



Course Number and Name: CST 117 Section Name - *Programming in C++*

Term Dates: Section Start Date - Section End Date

Meeting Location: Section Meeting Building Section Meeting Room

Meeting Days & Times: Section Meeting Days; Section Start Time - Section End Time

Lecture/Lab Hours: 3.00 Lecture Hours

Credits: 3.00

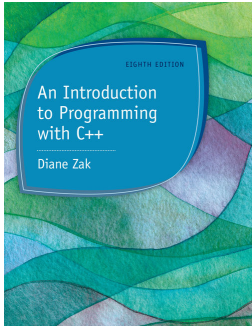
Course Description: This course serves as a comprehensive introduction to the fundamentals of computer systems and structured programming techniques. It covers essential topics such as problem-solving, flowcharts, programming methodology, and delves into the intricacies of hardware and software. Throughout the course, emphasis is placed on practical application, with hands-on experience using the C++ programming language.

Course Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Demonstrate a comprehensive understanding of fundamental programming concepts and methodologies, providing the foundation knowledge necessary for proficient C++ programming.
2. Develop the ability to design clear and efficient syntax and semantics in the C++ programming language.
3. Implement different functions, various data types, and basic operators.
4. Define the use of input and output in C++.
5. Declare and manipulate arrays, pointers, and dynamic memory.

Course Materials



An Introduction to Programming with C++

Authors: Diane Zak

Publisher: Cengage

Publication Date: 2016

Edition: 8th ed.

Course Requirements

- An IDE or Integrated development environment, enables students to write code to build and debug their programs.
- Internet access to login to Canvas LMS
- USB drive or cloud storage, such as Microsoft OneDrive or Google Drive to save all assignments

Evaluation Methods

Evaluation Methods	Weight
3 Exams	45%
Final Project	25%
Homework & Programming Assignments	25%
Class Attendance & Participation	5%

Grading

Grading scale

Letter Grade	Percentage
A	90-100
B+	87-89
B	80-86

Letter Grade	Percentage
C+	77-79
C	70-76
D+	67-69
D	60-66
F	< 60

Class Schedule

Week	Unit/Content	Learning Activities
1	Course Introduction Review Canvas	Read course syllabus Login to Canvas and introduce IDE Chapter 1
2 & 3	Beginning problem solving Variables and constants	Read Chapter 2 and 3 in the book and complete all the assignments described on Canvas
4	Making Decisions Nested If Flowcharts	Read Chapter 5, and 6 in the Books, Making Decisions, and complete all assignments described on canvas
5	Exam 1	Chapter 1,2,3,5, and 6
5 & 6	Loops Flowcharts	Read Chapter 7 in the Books, Loops, and complete all assignments described on canvas
7	More Loops Flowcharts	Read Chapter 8 in the Books, More Loops, and complete all assignments described on canvas

8	Exam 2	Chapters 7 and 8
8 & 9	Value returning Functions	Read Chapter 9 in the Books, Functions, and complete all assignments described on canvas
9 & 10	Void Functions	Read Chapter 10 in the Books, Functions, and complete all assignments described on canvas
11	Array	Read Chapter 11 in the Books, Functions, and complete all assignments described on canvas
11 & 12	Exam 3	Chapters 9, 10, and 11
12	Classes and Objects	Read Chapter 15 in the Books, Classes and Objects and complete all assignments described on canvas
13	Pointers	Pointers and complete all assignments described on canvas
14	Introduce Linked List	Introduce linked list and complete all assignments described in canvas
15	Final Exam	Pointers and Linked list

Experiential Learning

Students must complete an experiential learning activity that connects course content to career applications. This activity may be a content specific assignment or practical skill that is applied within a course assignment. This assignment supports the general education learning outcomes of scientific/critical thinking and quantitative reasoning; oral and written communication; and information literacy/technological competency.

Academic Policies

See College Catalog for more information: <http://onlinecatalog.ucc.edu/index.php>

Americans with Disabilities Act (ADA)

UCNJ Union College of Union County, NJ offers reasonable accommodations and/or services to persons with disabilities. Any student who has a documented disability and wishes to self-identify should contact the Director of Universal Accessibility Services and Veterans Affairs at (908) 709-7164, or email accessibility@ucc.edu. Accommodations

are individualized and in accordance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1992. In order to receive accommodations, students must be registered with Disability Support Services. Students should register with the office as soon as possible. Accommodations are not official until the Faculty Accommodations Alert Form(s) are issued from the student to his/her instructor(s).

Family Educational Rights and Privacy Act (FERPA)

The FERPA Statement can be found at <https://www.ucc.edu/admissions/the-family-education-rights-and-privacy-act/>.

Equal Opportunity Statement

UCNJ Union College of Union County, NJ does not discriminate and prohibits discrimination, as required by state and/or federal law, in all programs and activities, including employment and access to its career and technical programs.

UCNJ Union College of Union County, NJ Mission Statement

Transforming Our Community. . . One Student at a Time

Suggested Teaching Methodologies

Lecture, group discussion, pair programming assignments, multimedia/technology, projects, demonstrations, etc.

Mapping Course Learning Outcomes to Learning Activities and Evaluation Methods

Course Learning Outcomes (CLOs)	Learning Activities	Evaluation Methods
Demonstrate a comprehensive understanding of	Interactive Coding Exercises: Engage students in hands-on	Programming assignments. Peer Review.

fundamental programming concepts and methodologies, providing the foundation knowledge necessary for proficient C++ programming.	coding exercises where they apply fundamental programming concepts such as variables, control structures, and functions in C++. These exercises could include small programming tasks or challenges that progressively increase in complexity	Project.
Develop the ability to design clear and efficient syntax and semantics in the C++ programming language.	Code Analysis and Refactoring: Provide students with code snippets or small programs written in C++. Task them with analyzing the code to identify inefficiencies, ambiguities, or areas for improvement in terms of syntax and semantics. Encourage students to refactor the code to make it clearer, more concise, and more efficient while preserving its functionality.	Programming assignments. Peer Review. Project.
Implement different functions, various data types, and basic operators.	Implement different functions, various data types, and basic operators through interactive exercises and challenges, encouraging students to apply their understanding of C++ concepts in practical scenarios.	Programming assignments. Peer Review. Project.

Define the use of input and output in C++.	Engage students in coding exercises and real-world application analysis to explore the use of input and output in C++, covering concepts such as reading input from various sources, producing formatted output, handling errors, and manipulating files.	Programming assignments. Peer Review. Project.
Declare and manipulate arrays, pointers, and dynamic memory.	Engage students in exercises and workshops to declare and manipulate arrays, pointers, and dynamic memory in C++, covering concepts such as array manipulation, pointer declaration, dereferencing, pointer arithmetic, and dynamic memory allocation.	Programming assignments. Peer Review. Project.

Please note: all programs must integrate in one or more courses, discipline-specific course learning outcomes that reflect the College learning outcomes of scientific/critical thinking and quantitative reasoning, oral/written communication, and information literacy.

Visit <https://app.teachinghow2s.com/library/science-technology-engineering-math-stem> to access evidence-based-teaching techniques for this course.