SPRINGFIELD TECHNICAL COMMUNITY COLLEGE

# ACADEMIC AFFAIRS

Course Number: PROG 116

Department: Computer Information Technologies

Course Title: Introduction to Structured Programming Using **C#**

Semester: Fall Year: 2011

## Objectives/Competencies

| Course Objective | Competencies |
|---|---|
| 1. To understand basic hardware-software relationships. | 1. Use the Visual Studio **C#.NET** environment to develop programs.<br>2. Understand computer memory and its organization as it pertains to execution of instructions, the sequential processing of instructions, and its relationship to the identifiers used in code. |
| 2. To develop an understanding of software engineering. | 1. Use system software and resources to create, edit, save, and retrieve source files, to execute programs, and to obtain hard copy.<br>2. Understand the difference between an interpreter and a compiler.<br>3. Interpret generated error messages to debug code.<br>4. Appropriately apply control structures and the principles of structured programming.<br>5. Write programs that have well-documented source |

| Course Objective | Competencies |
|---|---|
|  | code (meaningful names, consistent indentation, comments). |
| 3. To investigate common algorithms and techniques used in business programming applications. | 1. Understand and use arithmetic, logical, and Boolean operators and operations.<br>2. Learn, in depth, the application of sequence, repetition, and selections structures and statements; i.e., If-else if, Switch, While-do, Do-while, For, and Variations of I/O statements.<br>3. Develop and write applications which produce well-formatted and accurate business reports. |
| 4. To encourage critical thinking skills. | 1. Solve problems analytically.<br>2. Debug programs by resolving run-time and logic errors.<br>3. Evaluate existing code and revise it for efficiency and clarity.<br>4. Ability to transfer learned skills and concepts to the study of other programming languages. |
| 5. To familiarize the student with the Microsoft **C#.NET** Integrated Development Environment (IDE). | 1. Utilize the following sections of the IDE to efficiently develop C#.NET projects: Solution Explorer, Properties, Toolbox, Code Editor and Form Design Tool.<br>2. Create new or modify existing projects, forms, code modules and save them to disk.<br>3. Understand the types of files that make a C#.NET solution.<br>4. Create projects with multiple forms and multiple source files. |

| Course Objective | Competencies |
|---|---|
|  | 5. Follow and utilize standard industry GUI design standards.<br>6. Customize the Visual Studio.NET developer profile to improve developer productivity. |
| 6. Instill in students the requirement to develop programs according to industry standards. Illustrate why the standards are important and how the use of standards actually makes the development of programs easier for the developer. | 1. Code all programming assignments utilizing the comprehensive industry standards. |
| 7. Establish student proficiency using the following C#.NET Controls: Label, Text Box, Radio Button, Check Box, List Box, Combo Box as well as Menus, the Common Dialog Control and Form Objects. | 1. Create well-designed forms by dragging various controls from the toolbox onto the form. The controls will be properly sized and aligned. The visual alignment tool will be used to attain precision alignment of the controls within the form.<br>2. Name each control used on form according to industry standards.<br>3. Utilize the following properties of the controls: Name, Text, Enabled, BorderStyle, MaxLength, BackColor, Visible.<br>4. Write event handlers containing code to react to the user's interaction with the controls on the form.<br>5. Demonstrate the ability to determine when and why a particular control should be used to develop a well-designed form that conforms to Windows guidelines.<br>6. Demonstrate the ability to create intuitive and efficient menus that make a form easy and intuitive for the user. |
| 8. Stress the 80/20 rule of application development (80% of the cost of an application is spent maintaining the | 1. Develop functions and sub-routines that break the logic down to small, easily maintained groups of code. |

| Course Objective | Competencies |
|---|---|
| application over its useful life while only 20% of the cost is attributable to the initial coding of the application). Coding shortcuts and bypassing standards may save a very tiny percentage of 20% of the cost while causing a disproportionate increase in the cost to maintain the application over its useful life. | Each function of sub-routine will have a very narrow focus relative to the function it will perform.<br>2. All variable and constant names will follow the industry standards.<br>3. Name all variables and constants with names that would be meaningful to a person that is unfamiliar with the program being written.<br>4. Utilize comments to add clarity where the purpose of a function or sub-routine contains complicated logic.<br>5. Utilize the object-oriented nature of C#.NET to simplify the maintenance of the application program being written. |
| 9. Instill in the students the fact that the debugger is one of the most crucial parts of Visual Studio.NET development environment. | 1. Utilize the debugger for every programming assignment.<br>2. Utilize breakpoints to stop the execution of the program at specific points within the program.<br>3. Examine the contents of variables after the program has started executing through use of the Immediate Window and Watch, Quick Watch debugging tools. |