COP 3502 Syllabus
Computer Programming I – Summer 2011

Course Prerequisites: COP 3223 and MAC 1105
Class Time: TuTh 12:00 – 1:50 PM
Class Location: HPA-119
Course Web Page: http://www.cs.ucf.edu/courses/cop3502/sum2011/

Lecturer: Jonathan Cazalas
Office: HEC – 313
Email: jcazalas@eecs.ucf.edu
Office Hours: TuTh 10:00 AM – 12:00 PM

TA Office Hours: TBD, and will be posted on the course website

Course Description: from the UCF catalog description, “Problem solving techniques, order analysis and notation, abstract data types, and recursion.” Now, in English, this class is a follow up to the COP 3223 material, in which you learned (ideally) the syntax and use of major constructs of the C language (conditional statements, loops, functions, arrays, pointers, strings, structures, and file I/O). This course now focuses on algorithmic design, analysis of running time, a variety of abstract data types (new data structures), and lastly, but definitely not least important, recursion.

Course Textbook: <u>Data Structures, Algorithms & Software Principles in C</u> by Thomas A. Standish. ISBN-13: 978-0201591187.

Recitations/Lab Sessions: In addition to the main lecture, this course also includes recitation sections. One of the primary purposes of the recitation is to reinforce the information presented in the main lecture. Additionally, due to the amount of material that must be presented in this course, there will be times where new material is presented in the Labs and not discussed during the main lecture. For this reason, **<u>recitation attendance is mandatory</u>**, resulting in 10% of your final grade. Attendance will be taken, with each day counting as 1%. In order to **<u>earn</u>** your attendance point, you must be present throughout the lab (physically and <u>mentally</u>) and actively paying attention and/or working on the given problems. If you come more than 10 minutes late, leave early, or are playing on a Laptop, phone, game system, etc., you will not get credit for the lab. There will be 11 (possibly 12) lab sessions over the semester. You should go to ALL of them. However, once you have attended 10 labs, you will have received the full 10% (attending more will not earn you extra credit).

Programming Assignments: There will be **<u>six</u>** programming assignments. Each assignment will be introduced in class and posted on WebCourses. All homework assignments are to be turned in through WebCourses by 11:55 PM on the day they are due. The official time a program is submitted will be determined by WebCourses. (Your wristwatch or cell phone time when you hit the submit button is not valid.) Due to possible server issues, it is **strongly suggested** that you attempt to submit programs **<u>at least three hours before the actual time it's due</u>**. All programming for assignments is to be done in C and must compile and run using DevC++. Details on the usage of Dev-C++ will be provided.

Late Assignment Policy: Late homework assignments will be accepted for the first five homework assignments, but will be assessed a late penalty. In particular, assignments will be accepted up to 48 hours after the due date of the assignment. If an assignment is less than 24 hours late, a 10% penalty will be assessed. If an assignment is in between 24 and 48 hours late, a 25% penalty will be assessed. Assignments will not be accepted beyond 48 hours after the due date. ***No late assignments will be accepted on the last programming assignment.***

Community Service Opportunity: In lieu of the last assignment (program 6), you may perform 5 hours (or more) of community service. **If you take this option, then you will automatically get a 100 for program 6.** In order to get this credit, you must complete the community service and turn in the signed requisite form by **1:50 PM on Thursday, July 14th**. I strongly suggest you do this early to guarantee you do not miss the deadline; remember, **this is a FREE 100 for Program 6.** There will be NO exceptions to this deadline. All grades for the community service will be posted under the column P6 two weeks before the last program is due.

Exams: There will be two midterm exams and one final exam. Students will be allowed one page (8-1/2" x 11" or smaller) of notes for the two midterms and 2 pages of notes for the final. However, calculators will NOT be allowed for any of the exams. As the material in this course builds on itself, each exam can be considered “cumulative”, and material from the beginning of the semester is certainly not off-limits for the 2nd Midterm. And of course, the Final exam is cumulative as well.

Quizzes: Quizzes will consist of a small number of basic questions on material that has been covered recently, with the goal of forcing students to keep up with the material. Quizzes will be announced either in class or on Webcourses, and they will be administered via WebCourses. They will normally be due the following day by 11:55 PM with no late exceptions. If you miss class and, therefore, miss the announcement, and happen to have not checked your WebCourses, then you will miss the quiz. There will be no makeup quizzes. It is your responsibility to go to class and to check WebCourses for announcements.

Tentative Grading Procedures: The final letter grade will be based upon the six items listed below. Plus/minus grades may be issued, when deemed appropriate. The grading scale will be based on the class average, standard deviation, and overall difficulty of the assignments and exams.

Item	**Percentage**
Lab Attendance	10
Quizzes	10
Homework Assignments	30
Exam #1	15
Exam #2	15
Final Exam	20

Also, in order to pass the class you must earn at least a 40% on the final exam. (Thus, if you have a 75% in the course but earn a 30% on the final, you still get a C- in the course even though your percentage may qualify for a B.)

