

CSC 153 C# PROGRAMMING

COURSE DESCRIPTION:

Prerequisites: CIS 115

Corequisites: None

This course introduces computer programming using the C# programming language with object-oriented programming principles. Emphasis is on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger. Upon completion, students should be able to design, code, test, debug, and implement objects using the appropriate environment at the beginning level. Course Hours Per Week: 2. Lab, 3. Semester Hours Credit, 3.

COURSE OBJECTIVES:

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

- a. Read, write, execute, and debug C# applications
- b. Understand algorithmic thinking and apply it to programming
- c. Program keyboard input and screen output
- d. Code with arithmetic, increment, decrement, assignment, relational, equality, and logical operators
- e. Code control structures (if, if/else, switch, while, do/while, for) and use primitive data types
- f. Write user-defined methods
- g. Understand and manipulate arrays
- h. Write programs using object-oriented programming techniques including classes, objects, and inheritance
- i. Use graphical user interface (GUI) components
- j. Understand C#'s Event Handling Model

OUTLINE OF INSTRUCTION

- I. Introduction
 - A. History of C#
 - B. Features of C#
 - C. How C# works
 - D. Types of C# Programs
 - E. Edit, compile, and run C# applications
- II. Variables, data types, and expressions
 - A. Identifier rules

- B. Naming variables, constants (final) and references
 - C. Primitive data types
 - D. Arithmetic Operators
 - E. Assignment Operators
 - F. Relational and Logical Operators
- III. Program control flow
- A. Sequence structure
 - B. Selection structure
 - C. Repetition structure
 - D. Jump (Sequence) structure
- IV. Methods
- A. .NET Framework and Foundation Class Library methods
 - B. User-defined methods
 - C. Scope and duration
 - D. Local and Field variables
 - E. Pass-by-value, Pass-by-reference
 - F. Recursion
 - G. Overloading
- V. Arrays
- A. Declaration and allocation
 - B. Passing arrays to methods
 - C. Sorting, searching
 - D. Multiple-subscripted
- VI. Object-Based Programming
- A. Classes and objects, instance variables, and instance methods
 - B. Member access modifiers: public, private, protected, package
 - C. Creating packages
 - D. Constructors, overloaded constructors
 - E. Set (mutator), Get (access), and predicate methods
 - F. Final instance variables
 - G. Composition
 - H. Finalizers, garbage collection
 - I. Static class members
 - J. `this` reference
- VII. Object-Oriented Programming
- A. Inheritance
 - B. Super class, subclass
 - C. Polymorphism
 - D. Dynamic method binding
 - E. Abstract class, Concrete class
 - F. Inner class definition

- G. Type-wrapper class for primitive data types
- H. Interfaces
- VIII. Graphical User Interface
 - A. Event-Driven Programming and Event Handling Model
 - B. Window Components
 - C. Mouse and keyboard event handling

REQUIRED TEXTBOOKS AND MATERIALS:

Deitel. Visual C# 2005: How to Program. 2nd ed. Prentice Hall. ISBN 0131525239.

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.