ACADEMIC AFFAIRS

Course Number: PROG 316   Department: Computer Information Technologies

Course Title: Advanced C#   Semester: Spring   Year: 2011

Objectives/Competencies

Course Objective	Competencies
1. To understand advanced C# programming concepts.	1. Use Visual Studio C# .NET to develop advanced-level programs
2. To explore and understand one-dimensional and multi-dimensional arrays.	1. Properly declare an array 2. Use subscripts to access array elements 3. Initialize an array 4. Sort an array 5. Retrieve values from an array
3. To utilize buttons, checkboxes, listboxes, combo boxes and radio buttons.	1. Create well-designed programs that demonstrate the proper use of combo boxes and radio buttons
4. To create user-defined classes.	1. Add a class to a project 2. Initializing class objects using constructors 3. Define class properties 4. Utilize the get accessor 5. Utilize the set accessor

Course Objective	Competencies
	6. Control access to members by using public and private member-access modifiers
5. To enhance knowledge of the debugger.	1. Utilize the capabilities of the Auto window 2. Utilize the capabilities of the Locals window 3. Utilize the capabilities of breakpoints 4. Use constructors that use dynamic memory allocation 5. Use a copy constructor 6. Use constant functions, pointers and arguments to functions
6. To understand the requirements of a program that requires user to enter large quantities of text.	1. Use the TabIndex and TabStop properties 2. Use Access Keys 3. Use Collections
7. To understand the for each statement.	1. Use the foreach repetition statement with a collection or an array.
8. To add menus to an application.	1. Create a menu 2. Code the Menu's Click events
9. To understand string processing.	1. Locate substrings in strings 2. Extract substrings from strings 3. Replace substrings in strings
10. To understand and use sequential files.	1. Demonstrate an understanding of data hierarchy 2. Code an event to create a file 3. Write to a sequential-access file

Course Objective	Competencies
	4. Read from a sequential-access file
11. To understand database programming.	1. Create a database connection 2. Add data command objects to a form 3. Retrieve information from a database 4. Updating information in a database
12. To understand data exception handling.	1. Code a program with try blocks and catch blocks 2. Use try Parse to handle exceptions