COURSE DESCRIPTION

Department and Course Number	CMPS 250	**Course Coordinator**	Frank Ducrest
Course Title	Honors Introduction To Data Structures and Software Design	**Total Credits**	4
URL	General CMPS 250 Resources Page: http://fidelio.cacs.louisiana.edu/250/	**Semester hours**	3

Current Bulletin Description
Honors Introduction To Data Structures and Software Design. Accelerated coverage of CMPS 150 and CMPS 260. Students passing CMPS 250 with a C or better will also receive credit for CMPS 150. Fa. Prereq: Prior programming experience in a high-level programming language. Coreq: MATH 270 or equivalent.

Textbook
C++ Programming: Program Design Including Data Structures, 3nd Edition**;** by D. S. Malik, Publisher – Thomsom Course Technology; ISBN-10: 1-4188-3640-0, ISBN-13: 978-1-4188-3640-5

References None

Course Goals
- To gain experience programming in C++
- To gain experience in using Linux and Unix operating systems
- To gain experience with object oriented development and related design issues
- To develop a working knowledge of multi-dimension arrays, object oriented programming; abstract data types lists, stacks, and queues, searching, C++ templates, pointers and dynamic structures

Course Outcomes
Each student passing CMPS 250 should:
- Be capable of working in Linux and Unix operating systems to develop software
- Be sufficiently knowledgeable of and capable of working independently in the C++ programming language such that they are able to study advanced data structures and algorithms in future courses
- Be familiar with basic object oriented concepts and related design issues

Prerequisites by Topic

Both of the following:
- some experience programming in a programming language
- good grounding in algebraic concepts

Or, one of the following:
- greater experience programming in a programming language with some object orientation
- mathematics background through some calculus

Major Topics Covered in the Course

1. **Material covered in CMPS 150 (23 1/3 hours total)**

   Students in this course typically have some programming experience. Degree of time given to individual topics is dependent on a series of survey-tests given before each topic group.

   - Introduction to Unix / Linux
   - Introduction to Computing and Problem Solving
   - Introduction to C++
   - Simple Data Types
   - Input/Output Operations
   - Selection/Decision Structures
   - Iteration Structures
   - File Input/Output
   - Functions
   - Structs (Records)
   - Single and Multi-Dimensional Arrays
   - Exam (50 minutes)

2. **Material covered in CMPS 260 (37 ½ hours total)**

   - Software design documentation (2 hours)
   - Object-based design and decomposition (3 hours)
   - Fundamental data structures, their operations and applications (3 hours)
   - Array implementations: lists, stacks, and queues (4 hours)
   - Pointers (3 hours)
   - Linked-list implementation: lists, stacks, and queues (8 hours)
   - Continued study of C++, separate compilation units, makefiles (2 hours)
   - Classes/objects, templated classes, inheritance (5 hours)
   - Algorithm analysis (3 hours)
   - Searching techniques: sequential and binary (1 hour)
   - Exam (50 minutes)
   - Final Exam (2 hours 30 minutes)

Laboratory projects (specify number of weeks on each)

1. CMPS 150 Material Practice
   - Unix / Linux (1 week)
   - Basic Data Types, I/O, Looping and Selection (1 week)
   - File I/O (1 week)
   - Functions (1 week)
2. Functional Decomposition, Structs and Arrays (2 weeks)
3. Create and use a Class (2 weeks)
4. Create or use Inherited , Overloaded , Template List Classes (2 weeks)
5. Use Template Linked List, Stack, Queue classes (2 weeks)
6. Requirements and Design
   - Requirements Document (1 week)
   - Design Document (1 week)

Oral and Written Communications

Every student is required to submit at least 2 written reports (not including exams, tests, quizzes, or commented programs) of typically 5 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues

Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

N.A.

Theoretical Content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

Object Orientation:	(2 hours)
Correctness: assertions, pre- and post-conditions	(1 hour)
Comparison of Algorithms: Big-O notation, common orders of magnitude	(3 hours)
Program execution environment: run-time stack, activation records	(1 hour)

Problem Analysis

Please describe the analysis experiences common to all course sections.

Students are asked to develop a *Requirements Document* based on a group interview with a prospective user of a proposed project.

Solution Design

Please describe the design experiences common to all course sections.

Students develop a Design Document, which is the formal presentation of the design solution to the problem for which students wrote the Requirements Document. The format of the Design Documentation is specified in the Documentation Standards. Students Receive the Design Documentation of a sample problem.