Inver Hills Community College

CS 1119: Computer Programming with C++

A. COURSE DESCRIPTION

Credits: 4

Lecture Hours/Week: 4

Lab Hours/Week: 0

OJT Hours/Week: *.*

Prerequisites:
This course requires any of these nine prerequisites
MATH 0940 - Intermediate Algebra (Minimum grade: 2.0 GPA Equivalent and Number of Years Valid: 2)
A score of 50 on test Accuplacer College Level Math
A score of 8 on test ACCP local Math History
A score of 250 on test Accuplacer NG Advanced Algebra Functions
A score of 250 on test Accuplacer NG COMP Advanced Algebra Func
A score of 1158 on test MN Comprehensive Assessment Math
A score of 22 on test ACT Math
A score of 530 on test SAT Math Composite
Algebra College Level

Corequisites: None

MnTC Goals: None

Focuses on the use of the C++ language to provide a comprehensive introduction to fundamental programming concepts for students interested in exploring computer programming. Program logic, algorithm design, object-oriented programming, and fundamental programming structures are emphasized. Course activities focus on the use of C++ to design, code, debug, and test programs of moderate complexity. Gaining knowledge of these concepts and techniques will prepare the student to develop programs to solve real-life problems.

B. COURSE EFFECTIVE DATES: 01/01/1998 - Present

C. OUTLINE OF MAJOR CONTENT AREAS

1. Input/output using standard input/output.
   Input/output using secondary storage.
   Convert mathematical formulas into equivalent programming language expressions.
   Convert an algorithm in pseudocode or diagrams into a high-level programming language.
   Create appropriate test cases to verify the correctness of the program.
   Sequential structures.
   Selection structures.
   Repetition structures.
2. Use structured programming concepts to write efficient and elegant programs.
   Modularization, procedures, and functions with use of parameters.
   Employ recursion.
   Arrays.
   Sequential and binary search.
   Insertion, selection, and bubble sort.
   Pointers/Object References.
   Introduction to objects.

D. LEARNING OUTCOMES (General)

1. Apply consistent documentation and program style standards that contribute to the readability and maintainability of software.
   Develop, design, analyze, and implement logic within a program that solves a problem with a finite number of operations.
   Design and develop programs that implement fundamental logic structures of sequence, selection, and repetition.
   Write programs that uses file I/O to provide persistence across multiple executions.
2. Develop proficiency in specification and use of appropriate primitive data types and their aggregation into simple linear data structures.
   Decompose problems into clearly defined sub-problems based on program requirements.
   Implement algorithms utilizing recursive structures.
   Create appropriate test cases and use debugging skills to verify correctness of output.

E. Minnesota Transfer Curriculum Goal Area(s) and Competencies

None

F. LEARNER OUTCOMES ASSESSMENT

As noted on course syllabus

G. SPECIAL INFORMATION

None noted