

## COURSE DESCRIPTION

**Department and Course Number:** COMP 2210

**Course Title:** Fundamentals of Computing II

**Total Credits:** 4

**Required:** Yes (CSCI, SWEN, WIRS)

**Prerequisites:** COMP 1210

**Class meetings per week:** 3 hours

**Lab meetings per week:** 3 hours

**Course Coordinator:** Dr. Dean Hendrix

**Date Prepared:** February 10, 2004

### **Current Catalog Description:**

Continuation of COMP 1210 with emphasis on data structures such as lists, trees, graphs, and hash tables.

### **Textbooks:**

Carrano, Frank, and Walter Savitch. 2003. *Data Structures and Abstractions with Java*. Pearson Education, Prentice Hall. ISBN 0-13-017489-0.

### **References:**

None.

### **Course Objectives:**

1. Be able to design and implement data structures based on specifications of abstract data types.
2. Be able to apply concepts and techniques from object-oriented programming.
3. Be able to perform fundamental testing and debugging techniques.
4. Be able to perform fundamental maintenance activities.
5. Be able to perform fundamental time and space analysis on algorithms.

### **Prerequisites by Topic:**

1. Fundamentals of object-oriented programming
2. Familiarity with Java

### **Topics Covered:** (specify number of hours on each)

1. Review of Java and object-oriented programming (3 hours)
2. Lists (4.5 hours)
3. Algorithm analysis (3 hours)
4. Recursion, searching, and sorting (3 hours)
5. Iterators (1.5 hours)
6. Stacks and queues (3 hours)
7. Hash tables (3 hours)
8. Binary trees (1.5 hours)
9. Binary search trees (3 hours)
10. Balanced binary search trees (7.5 hours)

11. Heaps (3 hours)
12. Graphs (6 hours)
13. Exams (3 hours)

**Laboratory Projects:** (specify number of weeks on each)

1. Interfaces and generic programming in Java (1.5 weeks)
2. Linked lists (2 weeks)
3. Queues, software testing and debugging (1 week)
4. Hash tables, software maintenance (2 weeks)
5. Binary search trees, testing (1.5 weeks)
6. Balanced search trees (2 weeks)

**Oral and Written Communications:**

All students are required to develop and apply program documentation skills as part of the course programming assignments.

**Social and Ethical Issues:**

None.

**Theoretical Content:**

Fundamentals of algorithm analysis and computational complexity are presented in focused lessons (3 hours) and then applied as appropriate throughout the course.

**Problem Analysis and Solution Design:**

All students apply fundamental software engineering practices to analyze, design, implement, test, and document solutions to all programming assignments. Students are given more responsibilities with respect to both analysis and design, though still at a fundamental level. Students are taught to apply a controlled, iterative approach to developing software components that meet desired needs.

