COURSE DESCRIPTION

Department and Course Number	CS 312	**Course Coordinator**	Valentino Crespi
Course Title	Data Structures and Algorithms	**Total Credits**	4

Current Catalog Description:

Abstract data types and their use in constructing algorithms for manipulating lists, trees, and graphs; analysis of algorithms for searching, sorting, and data structure manipulation.

Textbook:

Richard Johnsonbaugh, Marcus Schaefer. *Algorithms.* Prentice Hall, 2004.

References:

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms (2nd edition).* MIT Press and McGraw Hill, 2001.
- Michael T. Goodrich, Roberto Tamassia. *Data Structures and Algorithms in Java (4th edition).* John Wiley & Sons, Inc, 2006.
- Jon Kleinerg, Eva Tardos. *Algorithmic Design*. Addison-Wesley, 2005.
- Robert L. Kruse. *Data structures and program design.* Prentice Hall, 1987.
- Mark A. Weiss**.** *Data Structures and Algorithm Analysis in Java.* Addison-Wesley, 1998.
- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

Course Goals:

At the end of the course, students are able to

- Analyze the correctness and computational complexity of computer algorithms.
- Design (specify and implement) efficient advanced Data Structures.
- Know advanced design techniques and their nontrivial application to classic problems of searching, sorting, graph optimization, and combinatorial optimization.

These course goals contribute to the success of **Student Learning Outcomes 1.a, 1.d, 1.e, 5, and 6**.

Prerequisites by Topic:

- Java Programming with Data Structures.
- Discrete Mathematics and Basic Calculus.

Major Topics Covered in the Course:

- Mathematical Foundations (Summation Formulas, Logarithms, Induction, Lower and Upper bounds, Asymptotic Notation, Recurrence Relations, Master Theorem, Loop Invariants).
- Analysis of the Correctness and of the Computational Complexity of Computer Algorithms.
- Advanced Data Structures (Binary Search Trees, Balanced Trees, Heaps, Indirect Heaps, Priority Queues, Dictionaries, Hash Tables, Union-Find).
- Graph Algorithms and Searching and Sorting Algorithms.
- Design Techniques (Divide and Conquer, Greedy and Dynamic Programming).

Laboratory Projects (specify number of weeks on each):

At the discretion of the instructor. Projects range from weekly assignments to many class projects over the course of the term.

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	2.5		Data Structures	1.5	
Software Design			Prog. Languages		
Comp. Arch.					

Oral and Written Communications:

Written analysis of algorithms and data structures plus homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

Mathematical tools for the analysis of the correctness and of the computational complexity of Computer Algorithms and Data Structures.

Problem Analysis:

In the first part of the course, students become familiar with basic concepts for analyzing the correctness and computational complexity of algorithms. Those include Loop Invariants, the Asymptotic Notation (big-O) and Recurrence Relations. Next, they are introduced to the design and analysis of advanced data structures, which includes their specification through Abstract Data Types and their efficient implementation through Binary Search Trees, Hash Tables, Heaps and Union-Find structures. Finally they learn advanced algorithm design techniques such as Greedy, Dynamic, and Divide-and-Conquer Programming, and they apply them to solving

classic problems of graph optimization, combinatorial optimization, searching, and sorting.

Solution Design:

Solution design in this course mostly involves a) the experimental analysis of implemented algorithms to be compared with analytical results; and b) the design (specification and implementation) of efficient data structures appropriate for a variety of programming tasks.