

CSCI 0027 - VISUAL BASIC .NET PROGRAMMING I

Catalog Description

Prerequisite: Completion of CSCI 10 with grade of "C" or better
Hours: 72 (54 lecture, 18 laboratory)

Description: Introduction to methods and techniques of Visual Basic .NET programming. Includes coverage of user interface design, variables, decisions, menus, functions, object-oriented programming, looping, arrays, and printing. Designed to bring students up to the necessary skill and knowledge level for an intermediate-level programming course. (CSU)

Course Student Learning Outcomes

- CSLO #1: Demonstrate techniques for good program design, including making the interface easy for users as well as following guidelines for designing maintainable programs.
- CSLO #2: Differentiate between various data types, apply proper naming conventions, and declare variables/constants. Perform calculations, convert/format various data types, and use Try/Catch blocks for error handling.
- CSLO #3: Use IF statements to control program flow, evaluate conditions using comparison operators, and combine conditions using logical operators. Perform validation on numeric fields, use a CASE structure for multiple decisions, and debug projects using breakpoints.
- CSLO #4: Create menus for program control, display and use common dialog boxes, and write reusable code in sub procedures and functions.
- CSLO #5: Include multiple forms in an application, understand the various form events and how to handle them, and declare variables with the correct scope and access level for multiform projects.
- CSLO #6: Differentiate between various types of lists, describe standard list operations, and use loops to process list elements.

Effective Term

Fall 2020

Course Type

Credit - Degree-applicable

Contact Hours

72

Outside of Class Hours

90

Total Student Learning Hours

162

Course Objectives

Lecture Objectives:

1. Describe the process of visual program design and development.
2. Explain object-oriented programming.

3. Compare and contrast the concepts of classes, objects, properties, methods, and events.
 4. List the three steps for writing a Visual Basic project.
 5. Identify the elements in the Visual Studio environment.
 6. Define design time, run time, and break time.
 7. Explain how to concatenate strings of text.
 8. Describe how to make a control visible or invisible at run time.
 9. Distinguish between variables, constants, and controls.
 10. Differentiate among the various data types.
 11. Formulate calculations using variables and constants.
 12. Compare and contrast numeric data type conversion using implicit and explicit conversions.
 13. Summarize how to use Try/Catch blocks for error handling.
 14. Explain how to accumulate sums and generate counts.
 15. Describe how to use If statements to control the flow of logic.
 16. Create flowcharts indicating the logic in a selection process.
 17. Evaluate conditions using the relational operators.
 18. Compare and contrast the AND and OR operators.
 19. Explain how to use one event procedure to respond to the events for multiple controls.
 20. Compare and contrast sub procedures and functions.
 21. List the Windows common dialog boxes that relate to files, fonts, colors, and printing.
 22. Define the following object-oriented terminology: encapsulation, inheritance, and polymorphism.
 23. Differentiate between a class and an object.
 24. Illustrate how to declare object variables and assign values to the properties with a constructor.
 25. Compare and contrast shared members and instance members.
 26. Explain the purpose of the constructor and destructor methods.
 27. Differentiate among the available types of combo boxes.
 28. Describe how to add and remove items in a list at run time.
 29. Determine which item in a list is selected.
 30. Explain how to use the Items.Count property to determine the number of items in a list.
 31. Compare and contrast Do Loops and For/Next Loops.
 32. Distinguish between direct access and indirect access of a table.
 33. Analyze the advantages of using ListBox controls with arrays.
- Laboratory Objectives:

1. Design a user interface that has text boxes, group boxes, check boxes, radio buttons, and picture boxes.
2. Manipulate multiple controls simultaneously so that they are aligned vertically and horizontally.
3. Construct a project that has access keys, accept and cancel buttons, proper tab sequence, and Tool Tips.
4. Diagnose program errors using breakpoints, stepping program execution, and displaying intermediate results.
5. Create menus and submenus for program control.
6. Design a program that uses the Windows common dialog boxes.
7. Write reusable code in sub procedures and functions, then call them from other locations.
8. Create a two-tier application that separates the user interface from the business logic.
9. Create a program with list boxes and combo boxes.
10. Establish an array and refer to individual elements in the array with subscripts.
11. Create a structure for multiple fields of related data.
12. Write a table lookup for matching an array element.

