Foundations of Programming and Problem Solving II

Course Description:
Catalog description: A continuation of COSC 5311 completing the syntax of the language used as the programming tool in COSC 5311. An introduction to data structures in multiple computing platforms. (Does not count toward total hours required for MS in Computer Science.) Prerequisite: COSC 5311.
Detailed description: This course is a continuation of COSC 5311, completing the syntax of the language used as the programming tool in COSC 5311 and providing an introduction to elementary data structures. It includes the intermediate study of the basic concepts of problem solving using C++ on a UNIX platform. Concepts include basic one- and two-dimensional array handling, classes, recursion, basic searching and sorting algorithms applied to linear dynamic data structures, and an introduction to dynamic data structures. Emphasis will be on developing high quality programs of intermediate length (several hundred lines). This course does not count towards the total hours required for an MS in Computer Science.

Prerequisites:
COSC 5311 is a prerequisite I expect you to be able to use Unix for your programming environment and write well-documented code using the C++ Style Guide posted on the course Web site.

Learning Objectives:
Upon successful completion of this course, the student will:
• Be able to use the UNIX operating system to manage and manipulate files and folders and to create, compile, and execute computer programs of intermediate length.
• Understand the syntax of and effectively use a high-level language to write computer programs of intermediate length including the use of functions, structs, dynamic memory allocation, and classes.
• Understand and effectively use single- and multi-dimensional arrays.
• Understand and effectively use linear and binary search algorithms.
• Understand and effectively use basic bubble, insertion, and selection sorting algorithms.
• Have a basic understanding of algorithm efficiency and be able to determine the Big-O efficiency of an algorithm.
• Develop, implement, and effectively use classes to create dynamic linear data
structures (list, stack, and queue).
• Understand and effectively use dynamic memory allocation to create and manipulate
variables within the scope of dynamic linear data structures.
• Have a basic understanding of hashing, quick sort, and prefix and postfix notation.
• Understand and effectively use basic recursive algorithms.

Assessment of Objectives:
Assessment of objectives will be conducted through exams, laboratory exercises, and
programming assignments.

Instructional Methods and Activities:
The methods and activities for instruction will include:
1. Presentation of new material and concepts in the classroom through the use of
lecture, tutorials, and sample programs.
2. Classroom and laboratory discussion using problem solving techniques.
3. Programming assignments to review and reinforce topics covered in the
classroom.
4. Optional one-on-one discussion as needed between the student and instructor
outside regularly scheduled class time.

Course Grades:
Programming Assignments: 35%
Midterm Exams (2): 30% (15% each)
Final Exam (comprehensive): 35%

Grade Ranges:
A: 90 - 100%
B: 80 - 89%
C: 70 - 79%
D: 60 - 69%
F: <60%

Please note: NO test grades will be given out via email or over the phone. Don't even
bother asking.

Texts: Starting Out with C++, Seventh edition, by Tony Gaddis

Lab Supplies: Flash drive to archive your programs (optional - but strongly
recommended)

Other Course-related Material:
All lecture notes, sample programs, programming assignments, and data files are online
at www.sci.tamucc.edu/~ptedford.
Class Policies:

Computer Security Issues:
It is your responsibility to read, understand, and follow the Student Security Statement (www.sci.tamucc.edu/~ptedford/MiscPages/StudentSecurityStatement.pdf) handed out the first day of class that discusses computer security issues. Infractions will have a direct affect on your grade in this course.

My Expectations of You:
- Come to lecture and lab on time every day the class meets
- Read the chapter to be discussed before coming to class
- Participate in discussions
- Answer the Checkpoint and Review Questions in each chapter
- Ask questions of material you do not understand
- If I cannot explain the answers to your satisfaction, make an appointment with me to discuss the question
- **DO YOUR OWN WORK!!** Do NOT share your work with others.
- Seek help, if needed, from the Program Assistants in CI 346A.
- Demonstrate integrity, maturity, and ethical behavior

Attendance:
If you want to succeed in this course, your attendance and active participation is crucial. Attendance is taken every day the class meets. If you are not in the room and in your seat before the lecture starts, you will NOT be counted as present that day. This may have serious implications on any financial aid you might be receiving. Although grades will not be directly based on attendance, regular attendance and active participation increases your chances of successfully completing this course. You are expected to know all material presented in class. **Turn off all cell phones when you enter the classroom! If you are caught texting, surfing the Web, playing games on your cell phone or laptop, or otherwise participating in distracting behavior, YOU WILL BE TOLD TO LEAVE THE CLASSROOM for the remainder of that class period.** Such behavior will **NOT** be tolerated.

