GAVILAN COLLEGE
5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

COURSE: CSIS 44 DIVISION: 50 ALSO LISTED AS:

TERM EFFECTIVE: Spring 2018 CURRICULUM APPROVAL DATE: 10/09/2017

SHORT TITLE: C# .NET PROGRAMMING

LONG TITLE: C# .NET Programming

Units	Number of Weeks		Contact Hours/Week		Total Contact Hours
4	18	Lecture:	3	Lecture:	54
		Lab:	3	Lab:	54
		Other:	0	Other:	0
		Total:	6	Total:	108

COURSE DESCRIPTION:

This class will teach programming using the C# (C Sharp) language provided in the Microsoft .NET framework. Students will learn about variables and constants, expressions and statements, operators and namespaces. Most importantly, students will learn how to create classes and instantiate objects. This course will provide a solid foundation for exploring the .NET framework as well as advanced topics in C#. This course has the option of a letter grade or pass/no pass. ADVISORY: CSIS 45 C++ Programming

PREREQUISITES:

COREQUISITES:

CREDIT STATUS: D - Credit - Degree Applicable

GRADING MODES
- L - Standard Letter Grade
- P - Pass/No Pass

REPEATABILITY: N - Course may not be repeated

SCHEDULE TYPES:
- 02 - Lecture and/or discussion
- 03 - Lecture/Laboratory
- 04 - Laboratory/Studio/Activity
- 72 - Dist. Ed Internet Delayed
- 73 - Dist. Ed Internet Delayed LAB

STUDENT LEARNING OUTCOMES:

1. Design and Create C# programs and use .NET system.

Measure of assessment: Homework, projects, exercises.

Year assessed, or planned year of assessment: 2018

Semester: Spring

2. Use selection and repetition commands.

Measure of assessment: Homework, exercises.

Year assessed, or planned year of assessment: 2018

Semester: Spring

3. Create and use classes and inheritance.

Measure of assessment: Programs, exercises.

Year assessed, or planned year of assessment: 2018

CONTENT, STUDENT PERFORMANCE OBJECTIVES, OUT-OF-CLASS ASSIGNMENTS

Curriculum Approval Date: 10/09/2017

6 Hours

Lecture Content:

Introduction to C# and .NET technology.

Object-oriented programming and the .NET system.

Compiling and executing C# programs.

Data Types and variables.

Arithmetic operations and formatting output.

Input and named constants.

Student Performance Objectives: Describe how to enter, compile, and debug programs. Discuss how to write new programs.

6 Hours

Lecture Content:

Using methods with several types of arguments and return types.

Writing methods that have no arguments and return no values.

Utilizing methods that have arguments and return values.

Avoiding ambiguous methods and over loading methods.

Student Performance Objectives: Discuss how to write C# methods that do not have arguments and do not return values. Explain how to develop C# programs with methods that have arguments and return values.

6 Hours

Lecture Content:

Creating and using classes.

Understanding class concepts.

Declaring and instantiating objects.

Using public fields and private methods.

Understanding this reference and constructor methods.

Overloading constructors and passing parameters to constructors.

Student Performance Objectives: Explain how to write programs that use public fields and private methods. Discuss how to create programs that use "this" reference and constructors. Discuss programs that overload constructors and pass parameters to constructors.

6 Hours

Lecture Content:

Selection and repetition.

Making decisions using the if-else statements and conditional operators.

Select using the switch statement.

Using while, for and do-while loops.

Student Performance Objectives: Describe how to write programs that use the if-else statements. Explain how to create programs that use switch statements. Discuss how to develop programs that use while, for and do-while loops.

6 Hours

Lecture Content:

Using arrays and array methods.

Declaring and initializing arrays.

Using array indexes and array length.

Using for loops to process arrays.

Searching through and sorting arrays.

Writing array methods and using arrays of objects.

Student Performance Objectives: Describe how to write programs that declare and initialize arrays. Discuss how to design programs that use subscripts and for each to process arrays. Explain how to develop programs that search and sort arrays. Describe how to create programs that use methods with arrays and arrays of objects.

6 Hours

Lecture Content:

Introduction to inheritance and understanding the concept of inheritance.

Extending classes and using the protected access specifier.

Using object classes and methods.

Working with superclasses and constructors.

Student Performance Objectives: Describe how to write programs that use inheritance. Explain how to design programs that extend classes and use the protected access specifier. Describe how to develop programs that use superclasses and constructors.

6 Hours

Lecture Content:

Continuation of inheritance.

Creating and using abstract classes and class interfaces.

Student Performance Objectives: Describe how to write programs that use abstract classes and class interfaces.

6 Hours

Lecture Content:

Understanding and using exceptions.

Working with traditional error-handling methods.

Using system-provided exceptions and purposely generating a system exception.

Writing, throwing and catching programmer created exceptions.

Using "finally" blocks and tracing exceptions.

Student Performance Objectives: Describe how to write programs that use language provided exceptions. Discuss how to develop programs that generate a normal exception. Explain how to design programs that write, throw and catch programmer created exceptions. Describe how to create programs that use finally blocks and trace exceptions.

3 Hours

Lecture Content:

Using GUI objects and Visual Studio IDE.

