Western Technical College

10152190 Introduction to .NET

Course Outcome Summary

Course Information

Description	Presents Windows-based programming through the use of .NET. Emphasis is on windows form creation, .NET syntax, control structures, common Windows controls, arrays, text files, common dialog boxes and object oriented principles.
Career Cluster	Information Technology
Instructional Level	Associate Degree Courses
Total Credits	3
Total Hours	72

Pre/Corequisites

Prerequisite	10804123 Math with Business Applications (OR) 10804133 Math & Logic
Prerequisite	10152153 Introduction to Java

Textbooks

Murach's C#2015. Copyright 2016. Boehm, Anne. Publisher: Mike Murach & Associates, Inc. **ISBN-13**:978-1-890774-94-3. Required.

Success Abilities

1. Cultivate Passion: Enhance Personal Connections
2. Cultivate Passion: Expand a Growth-Mindset
3. Cultivate Passion: Increase Self-Awareness
4. Live Responsibly: Develop Resilience
5. Live Responsibly: Embrace Sustainability

6. Live Responsibly: Foster Accountability
7. Refine Professionalism: Act Ethically
8. Refine Professionalism: Improve Critical Thinking
9. Refine Professionalism: Participate Collaboratively
10. Refine Professionalism: Practice Effective Communication

Course Competencies

1. Identify the key features of the Visual Basic .NET programming language.

Assessment Strategies

1.1. Written Product

Criteria

You will know you are successful when

1.1. you write short C# programs using Visual Studio.
1.2. you compile and run C# programs using Visual Studio.
1.3. you explain the .NET framework, MSIL, and the CLR.
1.4. you produce code snippets to solve problems.
1.5. you use .NET help features.

Learning Objectives

1.a. Utilize the Visual Studio IDE to write short programs (approx. < 500 lines).
1.b. Describe the .NET framework, MSIL, and the CLR.
1.c. Utilize .NET help features.
1.d. Explore the features of the Visual Studio IDE.

2. Explore computing architecture and programming fundamentals.

Assessment Strategies

2.1. Written Product

Criteria

You will know you are successful when

2.1. you demonstrate a fundamental understanding of core computer components.
2.2. you define the terms bit and byte.
2.3. you convert between decimal and binary number systems.
2.4. you describe the differences between an interpreted and compiled language.
2.5. you produce C# code snippets and C# projects that solve problems.
2.6. you obtain user input and produces output using Windows Forms.
2.7. you describe the components and benefits of a three-layer architecture.

Learning Objectives

2.a. Contrast interpreted and compiled languages.
2.b. Formulate basic algorithms.
2.c. Obtain user input and produce output using Windows Forms applications.
2.d. Describe the benefits of using three-layer applications.

3. Perform basic computation using variables and data types.

Assessment Strategies

3.1. Written Product

Criteria

You will know you are successful when

3.1. you declare and write assignment statements using numeric and string data types.
3.2. you explain the order of execution.
3.3. you use parenthesis in assignment statements.

3.4. you describe the C# operators.
3.5. you declare named constants.
3.6. you manipulate text.
3.7. you convert between data types.

Learning Objectives

3.a. Declare and write assignment statements using number and character data types.
3.b. Define named constants.
3.c. Manipulate strings of characters.
3.d. Convert and cast from one type to another.

4. Apply industry standard best practices.

Assessment Strategies

4.1. Written Product

Criteria

You will know you are successful when

4.1. you use comments, correct case, indentation, and other formatting rules in programs.
4.2. you describe the importance of writing quality code.
4.3. you demonstrate basic usage of a version control system.
4.4. you use classes, methods, and other code structures that facilitate code reuse.

Learning Objectives

4.a. Write C# programs using comments, correct casing, indentation, and other formatting rules.
4.b. Demonstrate coding for software reuse and sustainability.
4.c. Commit and checkout a program to a version control system.

5. Use code control structures.

Assessment Strategies

5.1. Written Product

Criteria

You will know you are successful when

5.1. you control program flow using if-else statements.
5.2. you control program flow using the case statement.
5.3. you use nested and multibranch statements.
5.4. you evaluate boolean operator precedence.
5.5. you iterate using for, while, and do-while loops.
5.6. you use the exit and continue statements in loops.
5.7. you debug code control structures.

Learning Objectives

5.a. Use branching statements.
5.b. Compare built-in value types and object values.
5.c. Iterate using for, while, and do loops.

6. Apply fundamental object-oriented principles.

Assessment Strategies

6.1. Written Product

Criteria

You will know you are successful when

6.1. you describe the terms: class, object, property, method, and constructor.
6.2. you write code to define a method.
6.3. you use dot notation.
6.4. you instantiate objects.
6.5. you create custom classes.
6.6. you use the Me keyword.
6.7. you encapsulate using the public and private keywords and properties.
6.8. you write class constructors.

6.9. you describe encapsulation, polymorphism, and inheritance.
6.10. you demonstrate knowledge of variable scope.
6.11. you demonstrate object comparison.
6.12. you demonstrate understanding of pass-by-value and pass-by-reference.