Preparing for Lecture:
Class topics will follow the order of topics given in the schedule posted on the course’s Web page. You should read ahead to prepare for each class. Be prepared to study 2 - 3 hours for every hour you spend in class.

Class Problems:
You may be asked to work in groups in class to solve problems similar to those that will appear on exams. You are expected to actively participate in these activities. In addition, you may occasionally be asked to write answers to group problems on the board.

Academic Integrity/Plagiarism:
University students are expected to conduct themselves in accordance with the highest standards of academic honesty. Academic misconduct for which a student is subject to penalty includes all forms of cheating, such as illicit possession of examinations or examination materials, falsification, forgery, complicity or plagiarism. (Plagiarism is the presentation of the work of another as one’s own work.) This includes NOT sharing code or answers for the individual lab assignments! In this class, academic misconduct or complicity in an act of academic misconduct on an assignment or test will result in sanctions ranging from being given a zero for the assignment(s) involved to being given an F for the course. For additional information, see the “Assignments” section under “Lab Policies”.

**Assignments:**
Lab work will be assigned on a regular basis. Please refer to the "Lab Policies" section below for specific information and instructions about the lab assignments.

**Exams:**
You MUST read the text to do well in this class. As much as one third of the material on the tests may be information in the text not discussed in class. Exams may contain short answer, hand tracing, and/or programming questions.

**Make-up exams:**
Makeup exams will not be given under normal circumstances. If you notify me immediately that serious, unavoidable, documentable (e.g., with a letter from your doctor) circumstances have arisen, I will discuss options for replacing the missing grade. For example, I may allow the grade earned on the comprehensive final to replace the grade for the missed exam. You must take at least one of the midterm exams in this course. In other words, only one midterm exam may be missed due to unforeseen circumstances. Excused absences due to school sponsored activities, religious observations, family rituals, etc. should be discussed with me in advance.

**Lab Policies:**

**Attendance:**
Lab attendance is not required. However, if you would like some one-on-one assistance, please feel free to attend the lab. If you want to surf the Web, play video games, or watch videos on YouTube, do it on your own time. Don't do it in lab.

**Assignments:**
A list of assigned lab work and the criteria on which the labs will be graded is available from the lab assignments list on the first day of class. The assignments will be similar to those assigned to the undergraduates in COSC 1436. However, since this is a graduate course, each of your assignments will have additional components. While there are due dates (see below), there is no actual "assigned date" for each lab. That is, you may start working on the lab assignments immediately. Here are some guidelines you need to follow in order to do well on the labs:
• The assignments are to be completed individually. You may ask each other for general advice, but do NOT share final answers and/or source code unless you have been told to do so by me. Be sure to protect your programs. **Sharing or giving your work to others is grounds for an immediate "F" for the assignment involved. Submitting another person’s work as your own is grounds for an immediate "F" in the course.**

• Plan on spending 5 - 10 hours working on your lab assignments **outside of the scheduled lab time.** You cannot learn and understand the material by simply sitting through lectures. And I guarantee you will not be able to complete an entire lab assignment in the scheduled lab time.

• **Be sure to keep backup copies of ALL your programs!** Storage media have been known to fail. **Not having a backup copy of your work is NOT an acceptable excuse for submitting a late lab.**

• Keep in mind that when we grade your programs, we open the source code in Microsoft Word to comment/critique it. Save yourself some points (and us some frustration) by **not** extending your source code beyond column 73 in the text editor you use to type your code. Anything extending beyond column 73 tends to result in “word wrapping” when opened in Word. Word wrapping makes your code difficult to read and, as such, will result in your losing points unnecessarily.

• Do not resubmit a lab until you have received a grade for the previous submission. For example, if you turn in lab 6, but know it was not complete, wait until you get the results back before resubmitting it again.

