Professor: Steve Brinton
Office: MacDonald 214 x4183 stephen.brinton@gordon.edu
Office Hours: MWF 11:25-12:25, MWF 2:10-3:10

Class Hours: MWF 1:00 – 2:00pm (KOSC 128)
Lab: Thurs 1:15 – 4:15pm (KOSC 118)
Class Website: http://www.math-cs.gordon.edu/courses/cps212/

PREREQUISITE: CPS 211
CATALOG DESCRIPTION:
Introduces the 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: CPS211.

COURSE OBJECTIVES:
In general, this course is intended to familiarize you with a broad range of "standard" data structures and algorithms using C++ as the programming platform so that you will be better equipped to serve the one and only King, Jesus Christ of Nazareth.

Detailed Course Objectives:
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.
2. You should be familiar with the major types of algorithms listed in the schedule of topics below.
3. You should be able to design and create programs using C++.
4. You should be able to select the appropriate data structure(s) and/or algorithms for the programming task at hand.
5. You should be a much stronger and more intelligent programmer.

TEXT: Ford, William & Topp, William. *Data Structures with C++ Using STL* (2nd ed.) (Prentice Hall, 2001).

ON RESERVE:
Weiss, Mark Allen. *C++ for Java Programmers* (Pearson Prentice Hall, 2004)

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 textbook and to do chapter review problems, and to apply the material you have learned through a laboratory assignment 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 may not 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++. Learning C++ will give you experience with a second programming language. Our textbook, book on-reserve, class time in lecture and laboratory, and programming projects will provide you with opportunity to grapple with the C++ programming language. In other words: little official emphasis will be put on learning the C++ language – this will be left primarily to your investigation. 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. 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 day before lab, and **must** be read over carefully **before** coming to lab.

The following are the tentative emphases for the lab sessions *(order and/or topic subject to change)*:
Lab Emphasis
1 Basic C++ Programming
2 Algorithm Development – substring matching
3 Extending a class
4 Recursive Programming
5 Pointers

6 Recursive Lists
7 Linked Lists
8 Stacks and Queues
9 Binary Search Trees
10 Binary Expression Trees (text Concordance)
11 Hashing
12 Performance of Sorting Algorithms

3. Five programming projects will be assigned. These projects must be done in accordance with the handout "*Guidelines for CPS212 Projects*", which will be distributed with/before the first project. You are expected to read these carefully and comply with them exactly.

4. A one-third examination (worth 10% of the final course grade), a two-thirds exam (worth 10%) and a final examination (worth 22%) 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 practice problems or projects.

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

Summary:
33%   Labs
25%   Programming Projects
10%   Exam 1
10%   Exam 2
22%   Final Exam (comprehensive)
100%

Grading Scale:

A: <=100% >=95%	A-: <95% >=90%	B+: <90% >=86%
B: <86% >=84%	B-: <84% >=80%	C+: <80% >=76%
C: <76% >=74%	C-: <74% >=70%	D+: <70% >=66%
D: <66% >=64%	D-: <64% >=60%	F: <60% >=0%

BEING RESPONSIBLE:
At times, the amount of work required may seem insurmountable – however it is expected that each student be responsible to strive to do his or her best. On this note, pilfering or borrowing code/answers is not acceptable. It will result in a failing grade and any further action deemed appropriate by the college.

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 and lab 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 and approved by the professor supporting an absence from class due to circumstances beyond the student's control (such as illness, family emergencies, etc.) An undocumented absence is any other absence. Students who have more than 3 undocumented absences will have penalty added to their final grade at the discretion of the professor. Students who have more than 9 undocumented absences will fail the course automatically.

Labs are special occasions for C++ students – the only way to be able to miss a lab is to have it well documented and for a very good reason. After the documentation is accepted, the lab must be made up to receive credit. An undocumented lab absence will result in missed credit for the lab and 3 undocumented absences added to the course total.

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

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 31 of the catalog, and should then discuss alternatives to class attendance with the professor at the start of the semester.

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 Grievance Procedures available from the ASC.