CS 19.21B Course Outline as of Spring 2010

CATALOG INFORMATION

Dept and Nbr: CS 19.21B   Title: ADVANCED C# PROGRAMMING
Full Title: Advanced C# Programming
Last Reviewed: 1/26/2015

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	3.00	Lecture Scheduled	3.00	17.5	Lecture Scheduled	52.50
Minimum	3.00	Lab Scheduled	0	17.5	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	3.00		Contact Total	52.50
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 105.00   Total Student Learning Hours: 157.50

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 19B

Catalog Description:
This is the second semester course in C# programming for the CIS 19A graduate or the person who has prior programming experience but has not worked with C#. Students will prepare reasonably complex programs and work with object oriented programming and features of the .Net framework class libraries. This course is taught using the current version of Visual C# from Microsoft

Prerequisites/Corequisites:
Completion of CIS 19A or CIS 16A or CIS 10 or CS 10

Recommended Preparation:
Eligibility for ENGL 100 or ESL 100

Limits on Enrollment:

Schedule of Classes Information:
Description: This course is the second semester in C# for CIS 19A graduates or those who have prior programming experience but have not worked with C#. Students will prepare reasonably complex programs and work with object oriented programming and features of the .Net framework class libraries. (Grade Only)

Prerequisites/Corequisites: Completion of CIS 19A or CIS 16A or CIS 10 or CS 10
Recommended: Eligibility for ENGL 100 or ESL 100
Limits on Enrollment:
Transfer Credit: CSU;UC.
Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

				Effective:		Inactive:
AS Degree:	**Area**			Effective:		Inactive:
CSU GE:	**Transfer Area**			Effective:		Inactive:
IGETC:	**Transfer Area**			Effective:		Inactive:
CSU Transfer:	Transferable	Effective:	Fall 2009	Inactive:		Fall 2018
UC Transfer:	Transferable	Effective:	Fall 2009	Inactive:		Fall 2018

CID:

Certificate/Major Applicable:
Certificate Applicable Course

COURSE CONTENT

Outcomes and Objectives:
Upon completion of the course, students will be able to:
1. Summarize the beginning concepts and instructions of the C# programming language.
2. Demonstrate understanding of event-driven programming and graphical user interface design.
3. Design, write, test, debug and document reasonably complex computer programs in C# using object oriented programming techniques to solve a variety of advanced problems.
4. Create, access and maintain accurate data files (text and relational database) through a C# program interface.
5. Develop programs incorporating computer graphics elements.

Topics and Scope:

1. Review and Overview
a. Review of the C# programming environment, language rules and structure.
b. Review of forms and simple C# controls and their properties and methods
c. Concepts of event-driven programming
d. User interface design

2. Advanced Use of the C# Integrated Development Environment
a. Advanced environment options
b. Advanced debugging tools and techniques

3. Object Oriented Design Techniques
a. What is an object?
b. Modeling real world interaction
c. Unified modeling language

4. Object Oriented Programming
a. Classes and structures
b. Properties and methods
c. Interface programming
d. Inheritance, polymorphism
e. Information hiding and encapsulation

5. Exploring the .Net framework
a. Built-in controls
b. Built-in data types
c. The common language runtime
d. Advanced string manipulation

6. Error handling
a. Exception handling
b. User defined exceptions

7. Relational Database Programming
a. Microsoft SQL server
b. Introduction to database file concepts and operations
c. Data controls, data-bound controls and grid controls

8. Web programming
a. Creating a C# website
b. Microsoft internet information server
c. Web services

Assignment:

1. Read 30-50 pages from the textbook each week.
2. Write reasonably complex computer programs using the C# programming language using proper structure and style.
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:

Visual C# How to Program, (3rd Edition) by Harvey & Paul Deitel & Associates September 26, 2008