• Each assignment is to be submitted using the dropbox as outlined on the dropbox instructions page. Lab assignments will **not** be accepted for grading if they are submitted via email, as a "hard copy," or on disk. **If you have problems using the dropbox, ASK FOR HELP!**

**Assignment Due Dates:**
Assignments are to be submitted by 11:59 PM on the due date. Due dates are listed with each assignment. If you want an A on any lab, **it must be submitted on time.** If you are submitting a lab that is less than complete, **you must have it checked by me to make sure it is an acceptable submission.** If you submit a lab late, your grade for that lab will drop according to the following schedule:

<table>
<thead>
<tr>
<th>Late Days</th>
<th>Highest Possible Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2</td>
<td>89</td>
</tr>
<tr>
<td>3 - 4</td>
<td>79</td>
</tr>
<tr>
<td>5 - 6</td>
<td>69</td>
</tr>
<tr>
<td>7 or more</td>
<td>50</td>
</tr>
</tbody>
</table>

No labs will be accepted after Tuesday, December 6, 2011, and there will be a lab due the final week of lecture. If an unavoidable, documentable emergency arises, please discuss it with me. However, under normal circumstances, if you have not completed your assignment by the due date, you should submit the **approved** work you have done for partial credit. For the sake of your grade, you should ALWAYS turn in SOMETHING that shows you’ve attempted to solve the problem. **Be sure to keep ALL graded material.**
Resubmitting Lab Assignments:
I allow you to resubmit lab assignments if you are not satisfied with the grade you have earned on a lab. However, there are a couple of "ground rules".

- The grade of a lab you want to resubmit must be less than a 90.
- The original version of the lab was submitted on time (that means by 11:59 PM on the due date). Labs that were submitted late the first time are NOT eligible to be resubmitted!
- The material originally submitted must be one of the following: 1) typed original pseudo-code (not pseudocode that was included in the original lab assignment), 2) the source code for a partially completed program that attempts to solve the problem, or 3) the source code for the program to solve the problem that will not compile and/or execute. In other words, you must have originally submitted something that indicates you made an honest effort to solve the problem. Submitting a ".cpp" file with your name on it and the words "Will resubmit later" does not count as an honest effort. Submitting only the answers to the UNIX exercise questions does not count as an honest effort. Submitting only the header and implementation files for a class does not count as an honest effort. If you are submitting a lab that is less than complete, you must have it checked by me to make sure it is an acceptable submission. If you do not have incomplete worked checked for admissibility prior to submission, you may not be eligible to resubmit that lab. All decisions about what is eligible for resubmission are final.
- Do not resubmit a lab until you have received a grade for the previous submission. For example, if you turn in lab 6, but know it's not complete, wait until you get the results back before resubmitting it again.
- Do not resubmit a lab unless you have made the changes/corrections necessary to improve your grade! Resubmitting a lab without making any corrections will not earn you any extra points!
- Only one lab per directory will be accepted. You may submit more than one lab at a time, but each must be in a separate directory.
- Be sure to include all files/documents that have been modified or corrected. This includes a list of what has been corrected/modified! Don't forget to include a new executable file, too!
- This is VERY important: FOLLOW THE INSTRUCTIONS FOR RESUBMITS IN THE DROPBOX INSTRUCTIONS VERY CAREFULLY!
- No resubmits will be accepted after 11:59 PM on Tuesday, December 6, 2011!

Classroom/professional behavior: Texas A&M University – Corpus Christi, as an academic community, requires that each individual respect the needs of others to study and learn in a peaceful atmosphere. Under Article III of the Student Code of Conduct, classroom behavior that interferes with either (a) the instructor’s ability to conduct the class or (b) the ability of other students to profit from the instructional
program may be considered a breach of the peace and is subject to disciplinary sanction outlined in article VII of the Student Code of Conduct. Students engaging in unacceptable behavior may be instructed to leave the classroom. This prohibition applies to all instructional forums, including classrooms, electronic classrooms, labs, discussion groups, field trips, etc.

**Academic Advising:** The College of Science and Technology requires that students meet with an Academic Advisor as soon as they are ready to declare a major. The Academic Advisor will set up a degree plan, which must be signed by the student, a faculty mentor, and the department chair. The College’s Academic Advising Center is located in Center for Instruction room 350, and can be reached at 825-6094.

**Dropping a Class:**
I hope that you never find it necessary to drop this or any other class. However, events can sometimes occur that make dropping a course necessary or wise. Please consult with me before you decide to drop to be sure it is the best thing to do. Should dropping the course be the best course of action, you must initiate the process to drop the course by going to the Student Services Center and filling out a course drop form. Just stopping attendance and participation WILL NOT automatically result in your being dropped from the class. Friday, November 5, 2011, is the last day to drop a class with an automatic grade of “W” this term.

