COURSE DESCRIPTION

Department and Course Number CMPS 150 **Course Coordinator** Nona Etheredge

Course Title Introduction to Computer Science **Total Credits** 3

URL http://fidelio.cacs.louisiana.edu/150/ **Semester hours** 3

Current Bulletin Description
Problem solving, structured design of algorithms, implementation of algorithms, and testing and debugging of programs. Data types, control structures, and abstractions. The laboratory component focuses on algorithm design and implementation. Fa, Sp. Prereq: MATH 109 or 201 with a grade of C or better. Coreq: MATH 110.

Textbook C++ Programming: Program Design Including Data Structures, 3rd Ed., D.S. Malik; Thomson; 2007; ISBN(10): 1-4188-3640-0; ISBN(13): 978-1-4188-3640-5

References http://www.course.com/downloads/computerscience/malikdatastructures3e/index.cfm

Course Goals

1. To acquire a foundation in the UNIX platform and its C++ programming facilities.
2. To understand the fundamentals of the program development process (program definition, requirements specification, design, implementation, testing and maintenance).
3. To develop working knowledge of structured programming techniques and their implementation in C++.
4. To gain experience with top-down design, as well as an introduction to beginning data structures.

Course Outcomes

1. The ability to analyze, design, implement, and test computer programs or components with respect to a variety of criteria relevant to the task.
2. To appreciate the responsibilities and ethical issues in the design and application of computer systems.
3. To understand the concepts and analytical approaches used in the sciences and mathematics.
4. To possess the ability to solve problems using simple algorithms and beginning data structures.

Prerequisites by Topic Pre-Calculus Algebra or Decision Mathematics

Major Topics Covered in the Course

1. Introduction to Computing and Problem Solving (2 hours)
2. Introduction to C++ (2 hours)
3. Remote Operations (1 hour)
4. Simple Data Types (2 hours)
5 Input/Output Operations (4 hours)
6 Selection/Decision Structures (5 hours)
7 File Input/Output (3 hours)
8 Repetition/Iteration Structures (6 hours)
9 Functions (6 hours)
10. Arrays (5 hours)
11. Structs (Records) (3 hours)

Laboratory projects (specify number of weeks on each)

1. UNIX / Compiling a C++ Program (1 week)
2. Parts of a C++ Program (1 week)
3. Arithmetic Expressions (1 week)
4. Selection/Decision Statements (2 weeks)
5. File Input/Output (1 week)
6. Repetition/Iteration Statements (2 weeks)
7. Functions (2 weeks)
8. Structs and Arrays (3 weeks)

Oral and Written Communications

Other than exams and programming assignments, there are no other requirements for oral and/or written communication skills.

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)?

1. Programmer Responsibility / Liability (lecture)
2. Collaboration / Academic Honesty (lecture and programming assignments)

Theoretical Content

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

N/A

Problem Analysis

Students in this course are required to perform the analysis necessary to design, create, debug, and test programs to implement the concepts covered. Although the analysis process at this level is quite simple, classroom lectures, laboratory work, and programming assignments begin to teach the student the process of examining programming specifications, as well as examining the implementation code.

Solution Design

Again, the design process at this level is quite simple, but the simple input / process / output steps are used and refined throughout the semester.