COURSE DESCRIPTION

Department and Course Number: COMP 2710
Course Title: Software Construction
Total Credits: 3
Required: Yes (CSCI, SWEN, WIRS)
Prerequisites: COMP 2210
Class meetings per week: 3 hours
Lab meetings per week: 0 hours
Course Coordinator: Dr. David Umphress
Date Prepared: February 18, 2004

Current Catalog Description:
Intensive experience in software construction, to include topics such as testing, debugging, and associated tools; configuration management; low-level file and device I/O; systems and event-driven programming.

Textbooks:

References:
None.

Course Objectives:
1. Provide insight into a disciplined approach to building code.
2. Provide exposure to building non-trivial applications.
3. Provide experience in C++.

Prerequisites by Topic:
1. Fundamentals of program design
2. Fundamental data structures

Topics Covered: (specify number of hours on each)
1. Software processes (2 hours)
2. OO design (4 hours)
3. Testing (5 hours)
4. Syntax fundamentals (1 hour)
5. Classes and objects (4 hours)
6. Arrays, pointers, references (2 hours)
7. Memory management (1 hour)
8. Functions (3 hours)
9. Inheritance (3 hours)
10. I/O (5 hours)
11. Virtual functions and polymorphism (3 hours)
12. Templates (3 hours)
13. Exceptions (3 hours)
14. Large scale development (4 hours)
15. Exams (2 hours)

**Laboratory Projects:** (specify number of weeks on each)
1. OO design (1 week)
2. Testing (1 week)
3. Getting acquainted with C++ SDK and IDEs (1 week)
4. Classes (1 week)
5. Data structures (1 week)
6. Functions (1 week)
7. Inheritance (1 week)
8. I/O (1 week)
9. Virtual functions and polymorphism (1 week)
10. Templates (1 week)
11. Exceptions (1 week)
12. Term Project (4 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:**
None.

**Problem Analysis and Solution Design:**
All students apply fundamental software engineering practices to analyze, design, implement, test, and document a solution to a significant problem. Students are taught to apply a controlled, iterative approach to developing a software component and/or subsystem to meet desired needs. Software process is formally introduced to provide a framework in which to develop their solutions.