CS 19.21A Course Outline as of Spring 2010

CATALOG INFORMATION

Dept and Nbr: CS 19.21A Title: INTRO TO PROG WITH C#
Full Title: Introduction to Programming with C#
Last Reviewed: 10/13/2008

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	4.00	Lecture Scheduled	4.00	17.5	Lecture Scheduled	70.00
Minimum	4.00	Lab Scheduled	0	17.5	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	4.00		Contact Total	70.00
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 140.00 Total Student Learning Hours: 210.00

Title 5 Category:	AA Degree Applicable
Grading:	Grade Only
Repeatability:	00 - Two Repeats if Grade was D, F, NC, or NP
Also Listed As:	
Formerly:	CIS 19A

Catalog Description:
This course is designed to introduce students to the concepts of computer programming using the C# programming language. The student will write programs to implement a variety of typical applications. There will be an emphasis on structured programming techniques, writing readable code and developing user-friendly programs. Concepts introduced will include: data types, constants and variables, flow of control, decisions and loops, forms and simple C# controls as elements of the user interface, arrays, scope of variables, functions, string manipulation, rudimentary file operations, and error handling.

Prerequisites/Corequisites:

Recommended Preparation:
Eligibility for ENG 100 or ESL 100 and Completion of CIS 101A or CIS 5

Limits on Enrollment:

Schedule of Classes Information:
Description: Introduces programming concepts using C# with emphasis on structured programming, readable code, and user-friendly programs. (Grade Only)

Prerequisites/Corequisites:
Recommended: Eligibility for ENG 100 or ESL 100 and Completion of CIS 101A or CIS 5
Limits on Enrollment:
Transfer Credit: CSU;UC.
Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

AS Degree:	**Area**			Effective:	Inactive:
CSU GE:	**Transfer Area**			Effective:	Inactive:
IGETC:	**Transfer Area**			Effective:	Inactive:
CSU Transfer:	Transferable	Effective:	Fall 2009	Inactive:	Fall 2015
UC Transfer:	Transferable	Effective:	Fall 2009	Inactive:	Fall 2015

CID:

Certificate/Major Applicable:
Certificate Applicable Course

COURSE CONTENT

Outcomes and Objectives:
Upon completion of the course, students will be able to:
1. Use appropriate logic design tools to develop program logic prior to writing programs.
2. Demonstrate competency in using micro computers and proper editing techniques when writing computer programs.
3. Demonstrate ability to design screen forms and program output.
4. Write, test and debug simple to reasonably complex computer programs in C#, using structured programming techniques to solve a variety of typical problems.
5. Produce complete documentation for any given program.

Topics and Scope:

1. Introduction
a. Review of basic computer skills and the Windows graphical user interface
b. The C# programming environment
c. Elements of the user interface; forms and simple C# controls and their properties and methods (controls including command buttons, labels, and text boxes)
d. Editing techniques, program format, and documentation

2. Language rules and structure
a. Obtaining user input (working with additional controls including message and input boxes, check boxes, option buttons, common dialog controls, and menus)
b. Data types: use of constants and variables; calculations and built-in functions
c. Simple data validation and error handling
d. String manipulation and formatting data for output
e. Syntax and logic errors; introduction to using debugging tools

3. Flow of control and programming logic design
a. Decision structures and logical comparisons
b. Loop structures

4. Program decompositions and structured programming techniques
a. Modules, subprograms, and subfunctions
b. Working with multiple forms and standard code modules
c. Scope of variables and constants
d. Arguments and parameter passing

5. Lists and Arrays
a. Introduction to the concepts of lists and simple arrays and their uses
b. Working with list and combo boxes

6. Introduction to disk file processing
a. Elementary storage concepts
b. Rudimentary file operations using sequential access disk files

7. Introduction to graphics with simple images

Assignment:

1. Read 30-50 pages from the textbook each week.
2. Write computer programs using the C# programming language.
3. Test, detect and fix errors in computer programs.
4. Formulate accurate and descriptive program documentation.
5. Complete a team programming project.
6. Take objective examinations.

Methods of Evaluation/Basis of Grade:

Writing: Assessment tools that demonstrate writing skills and/or require students to select, organize and explain ideas in writing.

Written program documentation	Writing 10 - 20%

Problem Solving: Assessment tools, other than exams, that demonstrate competence in computational or non-computational problem solving skills.

Homework problems, programming assignments	Problem solving 30 - 60%

Skill Demonstrations: All skill-based and physical demonstrations used for assessment purposes including skill performance exams.

None	Skill Demonstrations 0 - 0%

Exams: All forms of formal testing, other than skill performance exams.

Multiple choice, true/false, matching items, completion, design and code programming exercises.	Exams 20 - 30%

Other: Includes any assessment tools that do not logically fit into the above categories.

Team programming project	Other Category 10 - 20%

Representative Textbooks and Materials:
Simply C#: An Application-Driven Tutorial Approach, (1st Edition) by Deital, Deitel, Hoey & Yaeger - Prentice Hall, 2004