Washtenaw Community College Comprehensive Report

CPS 171 Introduction to Programming with C++
Effective Term: Fall 2019

Course Cover

Division: Business and Computer Technologies
Department: Computer Science & Information Technology
Discipline: Computer Science
Course Number: 171
Org Number: 13420
Full Course Title: Introduction to Programming with C++
Transcript Title: Intro Prog With C++
Is Consultation with other department(s) required: No
Publish in the Following: College Catalog , Time Schedule , Web Page
Reason for Submission: Three Year Review / Assessment Report
Change Information:
Consultation with all departments affected by this course is required.
Other:
Rationale: Update Master Syllabus after assessment report.
Proposed Start Semester: Fall 2019
Course Description: In this course, students are introduced to programming using the C++ language. Students learn about problem solving strategies, top-down program development and programming style. Topics include sequential, decision and iterative control structures, functions, basic data structures and an introduction to classes. Students write and execute approximately eight C++ programs.

Course Credit Hours

Variable hours: No
Credits: 4
Lecture Hours: Instructor: 60 **Student:** 60
Lab: Instructor: 0 **Student:** 0
Clinical: Instructor: 0 **Student:** 0

Total Contact Hours: Instructor: 60 **Student:** 60
Repeatable for Credit: NO
Grading Methods: Letter Grades
Audit
Are lectures, labs, or clinicals offered as separate sections?: NO (same sections)

College-Level Reading and Writing

College-level Reading & Writing

College-Level Math

Level 4

Requisites

General Education

Degree Attributes
Statewide articulation approved

General Education Area 7 - Computer and Information Literacy
Assoc in Arts - Comp Lit
Assoc in Applied Sci - Comp Lit
Assoc in Science - Comp Lit

Request Course Transfer

Proposed For:
University of Michigan

Student Learning Outcomes

1. Identify appropriate use of simple programming constructs including loops and conditional logic.
   Assessment 1
   Assessment Tool: Test questions
   Assessment Date: Winter 2022
   Assessment Cycle: Every Three Years
   Course section(s)/other population: All sections
   Number students to be assessed: All students
   How the assessment will be scored: Answer key
   Standard of success to be used for this assessment: 70% of the students will score 70% or higher
   Who will score and analyze the data: Departmental faculty

2. Identify appropriate use of simple object-oriented concepts such as constructors, functions and overriding functions.
   Assessment 1
   Assessment Tool: Test questions
   Assessment Date: Winter 2022
   Assessment Cycle: Every Three Years
   Course section(s)/other population: All sections
   Number students to be assessed: All students
   How the assessment will be scored: Answer key
   Standard of success to be used for this assessment: 70% of the students will score 70% or higher
   Who will score and analyze the data: Departmental faculty

3. Identify appropriate use of arrays.
   Assessment 1
   Assessment Tool: Test Questions
   Assessment Date: Winter 2022
   Assessment Cycle: Every Three Years
   Course section(s)/other population: All sections
   Number students to be assessed: All students
   How the assessment will be scored: Answer key
   Standard of success to be used for this assessment: 70% of the students will score better than 70%
   Who will score and analyze the data: Departmental faculty

4. Develop C++ code that uses concepts and constructs.
   Assessment 1
   Assessment Tool: Programming exercises
   Assessment Date: Winter 2022
   Assessment Cycle: Every Three Years
   Course section(s)/other population: All sections

Number students to be assessed: 25% of all students with a minimum of one full section
How the assessment will be scored: Departmentally-developed rubric
Standard of success to be used for this assessment: 70% of the students will create a program that executes successfully
Who will score and analyze the data: Departmental faculty

Course Objectives

1. Edit, compile, execute, and get hard copy of a simple program.
2. Use good documentation, formatting and naming conventions to ensure program readability.
3. Write a program using the C++ arithmetic operators, input/output methods and appropriate manipulators for formatting.
4. Write a program using appropriate selection statements such as "if-else" and "switch".
5. Write a program using appropriate looping statements such as "while", "for", and "do-while".
6. Write a program using functions with parameters passed by value, by reference and by pointer.
7. Use structures in a program.
8. Write a program using classes with data members, member functions and constructors.
9. Use both one-dimensional and multi-dimensional arrays.
10. Describe different sorting and searching algorithms.
11. Use character data and string processing.
12. Write a program using Enums.

New Resources for Course

OER material and Lynda Videos

Course Textbooks/Resources

Textbooks
Manuals
Periodicals
Software

Equipment/Facilities

Data projector/computer

Reviewer	Action	Date
Faculty Preparer:		
Khaled Mansour	*Faculty Preparer*	*Jun 26, 2019*
Department Chair/Area Director:		
Philip Geyer	*Recommend Approval*	*Jul 22, 2019*
Dean:		
Eva Samulski	*Recommend Approval*	*Jul 22, 2019*
Curriculum Committee Chair:		
Lisa Veasey	*Recommend Approval*	*Aug 06, 2019*
Assessment Committee Chair:		
Shawn Deron	*Recommend Approval*	*Aug 19, 2019*
Vice President for Instruction:		
Kimberly Hurns	*Approve*	*Aug 19, 2019*