Course Abbreviation & Number: CSCI 1301

Course Title: PRINCIPLES OF COMPUTER SCIENCE I

Credit Hours: 4

Prerequisites:

- CSCI 1300 and MATH 1111 or MATH 1113 or MATH 2431, each with a "C" or better

Co-requisites:

- None

Course Description:

This course is an introduction to the fundamental principles of computer science. It emphasizes structured, top-down development and testing of computer programs. Major concepts include problem analysis, algorithm development, modular programming, software engineering, and good programming style using a fully capable modern programming language.

Expected Educational Results:

As a result of completing this course, the student will be able to:

1. Analyze a problem and clearly define the available data, desired results, and appropriate process for obtaining that result.
2. Create a structured, top-down design, in algorithmic form, of a solution for said problem.
3. Construct a modular, well-structured program in a specified programming language from a top-down design.
4. Use sequential statements, including input, output, and assignment statements, in a program.
5. Use selection and repetition statements appropriately in a program.
6. Understand and demonstrate proper use of specific basic data types in a program. Create and use routines, procedures and functions, appropriately in a program. Create and use text files for input and output in a program.
7. Implement a simple abstract data type using the appropriate data
constructs and routines.

8. Understand and demonstrate proper use of specific structured data types, including arrays, vectors and structures, in a program.

9. Use simple sorting and searching methods.

10. Demonstrate the correct function of a program by developing a test plan to verify correctness of said program.

11. Trace through and determine the output of a program containing any or all of the above constructs.

12. Using proper English, clearly and completely document a program, including internal comments and external documentation.

**General Educational Outcomes:**

This course addresses the general education outcome relating to communications by providing additional support as follows:

I. Students develop their reading comprehension skills by reading the text and handout materials.

   A. Students develop their listening skills through lecture and small group problem solving. Lecture material is presented that is not included in the text or handout material and is included as part of the tests or assignments.

   B. Students develop their reading and writing skills through the use of problems and activities, including development of computer programs and documentation, developed specifically to enhance their understanding of computer science principles and programming language skills. Students provide written or oral solutions to these problems in either individual or group format. They must also answer short-answer type questions on course exams.

II. This course addresses the general education outcome relating to problem-solving and critical thinking skills through programming assignments that take the student through the programming process from understanding the problem all the way to finalizing a correct program solution to the problem.

III. This course addresses the general education outcomes relating to mathematical concept usage and scientific inquiry as follows:

   A. Students apply mathematical concepts in the development of computer programs by creating mathematically based solutions to the assigned problems and communicating the results of those
solutions to the program user.

B. Students apply the scientific method in the set-up and solution of the problems presented to illustrate computer-programming principles.

IV. This course addresses the general education outcome relating to organization and analysis of information using a computer by using a modern, structured programming language in the solution of problems designed to illustrate the concepts and principles of computer programming.

### Course Content:

#### I. Overview of Computing and Programming (20%)
- A. Binary Data Representation
- B. Computer Organization
- C. Overview of Programming Languages
- D. Problem Solving and Algorithm Development
- E. Software Engineering Concepts

#### II. Basic Programming Technique (50%)
- A. Program Structure
- B. Simple Data Types
- C. Flow Control
  - i. Decisions and Relational Operators
  - ii. Repetition Structures
  - iii. Modular Programming with Functions and Arguments
  - iv. Function Overloading
- D. Working with Text and Data
  - i. Reading and Writing Files
  - ii. Structured Data Types

#### III. Object-Oriented Paradigm (30%)
- A. Abstraction
- B. Class-based Program Design and Inheritance

#### IV. Abstract Data Types (ADT)

### Assessment of Outcome Objectives

#### Course Grade:

The course grade will be determined by the individual instructor using a variety of evaluation methods. The course grade must weight examinations for at least 50% of the grade and programming assignments for not more than 50% of the grade. Five to seven student programming projects must be assigned. Testing must consist of at least two one-hour examinations.
and a comprehensive final examination. The final examination must be weighted at between 25% and 35% of the course grade.

**Course Assessment:**

CSCI 1301 and CSCI 1302 will be assessed together every five years. The assessment instrument will be determined by the CSCI course committee, and will consist of a common project and a set of free response questions that will be included as a portion of the final examination for all students taking the course.

**Use of Assessment Findings:**

The CSCI Committee, or a special assessment committee appointed by the Executive Committee of the Mathematics Academic Group, will analyze the results of the assessment and determine implications for curriculum changes. The committee will prepare a report for the Academic Group summarizing its findings.

**Last Revised:**

December 5, 2011