

CIS 115
INTRODUCTION TO PROGRAMMING AND LOGIC

COURSE DESCRIPTION

Prerequisites: MAT 070

Corequisites: CIS 110

This course introduces computer programming and problem solving in a programming environment, including an introduction to operating systems, text editor, and a language translator. Topics include language syntax, data types, program organization, problem-solving methods, algorithm design, and logic control structures. Upon completion, students should be able to manage files with operating system commands, use top-down algorithm design, and implement algorithmic solutions in a programming language. *This course has been approved to satisfy the Comprehensive Articulation Agreement for the general education core requirement in natural sciences/mathematics.* Course Hours Per Week: Class, 2. Lab, 2. Semester Hours Credit, 3.

COURSE OBJECTIVES:

Upon successful completion of this course, the student will be able to:

- a. Describe the major components in problem solving for a computer program.
- b. Explain the concept of data storage and named memory locations.
- c. Use sequence, selection, and repetition structures in program design.
- d. Use top-down concepts in algorithm design.
- e. Produce simple flowcharts to illustrate problem solutions.
- f. Use pseudocode to illustrate solutions for multiple problems.
- g. Use correct syntax for a programming language.
- h. Use a programming language to illustrate problem solutions.
- i. Use correct operating system's commands to execute programs.
- j. Explain the concepts of batch processing versus interactive processing.

OUTLINE OF INSTRUCTION:

- I. Introduction to problem solving for the computer
 - A. Strategy
 - B. Algorithms
 - C. Control structures
 - D. Testing results
 - E. Problem solving and computer programs

- II. Methods for depicting program design
 - A. IPO charts
 - B. Flowcharts
 - C. Pseudocode

- III. Data storage
 - A. Types of data
 - B. Literals
 - C. Identifiers
 - D. Named storage

- IV. Moving data
 - A. Assignment statements
 - B. Input
 - C. Output
 - D. Constants
 - E. Variables
 - F. Using a language to illustrate data movement

- V. Arithmetic
 - A. Arithmetic operators
 - B. Expressions
 - C. Order of precedence
 - D. Using arithmetic in a program

- VI. Top-down design
 - A. Concept of top-down design
 - B. Input/output design
 - C. Identifying major tasks
 - D. Subtask refinement
 - E. Subroutines

- VII. Selection structures
 - A. Relational operators
 - B. Logical operators
 - C. Selection
 - D. Solving problems using selection
 - E. Programming selection problems

- VIII. Repetition structures
 - A. Concept of looping
 - B. Conditional looping
 - C. Looping with counter
 - D. Solving problems with looping

- IX. File processing
 - A. Identification of files
 - B. Record layout
 - C. File organization
 - D. Control fields
 - E. Solving problems using files

- X. Batch and interactive processing
 - A. General algorithms
 - B. Algorithms for batch processing
 - C. Algorithms for interactive processing
 - D. Solving problems for batch processing
 - E. Solving problems for interactive processing

REQUIRED TEXTBOOKS AND MATERIALS:

Dann, Cooper, and Pausch. Intro to Programming with Alice (with CD-ROM). Prentice Hall.
ISBN 0131872893

and

Zak. Intro to Programming with C++ (with 6 CD folder). 4th ed. Prentice Hall. ISBN 01319872893

STATEMENT FOR STUDENTS WITH DISABILITIES:

Students who require academic accommodations due to any physical, psychological, or learning disability are encouraged to request assistance from a disability services counselor within the first two weeks of class. Likewise, students who potentially require emergency medical attention due to any chronic health condition are encouraged to disclose this information to a disability services counselor within the first two weeks of class. Counselors can be contacted by calling 686-3652 or by visiting the Student Development Office in the Phail Wynn Jr. Student Services Center, room 1309.