Bergen Community College
Computer Science Department
Course Syllabus

Instructor: _____   **Phone:** _____

Email: _____   **Office hours**_____

Course Title: CIS-265 Advanced Computer Concepts

Prerequisites: CIS-165 Fundamentals of Programming

Credits/Hours: 3 Credits   3 Lecture / 1 Lab

Gen'l Ed. Course: No

Course Description:

Advanced Computer Concepts is a continuation of CIS-165 Fundamentals of Programming. Topics considered include operators; functions; structured programming principles; pointer arithmetic; multi-dimensional arrays; fundamental sorting and searching algorithms; structures; unions; sequential and random access file processing algorithms; and the run-time behavior of programs.

Student Learning Outcomes: Upon completion of the course, the student will:

1. Be able to design a program using structured development techniques.
2. Know how to implement a structured design using functions.
3. Understand the structure and processing of single and multi-dimensional arrays.
4. Know the fundamental algorithms for creating and processing sequential access files and random access files.
5. Be able to organize and represent data using structures.
6. Understand the fundamental techniques for sorting and searching data sets.
7. Know the role that the preprocessor and linker play in the program development process.

Student Learning Outcomes Assessment Measurement:

Each of the above listed student learning outcomes will be assessed by: (1) written assignments and/or quizzes; (2) written examinations and a comprehensive final exam.

Course grade: see the grading policy for the course.

Textbook: Starting Out with C++ Brief Version, 7th Edition, Tony Gaddis, Pearson/Addison-Wesley, 2012, 978-0-13-2772891

Course Content:

1. <u>Structured Development Techniques</u>
   Style guidelines for C++ source code
   Creating a program template
   Top-down functional decomposition and structure charts
   Function declaration, definition, and call
   Program development – the preprocessor and linker
   Scope of a variable
   Return statement
   Reference types and pass by reference
   Menu-driven systems

2. <u>Arrays and Pointers</u>
   Arrays as function arguments
   Array processing algorithms
   Addresses and pointer variables
   Accessing arrays using pointer arithmetic

3. <u>Introduction to Data Files</u>
   Text files versus binary files
   File I/O in C++ - stream operating modes
   Text files – creation and access

4. <u>Sequential Access Files</u>
   Record and file terminology
   Structures and nested structures
   Binary file organization
   Creating and processing a sequential access file
   Algorithms for processing a sequential access file

5. <u>Searching and Sorting Techniques</u>
   Sequential search algorithm – analysis and implementation
   Boolean flags and the proper use of functions
   Arrays of structures
   Sequential search for matching records
   Sorting algorithms

6. <u>Modifying a Sequential Access File</u>
   Appending records
   Inserting records – file merge algorithms
   Deleting records

7. <u>Random Access Files</u>
   File functions for random access
   Binary search algorithm – analysis and implementation
   Editing a record

8. <u>Multi-Dimensional Arrays</u>
   Definitions and declaration forms
   Table handling algorithms