# COP 3503: Computer Science II

Spring 2014 course syllabus

**Instructor:**  Dr. Narsingh Deo (*deo@eecs.ucf.edu*) (Office: HEC-361)

**Course Meeting Time:**  Tu/Th, 1:30 – 2:45 PM, in ENG2-102

**Course Website:**  [http://webcourses.ucf.edu](http://webcourses.ucf.edu/)/

**Text:**  Introduction to the Design and Analysis of Algorithms, 3/E.  
Anany Levitin, ISBN-10: 0132316811, ISBN-13: 9780132316811  
 ©2012 • Addison-Wesley • Paper, 592 pp

## Introduction

In this course we will hone our problem-solving skills by designing and implementing efficient algorithms. We will also develop formal, mathematical techniques for analyzing algorithms in terms of time and space efficiency. We will also cover recursion, data structures, sorting algorithms, graph algorithms, and a variety of problem-solving techniques--including backtracking, divide and conquer, greedy algorithms, and dynamic programming.

## Weekly Course Activities

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mon** | **Tue** | **Wed** | **Thu** | **Fri** |
| **Lecture** |  | 1:30 – 2:45  (ENG2-102) |  | 1:30 – 2:45  (ENG2-102) |  |
| **Recitation** | 2:30 – 3:20  (BA1-209) |  | 2:30 – 3:20  (BA1-209) |  |  |
| **Recitation** | 3:30 – 4:20  (BA1-209) |  | 3:30 – 4:20  (BA1-209) |  |  |
| **Office** |  |  | Michael Gabilondo[[1]](#footnote-1)  12:00 – 2:00  (HEC-303) |  | Michael Gabilondo  12:00 – 2:00  (HEC-303) |
| **Office** | Michael Veazanchin[[2]](#footnote-2)  12:00 – 2:00  (HEC-213) |  |  |  | Michael Veazanchin  2:00 – 4:00  (HEC-213) |
| **Office** |  | Dr. Narsingh Deo  3:00 – 5:00  (HEC-361) |  | Dr. Narsingh Deo  3:00 – 5:00  (HEC-361) |  |

## Exams, Assignments and Grading

The grade distribution of assignments and exams is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Programming Assignments** | **Exam I** | **Exam 2** | **Final Exam** |
| 30% | 20% | 20% | 30% |

There will be two types of assignments: (1) Self-monitored: reading assignments from the textbook (and other material) plus relevant problems to be solved by the student will be given in class. Answers will be posted later for self-checking. No grading for this part. Your exam performance should reflect your effort here. (2) Programming assignments will be posted on Canvas (Webcourses2@UCF) and announced in class. Late assignments will only be accepted up to 24 hours after each submission deadline and will incur a 25% deduction from any points earned. All assignments must be done in Java.

Exams will be taken in class; and no books, notes or calculators are allowed.

## Cheating Policy

Students caught cheating (on exams or programming assignments) are subject to a variety of disciplinary actions, at the sole discretion of the course instructor. At the very least, students will receive zero credit for any work on which they are caught cheating. Furthermore, the instructor may assign a final course grade of F to any individual caught cheating in this course. Students may also be referred to the Office of Student Conduct for further disciplinary action, pursuant to the policies stated in the Golden Rule Student Handbook.

## Make-up Policy

Make-up exams will be given only in the event of university-excused absences. Excused absences, as defined by the Academic Regulations and Procedures governing the university, include: “illness, serious family emergencies, special curricular requirements (e.g., judging trips, field trips, professional conferences), military obligations, severe weather conditions, and religious holidays.” Furthermore, “participation in official University-sponsored activities, such as music performances, athletic competition, or debate” constitutes an excused absence.

In the case of planned absences, make-up exams must be arranged with the instructor prior to the absence in question. Of course, some excused absences cannot be foreseen (e.g., illness or serious family emergencies). In these cases, you must contact the instructor as soon as possible to schedule a make-up exam. If you do not contact the instructor in a timely manner (typically within 24 hours of missing the exam), you may be denied a make-up exam.

Assignment deadlines may be extended in these situations as well, or, in the case of planned absences, the instructor may require that the assignment be turned in early or on time, depending on the nature of the assignment and the circumstances of the absence; this decision is at the discretion of the course instructor. Your best course of action is to inform me as early as possible if you know you’re going to be out of town or missing classes!

Make-up exams and deadline extensions may be granted in other situations as well, at the discretion of the instructor. TAs are unable to give extensions. All such requests must go through the course instructor.

## Drop and Withdrawal Procedures

If you decide to drop, withdraw from, or stop attending this course for any reason, please send me a quick e-mail informing me of your decision. You do not need to explain your decision for leaving the class, but sending me an e-mail can protect you later, in the unlikely event that your cessation of the course does not get properly recorded with the university.

## Recitations

Recitations (lab sections) will generally serve as advanced problem solving sessions, although the TAs may also be called upon to present new material in some situations. It is in those sessions that you will delve into truly complex problems (beyond the scope of what can be covered in the general lecture), and where you will be able to get expert help on how to apply the theory learned in class in a more personal setting. This is a chance to reinforce your understanding of the material, to get answers to any questions you have, and to hear answers to the questions of other students. Recitation attendance, while not strictly required, is essential to your mastery of the course material, and to your success in this class. You are strongly encouraged to attend.

Because seating is limited, we ask that you attend the recitation for which you registered.

## Course Outline

### Introduction

* What Is an Algorithm?
* Fundamentals of Algorithmic Problem Solving
* Important Problem Types
* Fundamental Data Structures

### Analysis of Algorithms

* The Analysis Framework
* Asymptotic Notations and Basic Efficiency Classes
* Mathematical Analysis of Non-recursive Algorithms
* Mathematical Analysis of Recursive Algorithms
* An Example: Computing the nth Fibonacci Number

### Brute Force and Exhaustive Search

* Selection Sort and Bubble Sort
* Sequential Search and Brute-Force String Matching
* Exhaustive Search
  + Traveling Salesman Problem
  + Knapsack Problem
  + Assignment Problem
* Depth-First Search and Breadth-First Search

### Decrease-and-Conquer

* Insertion Sort
* Topological Sorting
* Decrease-by-a-Constant-Factor
  + Binary Search
* Variable-Size-Decrease

### Divide-and-Conquer

* Mergesort
* Quicksort
* Binary Tree Traversals and Related Properties
* Multiplication of Large Integers and Strassen’s Matrix Multiplication

### Transform-and-Conquer

* Presorting
* Gaussian Elimination
* Balanced Search Trees
  + AVL Trees
  + 2-3 Trees
* Heaps and Heapsort
* Problem Reduction

### Space and Time Trade-Offs

* Sorting by Counting
* Input Enhancement in String Matching
* Hashing
* B-Trees

### Dynamic Programming

* The Knapsack Problem
* Optimal Binary Search Trees
* Warshall - Floyd’s Algorithm

### Greedy Technique

* Prim’s Algorithm
* Kruskal’s Algorithm
* Dijkstra’s Algorithm
* Huffman Trees and Codes

### Algorithmic Limitations

* Lower-Bound Arguments
* Decision Trees
* P, NP, and NP-Complete Problems
* Coping with Limitations
  + Backtracking
  + Branch-and-Bound
  + Approximation Algorithms for NP-Hard Problems

1. mgabilon@knights.ucf.edu [↑](#footnote-ref-1)
2. michael@cs.ucf.edu [↑](#footnote-ref-2)