

Colorado BMEF CTE Course – Scope and Sequence

Course Name	Introduction to Programming	Course Details	Credits = .5 (Semester) Level 3
		Course = 0.50 Carnegie Unit Credit	
Course Description	Focuses on a general introduction to computer programming. This course emphasizes the design and implementation of structured and logically correct programs with good documentation. It is centered on basic programming concepts, including control structures, modularization, and data processing. A structured programming language is used to implement program designs. It emphasizes the writing of multiple programs following the software development process, from start to finish, including design, implementation, and testing.		
Note	This is a suggested scope and sequence for the course content in a BMEF Pathway. The content will work with any textbook or instructional resource. If locally adapted, make sure all competencies are covered. Course aligns to CCNS (CSC1019). Additional resources may be available through the Technology CTE Pathway . The appropriate Technology/IT Credential is required to teach this class in the BMEF pathway. Please contact your local community college for further information regarding opportunities for credit.		
SCED Identification # 10151		Schedule calculation based on 60 calendar days of a 90-day semester. Scope and sequence allows for additional time for guest speakers, student presentations, field trips, remediation, or other content topics.	
All courses taught in an approved CTE program must include Essential Skills embedded into the course content. The Essential Skills Framework for this course can be found at https://www.cde.state.co.us/standardsandinstruction/essentialskills			
Course Competencies The competencies Students will demonstrate in this course are as follows: A. Input data from the computer keyboard using standard input and output commands via a programming language. B. Create a program that demonstrates the use of variables and constants. C. Program statements that show a basic understanding of types of variables, at a minimum integer, real, character and string. D. Design a program that outputs processed data to the computer screen. E. Develop and implement programs from problem statements. F. Write multiple programs using control structures and functions/modules. G. Write functions/modules proving a basic understanding of variable scope. H. Design, write, and modify a program using 1D array processing. I. Document programs with standards required in the class. J. Convert numbers from base 10 to binary and hexadecimal.			

Instructional Unit Topic	Suggested Length of Instruction	CTE or Academic Standard Alignment	Competency / Performance Indicator	Outcome / Measurement	CTSO Integration
MODULE 1: Number Systems & Intro Programming					
Number system conversions (Decimal ↔ Binary ↔ Hex)		IT Programming & Software Dev – Data Representation	Convert numbers from base 10 to binary and hexadecimal	Students accurately convert between number systems with 90% accuracy on assessment	**FBLA: Coding & Programming Computer Game & simulation Programming **Appropriate throughout the course**
Binary ↔ Hex relationships		Computational Thinking	Convert binary numbers to hexadecimal and vice versa	Students demonstrate conversions through exercises and quizzes	
Intro Python input/output		Programming Fundamentals	Input data using keyboard and output results	Program prompts user and displays correct output	
Problem-solving program design			Develop programs from problem statements	Completed Python program that meets input/process/output requirements	
Program documentation basics		Professional Practices	Document programs with required standards	Code includes comments, meaningful variable names, and formatting	
MODULE 2: Program Design & Variables					
Variables and constants		Programming Fundamentals	Create programs using variables and constants	Program demonstrates correct declaration and use of variables	
Data types (int, real, char, string)		Data & Software Dev	Demonstrate understanding of variable types	Students correctly assign and manipulate multiple data types	
Flowcharting		Systems Design	Design algorithms using flowcharts	Completed flowchart representing program logic	
Pseudocode development		Computational Thinking	Write pseudocode using proper syntax	Students produce structured pseudocode matching a problem	
IPO model (Input–Process–Output)			Design program behavior with inputs, processes, outputs	Students create flowchart + pseudocode + working code	

Program development (Python)		Programming & Software Dev	Develop program from pseudocode to Python	Final working program matches design	
MODULE 3: Input/Output & Looping					
Keyboard input & screen output		Programming Fundamentals	Design program with input/output	Functional program using user input and displayed output	
Loop structures (iteration)		Algorithms & Control Structures	Use loop structures in programs	Programs correctly implement repetition (for/while loops)	
Integrated program design		Software Development Lifecycle	Develop programs combining I/O and looping	Completed program solving real-world scenario using loops	
MODULE 4: Functions, Scope & Arrays					
Functions & modules		Modular Programming	Write functions/modules	Program includes reusable functions	
Variable scope (local/global)		Programming Concepts	Demonstrate understanding of variable scope	Students correctly use local/global variables in functions	
Lists / 1D arrays		Data Structures	Design program using 1D array processing	Program processes lists (store, access, modify data)	
Modular program design		Software Engineering	Develop structured programs using modules	Multi-function program with organized logic	
MODULE 5: Advanced Structures & Data Selection					
Control structures (selection & iteration)		Algorithms & Logic	Write programs using control structures	Program uses if/else and loops correctly	
Program development (full integration)		Software Development	Develop programs from problem statements	Completed multi-step program meeting requirements	
Data structures selection		Data & Analysis	Select appropriate data structure	Students justify and implement correct structure (e.g., list vs variable)	
Modular programming integration		Software Engineering	Write programs using functions/modules and control structures	Fully functional program integrating all concepts	

Academic Standards Alignment

Colorado CTE courses are required to integrate academic and essential skills into course content. The following academic standards have been identified as aligned to this course and should be integrated throughout instruction.

[CO CTE Academic Standards Resources](#) (Rubrics, integration strategies, etc)

[CDE Colorado Academic Standards Online](#) (Online standards lookup tool)

Reading, Writing, & Communicating

Standard 2: Reading for All Purposes

- **Prepared Graduate 5:** Understand how language functions in different contexts, command a variety of word-learning strategies to assist comprehension, and make effective choices for meaning or style when writing and speaking.

Standard 3: Writing and Composition

- **Prepared Graduate 7:** Craft informational/explanatory texts using techniques specific to the genre.
- **Prepared Graduate 9:** Demonstrate mastery of their own writing process with clear, coherent, and error-free polished products.

Standard 4: Research Inquiry and Design

- **Prepared Graduate 10:** Gather information from a variety of sources; analyze and evaluate its quality and relevance; and use it ethically to answer complex questions.

Note: Mathematics and Science standards will be added in the future. Visit the [Colorado Academic Standards](#) page for the most current version.