**
W 2/4	Doubly-Linked and Circular Lists	§3.3-3.4	**LAB 2**
F 2/6	STL Lists and Iterators	§6.2	
M 2/9	Stacks	§5.1	
W 2/11	Stacks (ctd); Queues and Deques	§5.2-5.3	**(Nothing due for Lab 3); PROJECT 1**
F 2/13	Queues and Deques (ctd)		
	Trees		
M 2/16	Trees and Forests; Tree Traversals	§7.1-7.2	**HOMEWORK 2**
W 2/18	Binary Trees; Binary Tree Representation of General Trees/Forests; Expression Trees	§7.3	**LAB 4**
F 2/20	Complete Binary Trees; Heaps; Priority Queues	§8.3-8.3.4, §8.1	
	Search Structures		
M 2/23	Maps; Binary Search Trees	§9.1, 10.1	
W 2/25	Threaded Binary Search Trees		**(Nothing due for Lab 5); PROJECT 2**
F 2/27	Hash Tables	§9.2	**HOMEWORK 3**
M 3/2	Review and Catch up		
W 3/4	**MIDTERM EXAM (THROUGH TREES)**		**LAB 6**

F 3/6-F 3/13	*Spring Break*		
M 3/16	Height Balanced Binary Search Trees; 2-3-4 (2,4) and Red-Black Trees	§10.4-10.5	
W 3/18	Height Balanced Binary Search Trees (ctd)		**LAB 7**
F 3/20	Skip Lists	§9.4	
M 3/23	Disk-Based Search Structures: B-Trees	§14.3	
W 3/25	B-Trees (ctd)		**(Nothing due for Lab 8); PROJECT 3**
	Sorting		
F 3/27	Internal Sort Algorithms	§11.1-11.2 §6.4 §8.3.5	**HOMEWORK 4**
M 3/30	Internal Sort Algorithms (ctd)		
W 4/1	External Sorting; How Fast Can we Sort?	§11.3	**LAB 9**
F 4/3-M 4/6	*Easter Break - no class or lab*		
	Graphs		
W 4/8	Graphs; Representations for Graphs	§13.1-13.2	
F 4/10	Graph Algorithms	§13.3-13.4	**HOMEWORK 5**
M 4/13	Graph Algorithms (ctd)	§13.5-13.6	
W 4/15	Graph Algorithms (ctd)		**(Nothing due for Lab 10); PROJECT 4**
F 4/17	Graph Algorithms (ctd)		
	Algorithms		
M 4/20	Sets Operations	§11.4	
W 4/22	Pattern Matching in Strings	§12.1, §12.3 (skim 12.3.2)	**LAB 11**
F 4/24	Algorithm Design Strategies	§12.4	**HOMEWORK 6**
M 4/27	Algorithm Design Stratgies (continued)	§12.2	
W 4/29	Algorithm Design Strategies (continued)		**LAB 12**
F 5/1	Parallel Algorithms	TBA	
M 5/4	Parallel Algorithms (ctd)	TBA	**HOMEWORK 7**
W 5/6	Review and Catch up		**(Nothing due for Lab 13); PROJECT 5**
	F 5/8 - 12:00-2:00 pm - FINAL EXAM		