Important Dates:

First Midterm: Tuesday, June 14th
Withdrawal Deadline: Friday, June 24th by 11:59 PM
Second Midterm: Tuesday, July 12th
Community Service Due: Thursday, July 14th by 1:50 PM
Last day of class: Thursday, August 4th
Final Exam: Thursday, August 4th, last day of class. *

** The final exam is given during the last official day of class* ***at the regular class meeting time of 12:00 PM.*** *The exam will be structured to take no longer than the one hour and fifty minute time allotment.*

The UCF Creed

Integrity, scholarship, community, creativity, and excellence are the core values that guide our conduct, performance, and decisions.

Integrity
I will practice and defend academic and personal honesty.

Scholarship
I will cherish and honor learning as a fundamental purpose of my membership in the UCF community.

Community
I will promote an open and supportive campus environment by respecting the rights and contributions of every individual.

Creativity
I will use my talents to enrich the human experience.

Excellence
I will strive toward the highest standards of performance in any endeavor I undertake.

UCF Ethics Statement:

As reflected in the UCF creed, integrity and scholarship are core values that should guide our conduct and decisions as members of the UCF community. Plagiarism and cheating contradict these values, and so are very serious academic offenses. Penalties can include a failing grade in an assignment or in the course, or suspension or expulsion from the university. Students are expected to familiarize themselves with and follow the University's Rules of Conduct.

(see http://www.osc.sdes.ucf.edu/).

Other Important Course Policies:

1) **The TAs are your first, second, and even third points of** contact regarding the programming assignments. If you have any questions at all regarding the assignment, solving the program, how to code it, syntax errors, you name it, **contact the TAs**. There are plenty of TA office hours throughout the week, and I strongly suggest you take advantage of them. You can also email them with your questions, but understand that they may not respond immediately. If you want help via email, start your assignment early. Finally, the TAs will be grading the assignments. Therefore, any and all questions you have regarding your grade should be directed to them. If you feel your grade was unfair and you were not satisfied after contacting the TA, please come to my office hours to discuss.

2) Cheating will not be tolerated. **If a student is caught cheating, then the grade on that assignment for all students knowingly involved (the person providing answers as well as the one taking the answers) will be a -25%.** (**Note, this is less than 0%.**) Since discussion of concepts with other students is often helpful, cheating must be more clearly defined. In particular, the following items are cheating: copying a segment of code of three lines or more from another student from a printout or by looking at their computer screen, taking a copy of another student's work and then editing that copy, and sitting side by side while writing code for assignments and working together on segments of code.

   In all of these situations, **BOTH people responsible**, the one from whom the three lines of code are taken as well as the person who takes those lines of code are engaging in academic misconduct. For example, if someone makes an electronic copy of their code accessible to ANYONE in the class (except for themselves) before 48 hours after an assignment is due, they are automatically culpable of academic misconduct. It does not matter if the recipient of the code doesn't use it, uses it a little, or copies it directly. Furthermore, based on the severity of the case, the entire course grade for the student may be lowered an entire letter grade.

   If you get stuck on an assignment, please ask either a TA for help instead of getting help from another student. Part of the learning process in programming involves debugging on your own. In our experience, when a student helps another student with an assignment, they rarely allow the student getting help to "figure out" problems on their own. Ultimately, this results in a lack of debugging experience for the student receiving help. The goal of the TAs and instructors is to provide the facilitation necessary for students to debug and fix their own programs rather than simply solving their problems. **But, you are encouraged to work together on any non-graded programs to enhance and expedite the learning process.**

3) In order to take a make-up exam, you must request one from the instructor. The instructor will grant requests using his own judgment by applying the following general rule: "Make-up exams will only be given if the reason for missing the exam was out of the student's control." For example, being hospitalized unexpectedly is out of a student's control, but oversleeping or going to happy hour is not out of a student's control. ***If possible, it is recommended that the instructor be contacted before the exam.***

4) The penalty for assignments late by less than 48 hours will be waived only for circumstances described in #2 above at the instructor's discretion. Similarly, assignments will be accepted after 48 hours past the due date under these same types of circumstances. The student ***must*** contact the instructor within two days of the due date of the assignment in order to get credit for the assignment. ***TAs are NOT allowed to give extensions for assignments under any circumstances; only the instructor can.***

5) Both the course web page and WebCourses will be crucial elements of the course. ***It is your responsibility to check both of these before every class meeting for any updates that may be posted.*** Additionally, some clarifications may only be given in class and won't be posted online at all, so make sure you keep up with announcements in class.

6) **Class Attendance**. We may randomly take class attendance throughout the semester, which will count as bonus points on your Final Exam (up to five points). Most students have commented that this is one of their more enjoyable classes and that they enjoy and take benefit from the lectures. However, I still like to encourage attendance. The concept of class attendance is similar to lab attendance. We will take class attendance upwards of seven or eight times, with each recorded attendance counting as one bonus point, for a maximum of five bonus points. Thus, you can miss two or three "attendances" without being penalized.

