

## **CSC 134 C++ PROGRAMMING**

### **COURSE DESCRIPTION:**

Prerequisites: CIS 115  
Corequisites: None

This course introduces object-oriented computer programming using the C++ programming language. Topics include input/output operations, iteration, arithmetic operations, arrays, pointers, filters, and other related topics. Upon completion, students should be able to design, code, test, and debug C++ language programs. *This course has been approved to satisfy the Comprehensive Articulation Agreement for transferability as a premajor and/or elective course requirement.* Course Hours Per Week: Class, 2. Lab, 3. Semester Hours Credit, 3.

### **COURSE OBJECTIVES:**

After completing this course, the student will be able to:

- a. Define the terms: function, char, float, double, short, long, storage classes, and reference variables.
- b. Explain the data types, modifiers of data types and storage classes of C++.
- c. Identify the operators available in C++.
- d. Describe the program control constructs available in C++.
- e. Declare and initialize arrays.
- f. Define the terms: pointer, structure, union, enumerated type, and conditional compilation.
- g. Use the pointer data type and explain the use of indirect addressing.
- h. Explain the concept of using structures, unions, and enumeration to compose new data types.
- i. Illustrate how composed data types can be passed to and returned from functions.
- j. Explain how pointers and structures can be used to form dynamically allocated queues, stacks, and linked list types of data structures.
- k. Explain the concept of class and how it relates to object-oriented programming.
- l. Design, code, test, and debug C++ programs.

### **OUTLINE OF INSTRUCTION:**

- I. Introduction
  - A. History of C and C++
  - B. Higher-level languages
  - C. Operating systems
  - D. Compiling programs
  - E. Structured programming and problem solving
  - F. Symbolic debugger

- II. Basic structure of a C++ program
  - A. Generic form
  - B. Header files
  - C. Define constants
  - D. Main
    - 1. Local variables
    - 2. Basic input/output statements
    - 3. Simple program
  
- III. Variables, data types, and expressions
  - A. Variables
  - B. Composition
  - C. Length
  - D. Data types
    - 1. Basic types
    - 2. Modifiers
    - 3. Constants
  - E. Reference variables
  - F. Operators
    - 1. Arithmetic
    - 2. Increment and decrement
    - 3. Relational
    - 4. Logical
  
- IV. Program control
  - A. For loops
  - B. While loops
  - C. Do-while loops
  - D. Break
  - E. Continue
  - F. If and If-else statements
  - G. Conditional operator
  - H. Switch statement
    - 1. Flags and conditional testing
  
- V. Arrays
  - A. One-dimensional arrays
    - 1. Initializing arrays
    - 2. Character arrays
  - B. Multi-dimensional arrays
    - 1. Initialization
    - 2. Arrays of character arrays
  - C. String manipulation functions

## VI. Pointers

- A. Concept of pointers
- B. Defining and initializing a pointer
- C. Pointers and arrays
- D. Pointer arithmetic
- E. Pointers to dynamically allocate memory
- F. Pointers as function arguments

### **REQUIRED TEXTBOOK AND MATERIALS:**

Deitel. Small C++ How to Program. 5<sup>th</sup> ed. Prentice Hall. ISBN 0131857584.

### **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.