



**COWLEY COLLEGE
& Area Vocational Technical School**

COURSE PROCEDURE FOR

**C# PROGRAMMING
CIS1864 3 Credit Hours**

Student Level:

This course is open to students on the college level in either freshman or sophomore year.

Catalog Description of the Course:

CIS1864 - C# PROGRAMMING (3 hrs)

An introductory course to give computer science majors an introduction to programming in C#. Windows programs will be created using a structured programming approach. Various problems will be solved using C#.

Prerequisite:

None

Controlling Purpose:

This course is offered to teach programming skills in C# to students. The basic constructs learned in this course will apply to any programming language (with slight modifications).

Learner Outcomes:

Upon completion of this course, the student should be able to write programs using C#. The student should be able to use input/output statements, dialog boxes, selection statements, repetition structures, procedures, and arrays. The student will learn how to incorporate exception handling into their programs. They will also using the following windows components: buttons, labels, lists, textboxes, panels, menus, tabbed windows, ListView, TreeView, LinkLabel, ListBoxes, ComboBoxes and miscellaneous components. The students will also gain understanding of how to work with objects. Lastly, they will know how to write to and retrieve data from various sources.

Units Outcomes and Criterion Based Evaluation Key for Core Content:

The following defines the minimum core content not including the final examination period. Instructors may add other content as time allows.

Evaluation Key:

- A = All major and minor goals have been achieved and the achievement level is considerably above the minimum required for doing more advanced work in the same field.
- B = All major goals have been achieved, but the student has failed to achieve some of the less important goals. However, the student has progressed to the point where the goals of work at the next level can be easily achieved.
- C = All major goals have been achieved, but many of the minor goals have not been achieved. In this grade range, the minimum level of proficiency represents a person who has achieved the major goals to the minimum amount of preparation necessary for taking more advanced work in the same field, but without any major handicap of inadequacy in his background.

- D = A few of the major goals have been achieved, but the student's achievement is so limited that he is not well prepared to work at a more advanced level in the same field.
- F = Failing, will be computed in GPA and hours attempted.
- N = No instruction or training in this area.

CHAPTER 1: Introduction to Computers, the Internet, the Web and C#

Outcomes: Upon completion of this unit, the student will understand the evolution of the Internet. They will also gain an understanding of basic computer concepts and the history of the C# programming language. The student will also gain an understanding of the .NET initiative.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand basic computer concepts.
						Learn about various programming languages.
						Become familiar with the history of the C# programming language.
						Understand the Microsoft .NET initiative.

CHAPTER 2: Introduction to the Visual Studio .NET IDE

Outcomes: Upon completion of this unit, the student will have a working knowledge of the Visual Studio .NET IDE and be able to write, compile, and execute a simple Visual C# program.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Become familiar with the Visual Studio .NET integrated development environment (IDE).
						Become familiar with the types of commands contained in the IDE's menus and toolbars.
						Identify and understand the use of various kinds of windows in Visual Studio .NET.
						Understand the features provided by the Toolbox.
						Understand Visual Studio .NET's help features.
						Create, compile and execute a simple C# program.

CHAPTER 3: Introduction to C# Programming

Outcomes: Upon completion of this unit, the student will be able to write programs that use input/output statements, arithmetic operators, equality/relational operators and decision-making statements

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Write simple C# programs.
						Use input and output statements.
						Become familiar with primitive data types.
						Understand basic memory concepts.
						Use arithmetic operators.
						Understand the precedence of arithmetic operators.
						Write decision-making statements.
						Use relational and equality operators.

CHAPTER 4: Control Statements: Part 1

Outcomes: Upon completion of this unit, the student will be able to write programs that use the If, If/Else, While, and increment/decrement/assignment operators.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand basic problem-solving techniques of programming.
						Develop algorithms through the process of top-down, stepwise refinement.
						Use the if and if...else selection statements to choose among alternative actions.
						Use the while repetition statement to execute statements in a program repeatedly.
						Understand counter-controlled repetition and sentinel-controlled repetition.
						Use the increment, decrement and assignment operators.

CHAPTER 5: Control Statements: Part 2

Outcomes: Upon completion of this unit, the student will be able to write programs that use the Do/While, Switch, Break and Continue statements. They will also know how to use logical operators.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Use the for and do...while repetition statements to execute statements in a program repeatedly.
						Understand multiple selection with switch selection statement.
						Use the break and continue program-control statements.
						Use the logical operators.

CHAPTER 6: Methods

Outcomes: Upon completion of this unit, the student will be able to construct programs in a modular sense and understand the scope of the variables and calling techniques.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Construct programs modularly from small pieces called methods.
						Become familiar with the common math methods available in the Framework Class Library.
						Create new methods.
						Understand the mechanisms for passing data between methods.
						Introduce simulation techniques that use random number generation.
						Understand how the visibility of identifiers is limited to specific regions of programs.
						Understand how to write and use methods that call themselves.

