General guidelines

This course lays the foundations of computer science. It goes beyond “coding in C”. It will cover different ways to structure the data, and will show how each structure is useful in solving different types of problems. For each solution, we shall also examine the efficiency using a special mathematical tool.

This course involves extensive programming, which means besides attending the lecture classes you are required to spend substantial hours in the computer center labs working on your assignments. *You may fail this course if you do not submit all the assignments.*

Every student will be given an account on the Olympus. The TAs grading your programs would try to execute your code on Olympus using *gcc*, or they can try compiling using *jgrasp*. *If your programs cannot run on Olympus gcc or jgrasp, there is no way that the TAs would be able to award you any points for your work.*
Withdrawal guidelines

1. If you have no 'C' background and have never written any programs, you will find it extremely difficult to cope up with the requirements of the course. It is strongly recommended that you do not enroll simultaneously for COP3223 and COP3502.

2. If you cannot afford to spend time on doing the assignments, it would be in your interest to withdraw from the course.

Course Objectives

1  Provide an introduction to the field of computing: The central concept that underlies computer science is the design and implementations of algorithms to solve specific problems.

2  Provide Conceptual Content and Software Skills: The lecture component focuses on conceptual tools for constructing and analyzing algorithms – Time Complexity and recursion, while the lab component focuses on implementation issues involved in C programming.

3  Introduce elementary data structures: Arranging data in arrays, linked lists, stacks, queues, binary trees and hash tables.

4  Introduce searching and sorting techniques.

Lecture Classes:

As no single text book covers all the course contents, it is important that you take down notes during the lecture sessions. Some supplementary notes will be posted on the course webpage.

Recitation Sessions (Labs):

In the recitation sessions, the TAs (Teaching Assistants) will provide assistance in solving problems related to material covered in the lecture classes. They will also conduct pop-quizzes. It is very important that you do not miss any lab session.
Course Contents

1. Brief review of Pointers, arrays, dynamic allocation, files
2. Design of Algorithms for problem solving
3. Algorithmic Complexity - Big-O notation, summations
4. Searching - linear and binary search
5. Recursion - tracing, developing recursive functions, algorithmic complexity using recurrence relation
6. Stacks and Queues - converting infix expression to postfix form, evaluation of postfix forms, waiting line simulation, implementation using arrays
7. Linked Lists - Creation, deletion, insertion in sorted linked list, reversing, implementing stack and queue, circularly linked lists
8. Binary Trees - tree traversals, tree algorithms, Binary Search Trees, insertion and deletion, Balanced BST (AVL trees)
9. Heap trees - creation, insertion, heapify, deletion
10. Hash Table - collision resolving using linear probing, quadratic probing, dynamic hashing, double chaining
11. Sorting - Selection sort, Insertion sort, Bubble sort, Merge sort, Quick sort, heap sort
Reference Books:

Any book on Data structures will do for this course. The following book is available in the bookstore:

Data structures, algorithms & software principles in C
Thomas B. Standish
Addison - Wesley

Assignments:

1. You must submit all the assignments to pass this course.

2. All the programming assignments are required to be implemented in ‘C’. You will get an OLYMPUS account, and you can either use the computer labs at UCF, or dial up from home to run your programs on OLYMPUS. If you are using any other compilers make sure that your programs must compile under gcc or cygwin. You may not get any credits on an assignment if it does not compile under gcc or cygwin.

3. All assignments must be submitted through WebCT by 11 PM on the indicated due date. Assignments submitted after the due date would be accepted till the cut-off date with 10% penalty per day. No assignments would be accepted after the cutoff date.

4. Assignments must be carried out individually. Collaboration on any assignment is NOT acceptable. Cheating may result in a failing grade regardless of performance.
Exams:

There will be 3 mid term exams and one final exam. The dates for these exams will be announced shortly. In addition there may be number of pop-up Quiz tests. You must get at least 40% on the final exam to pass this course.

All tests will be closed book and notes. Calculators and cell phones are not allowed in any exam/quiz tests. You may get pop quizzes in lecture sessions or in recitation sessions.

Do NOT miss any test/exam. There are no make-up tests. If you miss a test for SERIOUS reasons you have to provide official written evidence. If evidence is accepted a form of grade recovery will be discussed with the instructor. Cheating may result in a failing grade regardless of performance.

Tentative Exam Dates:

<table>
<thead>
<tr>
<th>Exam</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exam 1</td>
<td>September 22</td>
</tr>
<tr>
<td>Exam 2</td>
<td>October 20</td>
</tr>
<tr>
<td>Exam 3</td>
<td>November 15</td>
</tr>
<tr>
<td>Final Exam</td>
<td>December 4, 7 AM</td>
</tr>
</tbody>
</table>

Tentative Grading Scheme:

- Pop-up Quiz tests: 10%
- Mid term exams: 45%
- Final exam: 20%
- Assignments: 25%

The instructor reserves the right to use plus/minus grading in this course.