Creating forms and form objects.

Using controls such as labels, checkboxes, radio buttons, and lists.

Student Performance Objectives: Describe how to write programs that use GUI objects and Visual Studio IDE. Explain how to design programs that create forms and form objects. Describe how to develop programs that process labels, checkboxes, radio buttons and lists.

2 Hours

Final exam and final projects.

Lab Content:

6 Hours

Lab Content:

Enter, compile, and debug the chapter sample programs.

Write new programs from the end-of-chapter exercises.

Student Performance Objectives: Demonstrate how to enter, compile, and debug programs. Demonstrate how to write new programs.

6 Hours

Lab Content:

Write C# methods that do not have arguments and do not return values.

Develop C# programs with methods that have arguments and return values.

Student Performance Objectives: Demonstrate how to write C# methods that do not have arguments and do not return values.

Demonstrate how to develop C# programs with methods that have arguments and return values.

6 Hours

Lab Content:

Write programs that use public fields and private methods.

Create programs that use "this" reference and constructors.

Develop programs that overload constructors and pass parameters to constructors.

Student Performance Objectives: Demonstrate how to write programs that use public fields and private methods. Create programs that use "this" reference and constructors. Develop programs that overload constructors and pass parameters to constructors.

6 Hours

Lab Content:

Write programs that use the if-else statements.

Create programs that use switch statements.

Develop programs that use while, for and do-while loops.

Student Performance Objectives: Demonstrate how to write programs that use the if-else statements. Create programs that use switch statements. Develop programs that use while, for and do-while loops.

6 Hours

Lab Content:

Write programs that declare and initialize arrays.

Design programs that use subscripts and for each to process arrays.

Develop programs that search and sort arrays.

Create programs that use methods with arrays and arrays of objects.

Student Performance Objectives: Demonstrate how to write programs that declare and initialize arrays. Design programs that use subscripts and for each to process arrays. Develop programs that search and sort arrays. Demonstrate how to create programs that use methods with arrays and arrays of objects.

6 Hours

Lab Content:

Write programs that use inheritance.

Design programs that extend classes and use the protected access specifier.

Develop programs that use superclasses and constructors.

Student Performance Objectives: Demonstrate how to write programs that use inheritance. Design programs that extend classes and use the protected access specifier. Develop programs that use superclasses and constructors.

6 Hours

Lab Content:

Write programs that use abstract classes and class interfaces.

Student Performance Objectives: Demonstrate how to write programs that use abstract classes and class interfaces.

6 Hours

Lab Content:

Write programs that use language provided exceptions.

Develop programs that generate a normal exception.

Design programs that write, throw and catch programmer created exceptions.

Create programs that use finally blocks and trace exceptions.

Student Performance Objectives: Demonstrate how to write programs that use language provided exceptions. Develop programs that generate a normal exception. Design programs that write, throw and catch programmer created exceptions. Demonstrate how to create programs that use finally blocks and trace exceptions.

3 Hours

Lab Content:

Write programs that use GUI objects and Visual Studio IDE.

Design programs that create forms and form objects.

Develop programs that process labels, checkboxes, radio buttons and lists.

Student Performance Objectives: Demonstrate how to write programs that use GUI objects and Visual Studio IDE. Design programs that create forms and form objects. Develop programs that process labels, checkboxes, radio buttons and lists.

2 Hours

METHODS OF INSTRUCTION:

Lecture, computer demonstrations, C# program examples, and web searches.

OUT OF CLASS ASSIGNMENTS:

Required Outside Hours: 108

Assignment Description: Reading the text, working on sample programs and homework programs.

METHODS OF EVALUATION:

Writing assignments

Percent of total grade: 20.00 %

Writing assignments: 10% - 30% Written homework

Problem-solving assignments

Percent of total grade: 20.00 %

Problem-solving demonstrations: 20% - 40% Homework problems, Quizzes, Exams

Skill demonstrations

Percent of total grade: 50.00 %

Skill demonstrations: 40% - 80% Demonstration, Performance exams

Objective examinations

Percent of total grade: 10.00 %

Objective examinations: 10% - 20% Multiple Choice, True/False, Matching Items, Completion

REPRESENTATIVE TEXTBOOKS:

Required Representative Textbooks

Tony Gaddis. Starting out with Visual C# (4th Edition). Pearson,2016.

ISBN: 0134382609

Reading Level of Text, Grade: 12th Verified by: MS Word

ARTICULATION and CERTIFICATE INFORMATION

Associate Degree:
CSU GE:
IGETC:
CSU TRANSFER:
Transferable CSU, effective 200630
UC TRANSFER:
Not Transferable

SUPPLEMENTAL DATA:
Basic Skills: N
Classification: Y
Noncredit Category: Y
Cooperative Education:
Program Status: 1 Program Applicable
Special Class Status: N
CAN:
CAN Sequence:
CSU Crosswalk Course Department: CSIS
CSU Crosswalk Course Number: 44
Prior to College Level: Y
Non Credit Enhanced Funding: N
Funding Agency Code: Y
In-Service: N
Occupational Course: B
Maximum Hours:
Minimum Hours:
Course Control Number: CCC000208967
Sports/Physical Education Course: N
Taxonomy of Program: 070710