CHAPTER 7: Arrays						
Outcomes: Upon completion of this unit, the student will be able to write programs that use arrays.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Become familiar with the array data structure.
						Understand how managed arrays store, sort and search lists and tables of values.
						Understand how to declare and initialize a managed array.
						Be able to refer to individual elements of a managed array.
						Understand how to pass arrays to methods.
						Understand basic sorting techniques.
						Be able to declare and manipulate multidimensional managed arrays.

CHAPTER 8: Object-Based Programming						
Outcomes: Upon completion of this unit, the student will be able to write programs that use objects. The student will gain an understanding of some of the basic concepts of objects and how to work with them.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand encapsulation and data hiding.
						Understand the concepts of data abstraction and abstract data types (ADTs).
						Create, use and destroy objects.
						Control access to data members and methods.
						Use properties to keep objects in consistent states.
						Understand the use of the this reference.
						Understand namespaces and assemblies.
						Use the Class View in Visual Studio .NET.

CHAPTER 11: Exception Handling						
Outcomes: Upon completion of this unit, the student will be able to incorporate error checking into their programs.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand exceptions and error handling.
						Use try blocks to delimit code in which exceptions may occur.
						Throw exceptions.
						Use Catch blocks to specify exception handlers.
						User the Finally block to release resources.
						Understand the C# exception-class hierarchy.
						Create programmer-defined exceptions.

CHAPTER 12: Graphical User Interface Concepts: Part 1						
Outcomes: Upon completion of this unit, the student will be able to incorporate the graphical controls: buttons, labels, lists, textboxes, and panels into their programs. The student also will get an introduction to events.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand the design principles of graphical user interfaces.
						Understand, use and create event handlers.
						Understand namespaces that contain graphical user interface components and event-handling classes and interfaces.
						Be able to create graphical user interfaces.
						Be able to create and manipulate buttons, labels, lists, textboxes, and panels.
						Be able to use mouse and keyboard events.

CHAPTER 13: Graphical User Interface Concepts: Part 2

Outcomes: Upon completion of this unit, the student will be able to incorporate the graphical controls: menus, tabbed windows, ListView, TreeView, LinkLabel, Listboxes, and Comboboxes into their programs. Also the student will know how to create a multiple-document-interface program.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to display lists using list boxes and combo boxes.
						Understand the use of the ListView and TreeView controls for displaying information.
						Be able to create menus, window tabs and multiple-document-interface (MDI) programs.
						Create custom controls.

CHAPTER 15: Strings, Characters and Regular Expressions

Outcomes: Upon completion of this unit, the student will be able to manipulate string/character objects and understand how to use regular expressions.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Create and manipulate immutable character string objects of class String.
						Create and manipulate mutable character string objects of class StringBuilder.
						Use regular expressions in conjunction with classes Regex and Match.

CHAPTER 17: Files and Streams

Outcomes: Upon completion of this unit, the student will be able to work with files and directory structures.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to create, read, write, and update files.
						Understand the C# streams class hierarchy.
						Be able to use classes File and Directory.
						Be able to use the FileStream and BinaryFormatter classes to read objects from, and write objects to, files.
						Become familiar with sequential-access and random-access file processing.

CHAPTER 19: Database, SQL and ADO.NET

Outcomes: Upon completion of this unit, the student will be able to access relational databases.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand the relational database model.
						Understand basic database queries written in Structured Query Language (SQL).
						Understand and use ADO.NET's disconnected model.
						Use the classes and interfaces of namespace System::Data to manipulate databases.
						Use the classes and interfaces of namespace System::Data::OleDb.

Projects Required:

Projects will vary according to the instructor.

Text Book:

Contact the bookstore for current textbook information.

Materials/Equipment Required:

Traditional Classroom Delivery: None. Although installing Visual Studio .NET 2003 at home will aid in doing the coursework.

Attendance Policy:

Students should adhere to the attendance policy outlined by the instructor in the course syllabus.

Grading Policy:

The grading policy will be outlined by the instructor in the course syllabus.

Maximum Class Size:

Based on classroom occupancy.

Course Time Frame:

The U.S. Department of Education, Higher Learning Commission and the Kansas Board of Regents define credit hour and have specific regulations that the college must follow when developing, teaching and assessing the educational aspects of the college. A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than one hour of classroom or direct faculty instruction and a minimum of two hours of out-of-class student work for approximately fifteen weeks for one semester hour of credit or an equivalent amount of work over a different amount of time, The number of semester hours of credit allowed for each distance education or blended hybrid courses shall be assigned by the college based on the amount of time needed to achieve the same course outcomes in a purely face-to-face format.

Refer to the following policies:

[402.00 Academic Code of Conduct](#)

[263.00 Student Appeal of Course Grades](#)

[403.00 Student Code of Conduct](#)

Disability Services Program:

Cowley College, in recognition of state and federal laws, will accommodate a student with a documented disability. If a student has a disability which may impact work in this class which requires accommodations, contact the Disability Services Coordinator.