



**COWLEY COLLEGE
& Area Vocational Technical School**

COURSE PROCEDURE FOR

**C PROGRAMMING
CIS1876 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:

CIS1876 C PROGRAMMING (3 hrs)

An introductory course to give computer science majors an introduction to programming in C. Applications will be created using a structured programming approach. Object oriented topics in C++ will be covered. 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 the C programming language. The student should be able to use input/output statements, selection statements, repetition structures, procedures, and arrays. The student will learn how to incorporate exception handling into their programs. 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 and the World Wide Web

Outcomes: Upon completion of this unit, the student will understand what computers are, how they work and how they are programmed. Also the student will understand the notion of structured programming and why this set of techniques has fostered a revolution in the way programs are written.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand basic computer concepts.
						Become familiar with different types of programming languages.
						Become familiar with the history of the C programming language.
						Become aware of the C Standard Library.
						Understand the elements of a typical C program development environment.
						Appreciate why it is appropriate to learn C in a first programming course.
						Appreciate why C provides a foundation for further study of programming languages in general and of C++, Java and C# in particular.
						Become familiar with the history of the Internet and the World Wide Web.

CHAPTER 2: Introduction to C Programming

Outcomes: Upon completion of this unit, the student will have a concise introduction to writing C programs and understand decision making and arithmetic operations.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to write simple computer programs in C.
						Be able to use simple input and output statements.
						Become familiar with fundamental data types.
						Understand computer memory concepts.
						Be able to use arithmetic operators.
						Understand the precedence of arithmetic operators.
						Be able to use simple decision-making statements.

CHAPTER 3: Structured Program Development in C

Outcomes: Upon completion of this unit, the student will be introduced to algorithms for solving problems, control statements: sequence, selection, and repetition.

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

CHAPTER 4: C Program Control

Outcomes: Upon completion of this unit, the student will be introduced to more control statements: for and switch. Also the students will understand the logical operators.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to use the "for" and do...while repetition statements.
						Understand multiple selection using the switch selection statement.
						Be able to use the break and continue program control statements.
						Be able to use the logical operators.

CHAPTER 5: C Functions

Outcomes: Upon completion of this unit, the student will be able to understand the design and construction of program modules.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand how to construct programs modularly from small pieces called functions.
						Introduce the common math functions available in the C standard library.
						Be able to create new functions.
						Understand the mechanisms used to pass information between functions.
						Introduce simulation techniques using random number generation.
						Understand how to write and use functions that call themselves.

CHAPTER 6: C Arrays

Outcomes: Upon completion of this unit, the student will understand the structuring of data into arrays of related data items of the same type.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Introduce the array data structure.
						Understand the use of arrays to store, sort and search lists and tables of values.
						Understand how to define an array, initialize an array and refer to individual elements of an array.
						Be able to pass arrays to functions.
						Understand basic sorting techniques.
						Be able to define and manipulate multiple subscripted arrays.

CHAPTER 7: C Pointers

Outcomes: Upon completion of this unit, the student will understand pointer operators, call by reference, pointer expressions, pointer arithmetic, the relationship between pointers and arrays, arrays of pointers, and pointers to functions.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand pointers and pointer operators.
						Be able to use pointers to pass arguments to functions by reference.
						Understand the close relationships among pointers, arrays and strings.
						Understand the use of pointers to functions.
						Be able to define and use arrays of strings.

CHAPTER 8: C Characters and Strings

Outcomes: Upon completion of this unit, the student will understand character and string processing functions available in C's libraries.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to use the functions of the character handling library (ctype).
						Be able to use the string and character input/output functions of the standard input/output library (stdio).
						Be able to use the string conversion functions of the general utilities library (stdlib).
						Be able to use the string processing functions of the string handling library (string).
						Appreciate the power of function libraries as a means of achieving software reusability.

CHAPTER 9: C Formatted Input/Output

Outcomes: Upon completion of this unit, the student will understand how to use the powerful formatting capabilities of printf and scanf.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand input and output streams.
						Be able to use all print formatting capabilities.
						Be able to use all input formatting capabilities.
						Be able to print with field widths and precisions.
						Be able to use formatting flags in the printf format control string.
						Be able to use literals and escape sequences.

CHAPTER 10: C Structures, Unions, Bit Manipulations and Enumerations

Outcomes: Upon completion of this unit, the student will understand how to use structures, unions, bit manipulations and enumerations in a C program.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to create and use structures, unions and enumerations.
						Be able to pass structures to functions by value and by reference.
						Be able to manipulate data with the bitwise operators.
						Be able to create bit fields for storing data compactly.

CHAPTER 11: C File Processing

Outcomes: Upon completion of this unit, the student will understand how to process text files with sequential access and random access.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to create, read, write and update files.
						Become familiar with sequential access file processing.
						Become familiar with random-access file processing.

CHAPTER 12: C Data Structures

Outcomes: Upon completion of this unit, the student will understand the techniques used to create and manipulate dynamic data structures.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to allocate and free memory dynamically for data objects.
						Be able to form linked data structures using pointers, self-referential structures and recursion.
						Be able to create and manipulate linked lists, queues, stacks and binary trees.
						Understand various important applications of linked data structures.

CHAPTER 13: C Preprocessor

Outcomes: Upon completion of this unit, the student will understand the preprocessor directives.

A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to use #include for developing large programs.
						Be able to use #define to create macros and macros with arguments.
						Understand conditional compilation.
						Be able to display error messages during conditional compilation.
						Be able to use assertions to test if the values of expressions are correct.

CHAPTER 14: Other C Topics						
Outcomes: Upon completion of this unit, the student will cover miscellaneous topics involved in C programming.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Be able to redirect keyboard input to come from a file.
						Be able to redirect screen output to be placed in a file.
						Be able to write functions that use variable-length argument lists.
						Be able to process command-line arguments.
						Be able to assign specific types to numeric constants.
						Be able to use temporary files.
						Be able to process unexpected events within a program.
						Be able to allocate memory dynamically for arrays.
						Be able to change the size of memory that was dynamically allocated previously.

Projects Required:

Projects will vary according to the instructor.

Text Book:

Contact the bookstore for current textbook.

Materials/Equipment Required:

On-line Delivery: C Programming Language (free download from internet)

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.