**Grade Appeal Process:**
As stated in University Rule 13.02.99.C2, Student Grade Appeals, a student who believes that he or she has not been held to appropriate academic standards as outlined in the class syllabus, equitable evaluation procedures, or appropriate grading, may appeal the final grade given in the course. The burden of proof is upon the student to demonstrate the appropriateness of the appeal. A student with a complaint about a grade is encouraged to first discuss the matter with the instructor. For complete details, including the responsibilities of the parties involved in the process and the number of days allowed for completing the steps in the process, see University Rule 13.02.99.C2, Student Grade Appeals, and University Procedure 13.02.99.C2.01, Student Grade Appeal Procedures. These documents are accessible through the University Rules Web site at http://www.tamucc.edu/provost/university_rules/index.html. For assistance and/or guidance in the grade appeal process, students may contact the Office of Student Affairs.

**Notice to Students with Disabilities:**
Texas A&M University-Corpus Christi complies with the Americans with Disabilities Act in making reasonable accommodations for qualified students with disabilities. If you suspect that you may have a disability (physical impairment, learning disability, psychiatric disability, etc.), please contact the Services for Students with Disabilities Office, located in Driftwood 101, at 825-5816. If you need disability accommodations in this class, please see me as soon as possible.

If you are a returning veteran and are experiencing cognitive and/or physical access
issues in the classroom or on campus, please contact the Disability Services office for assistance at (361) 825-5816.

Course Outline (See online schedule for more details)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/25/11</td>
<td>Introduction; Review of Problem Solving</td>
</tr>
<tr>
<td>8/30/11</td>
<td>cstrings and the string class</td>
</tr>
<tr>
<td>9/1/11</td>
<td>File operations</td>
</tr>
<tr>
<td>9/6/11</td>
<td>File operations (cont.)</td>
</tr>
<tr>
<td>9/8/11</td>
<td>Arrays</td>
</tr>
<tr>
<td>9/13/11</td>
<td>Arrays (cont.)</td>
</tr>
<tr>
<td>9/15/11</td>
<td>Dynamic Memory; Pointers</td>
</tr>
<tr>
<td>9/20/11</td>
<td>Dynamic Memory; Pointers (cont.), More on cstrings</td>
</tr>
<tr>
<td>9/22/11</td>
<td>Review for Exam #1: structs</td>
</tr>
<tr>
<td>9/27/11</td>
<td><strong>Exam #1: Chapters 7, 9, 10, 12</strong></td>
</tr>
<tr>
<td>9/29/11</td>
<td>structs and arrays of structs</td>
</tr>
<tr>
<td>10/4/11</td>
<td>Big O Efficiency; Searches</td>
</tr>
<tr>
<td>10/6/11</td>
<td>Searches (cont.)</td>
</tr>
<tr>
<td>10/11/11</td>
<td>Sorts</td>
</tr>
<tr>
<td>10/13/11</td>
<td>Sorts (cont.)</td>
</tr>
<tr>
<td>10/18/11</td>
<td>Intro to Classes</td>
</tr>
<tr>
<td>10/20/11</td>
<td>Intro to Classes (cont.)</td>
</tr>
<tr>
<td>10/25/11</td>
<td>Review for Exam 2; Templated Classes</td>
</tr>
<tr>
<td>10/27/11</td>
<td><strong>Exam #2: Big O Efficiency, Chapters 8 and 11</strong></td>
</tr>
<tr>
<td>11/1/11</td>
<td>Singly Linked List</td>
</tr>
<tr>
<td>11/3/11</td>
<td>Singly Linked List (cont.)</td>
</tr>
<tr>
<td>11/8/11</td>
<td>Singly Linked List (cont.)</td>
</tr>
<tr>
<td>11/10/11</td>
<td>Singly Linked List (cont.)</td>
</tr>
<tr>
<td>11/15/11</td>
<td>Singly Linked List (cont.)</td>
</tr>
<tr>
<td>11/17/11</td>
<td>Dynamic Stacks</td>
</tr>
<tr>
<td>11/22/11</td>
<td>Dynamic Stacks (cont.)</td>
</tr>
<tr>
<td>11/24/11</td>
<td>Thanksgiving Holiday!</td>
</tr>
<tr>
<td>11/29/11</td>
<td>Dynamic Queues</td>
</tr>
<tr>
<td>12/1/11</td>
<td>Recursion</td>
</tr>
<tr>
<td>12/6/11</td>
<td>Review for Final</td>
</tr>
<tr>
<td>12/8/11</td>
<td><strong>Final Exam 4:30 – 7:00</strong></td>
</tr>
</tbody>
</table>