General Education Information

- Approved College Associate Degree GE Applicability
- CSU GE Applicability (Recommended-requires CSU approval)
- Cal-GETC Applicability (Recommended - Requires External Approval)
- IGETC Applicability (Recommended-requires CSU/UC approval)

Articulation Information

- CSU Transferable

Methods of Evaluation

- Essay Examinations
 - Example: In an essay, explanation of each of the following OOP features: Encapsulation, Inheritance, Polymorphism. Rubric Grading.
- Objective Examinations
 - Example: Example Objective question: When using inheritance, you can call the base class constructor with this code: a. `BaseClass.Inherit()` b. `Inherit.Base()` c. `MyBase.New()` d. `New.BaseClass()`
- Problem Solving Examinations
 - Example: Write the code for a parameterized constructor that assigns the values to the `FirstName` and `LastName` properties in the class. Rubric Grading.
- Projects
 - Example: The semester project is an opportunity for you to apply what you have learned in class to a real-world situation. You get to choose the topic. The only requirement is that the project must incorporate some of the concepts and principles that you have learned in class this semester. The semester project must be an original program. If your program was not created from scratch, then you must show us what it looked like before you started. In other words, if you are planning to do a "makeover" on an existing program, then you must show the "before" and "after", so that we have a clear picture of your involvement. Rubric Grading.
- Reports
 - Example: Write a research paper that covers the following topic: Compare and contrast procedural programming with object-oriented programming. In particular, explain the advantages of Encapsulation, Inheritance, and Polymorphism. Use examples to illustrate the differences as they relate to a particular task. Cite references to current published works. Rubric Grading.

Repeatable

No

Methods of Instruction

- Laboratory
- Lecture/Discussion
- Distance Learning

Lab:

1. Following an instructor discussion on visual basics, students will conduct a hands-on group computer activity based on the discussion. For example, have each group create a program using Visual Basic. Tasks would be design, coding, and debugging. The topic for the

hands-on activity is chosen to reinforce the lecture and to prepare them for the homework assignment. (Laboratory Objectives 1 & 4)

Lecture:

1. Slide show presentation interspersed with topic-specific video clips and live demonstrations of programming techniques. For example, the lecture topic might be Visual Basic. First, review background and general terminology. Second, do a live demonstration on creating a program with Visual Basic. Third, inspire them to investigate further by showing a topic-specific video clip from a website like this one: <http://www.ted.com/tedtalks/> (Lecture Objective 1)

Distance Learning

1. The instructor will present a video lecture describing the process of visual program design and development, classes, objects, properties, and methods. After the student views the lecture, they will then be given a problem to design and implement a program. (Lecture Objectives 1 & 2)

Typical Out of Class Assignments Reading Assignments

1. Read the assigned pages from the textbook on classes and objects and be prepared to discuss in class the use of classes and objects in the context of visual inheritance.
2. Read the assigned pages from the textbook on arrays and be prepared to discuss in class the use of two-dimensional arrays to create dynamic lookup tables.

Writing, Problem Solving or Performance

1. Complete all of the Case Studies at the end of the chapter and create test cases for 'If' statements to make sure that all program branches are working correctly.
2. Complete all of the Case Studies at the end of the chapter and test their outputs with different seed values for the random number generator.

Other (Term projects, research papers, portfolios, etc.)

Required Materials

- Programming in Visual Basic 2010
 - Author: Julia Case Bradley & Anita Millspaugh
 - Publisher: Career Education
 - Publication Date: 2010
 - Text Edition: 1st
 - Classic Textbook?: No
 - OER Link:
 - OER:
- Programming in Visual Basic 2010: The Very Beginner's Guide
 - Author: Jim McKeown
 - Publisher: Cambridge University Press
 - Publication Date: 2010
 - Text Edition: 1st
 - Classic Textbook?: No
 - OER Link:
 - OER:
- Programming with Microsoft Visual Basic 2010

- Author: Diane Zak
- Publisher: Course Technology
- Publication Date: 2020
- Text Edition: 9th
- Classic Textbook?: No
- OER Link:
- OER:

Other materials and-or supplies required of students that contribute to the cost of the course.