NORMANDALE COMMUNITY COLLEGE **9/28/2011**
COMMON COURSE OUTLINE
CSCI 2001, STRUCTURE OF COMPUTER PROGRAMMING I

I. EFFECTIVE DATE OF OUTLINE

Spring Semester, 2006. To be reviewed by the department annually.

II. CATALOG DESCRIPTION

A. CSCI 2001
B. Structure of Computer Programming I
C. 4 Credits
D. Offered Fall Semester
E. Prerequisites: CSCI 1101 and MATH 1510
F. Introduces students to the fundamental principles of programming and to different programming paradigms, such as message-passing. Students will learn to use the computer language Scheme as a formal way of expressing algorithms and data. Procedures, recursion and iteration will be presented as algorithmic development techniques. Use of abstraction to hide program details and abstract data types will be emphasized throughout the course.

III. RECOMMENDED ENTRY SKILLS/KNOWLEDGE

Before taking CSCI 2001, students should be able to:

A. Use a top-down approach to problem-solving, designing functions to modularize problem solutions.
B. Express problem solutions as algorithms, using some sort of algorithmic representation, e.g., pseudocode.
C. Complete traces of algorithms showing their dynamics.
D. Use structured programming in some programming language.
E. Use a variety of control structures and data structures, e.g., iteration and arrays.
F. Demonstrate an understanding of a first semester of calculus.

IV. OUTLINE OF MAJOR CONTENT AREAS

A. The Scheme dialect of the LISP programming language
B. The elements of programming
C. Procedures, recursion and iteration, higher-order procedures, order analysis
D. Data abstractions, hierarchical data and structures
E. Symbolic data, data-directed programming
F. Assignment and local state, environment model of evaluation

V. LEARNING OUTCOMES

Upon successful completion of CSCI 2001, students will be able to understand and use:

A. The Scheme dialect of the LISP programming language
B. The elements of programming
C. Procedures, recursion and iteration, order analysis
D. Higher-order procedures
E. Sequential and binary searches
F. Data abstraction
G. Hierarchical data and structures
H. Symbolic data
I. Data-directed programming
J. Assignment and local state
K. The environment model of evaluation
L. Modeling with mutable data, queues and tables

VI. METHODS USED FOR EVALUATION OF STUDENT LEARNING

The instructor will choose from among various evaluation techniques including – but not limited to – in-class testing, take-home testing, assignments, quizzes, attendance, group or individual projects, and research. The instructor will also choose a method for end-of-the-semester evaluation.