Learning Objectives

6.a. Understand the concept of a class and instantiation of an object.
6.b. Understand class, procedure, and block scope.
6.c. Construct procedures and understand their mechanics – components of a signature, parameter passing, precondition and postconditions, overloading, named arguments, optional arguments, and etc.
6.d. Encapsulate appropriately.
6.e. Utilize object references and dot notation.
6.f. Describe the components of a class – constructor, method, field, and property.
6.g. Build class diagrams in Visual Studio.

7. Explore debugging and testing a program.

Assessment Strategies

7.1. Written Product

Criteria

You will know you are successful when

7.1. you differentiate and resolve syntax errors, run-time errors, and logic errors.
7.2. you perform basic testing on code.
7.3. you use tools to assist in debugging a program.
7.4. you write a basic testing document.

Learning Objectives

7.a. Differentiate and resolve syntax errors, run-time errors, and logic errors.
7.b. Perform basic testing.
7.c. Utilize an IDE for debugging.

8. Examine fundamental data structures.

Assessment Strategies

8.1. Written Product

Criteria

You will know you are successful when

8.1. you construct single and multidimensional arrays.
8.2. you manipulate array data.
8.3. you apply commonly used collections.
8.4. you explain the difference between typed vs. untyped collections.

Learning Objectives

8.a. Construct single and multidimensional arrays.
8.b. Manipulate array data.
8.c. Work with the commonly used collection classes.

9. Examine exception handling.

Assessment Strategies

9.1. Written Product

Criteria

You will know you are successful when

9.1. you explain the importance of exception handling.
9.2. you describe the kinds of exceptions and the exception hierarchy.
9.3. you demonstrate catching and throwing exceptions.
9.4. you explain the difference between data validation and exception handling.

Learning Objectives

9.a. Understand the importance of exception handling.

9.b. Understand the exception hierarchy.
9.c. Differentiate between data validation and exception handling.
9.d. Catch and throw exceptions.

10. Explore working with Windows Forms Applications.

Assessment Strategies

10.1. Written Product

Criteria

You will know you are successful when

10.1. you build simple Windows Forms applications using Visual Studio.
10.2. you choose the appropriate Windows Form Control -- textbox, label, button, checkbox, combobox, and etc -- and implement it.
10.3. you use the Visual Studio design environment and code to set and retrieve properties for Windows Form Controls.
10.4. you bind and respond to events invoked from a Windows Form Control using the IDE and code.

Learning Objectives

10.a. Utilize common Windows Form controls, like text boxes, labels, buttons, and etc to build basic user interfaces.
10.b. Set and retrieve properties from Windows Form controls via the IDE and code.
10.c. Bind and respond to events invoked from a Windows Form controls.

11. Develop a project proposal for an application that incorporates global goals.

Assessment Strategies

11.1. Project

Criteria

You will know you are successful when

11.1. you summarize your project.
11.2. you define the objectives of your application.
11.3. you describe the target clients for your application.
11.4. you identify the value of the project.
11.5. you highlight the features and description of the application.
11.6. you develop a timeline for project completion.
11.7. you summarize research from 5 reputable sources.

Learning Objectives

11.a. Identify global goal for project.
11.b. Develop a project proposal including timeline.
11.c. Identify sources for research (ex: NPR.org, World Health Organization, BBC, fivethirtyeight.com, New York Times, Washington Post, Wall Street Journal, gatesfoundation.org, information from the Global Goals website).
11.d. Incorporate numeric and string data into application.
11.e. Incorporate control structures into application.
11.f. Incorporate methods and event handlers into application.
11.g. Incorporate exceptions and data validation into application.
11.h. Incorporate arrays and collections into application.
11.i. Incorporate dates and strings into application.
11.j. Incorporate at least two of these: combo box, list box, radio button, check box, and group box.
11.k. Develop at least three Windows Forms.

12. Create a Windows Forms C# application based on the project proposal.

Assessment Strategies

12.1. Project

Criteria

You will know you are successful when

12.1. you incorporate numeric and string data into application.

12.2. you incorporate control structures into application.
12.3. you incorporate methods and event handlers into application.
12.4. you incorporate exceptions and data validation into application.
12.5. you incorporate arrays and collections into application.
12.6. you incorporate dates and strings into application.
12.7. you incorporate at least two of these: combo box, list box, radio button, check box, and group box.
12.8. you develop at least three Windows Forms.
12.9. you hyperlink sources within the application.
12.10. you verify the application is error-free.
12.11. you verify there are no unhandled events.
12.12. you comment your code.
12.13. you use proper coding format including object naming conventions.

Learning Objectives

12.a. Incorporate industry standard best practice - comments, object naming conventions, etc.
12.b. Perform basic computation using variables and data types within the application.
12.c. Use code control structures.
12.d. Apply fundamental object-oriented principles.
12.e. Apply fundamental data structures.
12.f. Debug and test the application.
12.g. Apply exception handling as you create the application.