7) **Grading Scale and Curving.** All UCF classes are ultimately graded on some type of grading scale, with the most popular being the famous "90 to 100 is an A, 80 to 89 a B, 70 to 79 a C", etc. In this course, we do **NOT** curve individual assignments, quizzes, or tests. **However**, at the end of the semester, we may choose to ultimately assign final letter grades based on a more "liberal" version of the aforementioned grading scale.

"What does this mean?"

For example, if the class average for tests was very low, resulting in a lower than normal final average, a final grade of 65 could most certainly be a C (instead of a typical D). Similarly, a 75 could be a B, or an 87 could be an A. **Do not quote me here!**

So for all intents and purposes, you can view this as a "curve" if you like. Additionally, the final grading scale is based on the class averages and specific grade distributions during the semester, which means that it can and does indeed fluctuate from semester to semester. Just like a professor cannot tell you, ahead of time, how much of a "curve" will be given on a test, similarly, I cannot tell you what the final grading scale will ultimately be. There are many variables that come into play when determining that scale.

So how should you treat this? To be safe, **assume** the normal grading scale. Meaning, assume that anything less than a 90 is not an A, or that anything less than 70 is not a C.

Regardless, it is the hope that your intention is to **care more about learning the material** than about your final grade. Those that do this will be fine!

Finally, it is our intention to grade as fairly as possible. You can be sure that you will get the grade you deserve.

Tentative Schedule for Lectures/Assignments

Week	**Tuesday Class**	**Thursday Class**	**Assignments**
May 16 – May 20	Lecture 1: Go over Syllabus + Intro PPT Slides Lecture 2: Linear vs Binary Search + Explanation of Program 1 C-Review: Read on your own.	Lecture 1: Sorted List Matching Problem + Review of Pointers Lecture2: Dynamic memory allocation for arrays	Assignment 1: BigInt I (aka "wake up call") (Due 5/25) Note: you only have ONE week for this program
May 23 – May 27	Lecture 1: Linked Lists Intro: Traversing a list, counting elements in a list, printing a list, and more Lecture2: Linked List operations: Insertion into a list	Lecture 1: Linked List operations: Deleting nodes from a list Lecture 2: Linked Lists Gone Wild: Circle and Doubly-Linked Lists	Assignment 2: Linked Lists (Due 6/8)
May 30 – June 3	Lecture 1: Recursion 1: Intro to Recursion, examples include count down, factorial, and Fibonacci Lecture2: Recursion 2: General structure, sum numbers, power, reversing a string, multiply, & Towers of Hanoi	Lecture 1: Recursion 3: Permutations + Info on Assignment 3 Lecture 2: Recursion 4: Recursive Binary Search & Fast Exponentiation	
June 6 – June 10	Lecture 1: Big-O notation, algorithm analysis, time complexity problems; use of summations Lecture 2: More Algorithm Analysis	Lecture 1: Summations Lecture 2: Even more Algorithm Analysis + Exam #1 Review	Assignment 3: Recursion (Due 6/22)
June 13 – June 17	**Exam #1**	Lecture 1: Time complexity using recurrence relations Lecture 2: Solving Recurrence Relations	

June 20 – June 24	Lecture 1: Stacks – applications, evaluation of postfix expressions Lecture 2: Use of stacks for infix to postfix; array and linked list implementation of a stack	Lecture 1: Queues: array and linked list implementation of a queue Lecture 2: Go Over Exam #1	Assignment 4: Stacks/Queues (Due 7/6)
June 27 – July 1	Lecture 1: Binary Trees, relation of height to number of nodes, tree traversals Lecture 2: Binary search tree, searching in a BST, insertion	Lecture 1: Deletion in BST Lecture 2: Various other binary tree functions	
July 4 – July 8	Lecture 1: Sorting- selection sort, insertion sort, bubble sort Lecture 2: Merge sort	Lecture 1: Quick Sort Lecture 2: Exam #2 Review	Assignment 5: Binary Search Trees (Due 7/20)
July 11 – July 15	**Exam #2**	Lecture 1: Heaps & Priority Queues Lecture 2: More Heaps & Heapsort	
July 18 – July 22	Lecture 1: Hash tables Lecture 2: Base Conversion Methods	Lecture 1: AVL Trees, insert Lecture 2: Go Over Exam #2	Assignment 6: Heaps or Hash Tables (tbd) (Due 7/27)
July 25 – July 29	Lecture 1: AVL Tree Insert (revisited) Lecture2: AVL Tree Deletion	Lecture 1: Backtracking Lecture 2: Introduction to Graphs	
Aug 1 – Aug 5	Final Exam Review	**FINAL EXAM on August 4th (Given during regular class time)**	