CS-203 Algorithmic Problem Solving II in C++

Pre-requisite: MA-441 and C or better in CS-101 Hours: 3 Class Hours 2 Laboratory Hours 4 credits

Course Description: User defined data types, pointers, abstract data types, vectors, queues, lists, elementary memory management, inheritance and polymorphism. Object oriented problem solving.

General Education Objectives: Use analytical reasoning skills to identify issues or problems and evaluate evidence in order to make informed decisions; reason quantitatively and mathematically as required in their fields of interest and in everyday life; integrate knowledge and skills in their program of study; use information management and technology skills effectively for academic research and lifelong learning.

Course Objectives/Expected Student Learning Outcomes: Introduction to problem solving methods and algorithm development: designing, debugging, and documenting programs in C++.

(Other texts will also be referenced: How to program in C++, 8th edition by Deitel and Deitel and C++ an Introduction to computing, 3rd edition by Adams and Nyhoff)

Methods by which student learning will be evaluated: The general guidelines for assessing grades are as follows:
- Examinations and Assignments, 65%
- Labs 10%
- Final Examination 25%

Academic Integrity: Academic honesty is taken extremely seriously and is expected of all students. All assignments must be the original work of the student (and partners or group, if applicable).

All questions or concerns regarding ethical conduct should be brought to the course instructor. “It is the official policy of the College that all acts or attempted acts that are violations of academic integrity be reported to the Office of Student Affairs (OSA). At the faculty member’s discretion and with the concurrence of the student or students involved, some cases, though reported to the OSA, may be resolved within the confines of the course and department.

The instructor has the authority to adjust the offender’s grades as deemed appropriate, including assigning an F to the assignment or exercise or, in more serious cases, an F to the student for the entire course.” (Taken from the QCC Academic Integrity Policy, 2/14/2005)

NOTE: Any student who feels may need an accommodation based upon a disability should contact the instructor privately to discuss his/her needs. Contact the office of Services for Students with Disabilities in the S-Building, room 132 (718 631 6257) for accommodations for students with documented disabilities.
CS 203 – Algorithm Problem Solving II

Textbook: C++ Without Fear by B. Overland, Prentice Hall, 2005 (and other texts mentioned in the syllabus).

6 Pointers:
1. C++ Address operator
2. C++ Pointer variables
3. Arrays and pointers
4. Pointer Arithmetic
5. Pointer Initialization
6. Pointer Comparison
7. Pointers as function parameters
8. Dynamic Memory Allocation

12 Constructors & Destructors:
1. Constructors
2. Default & Implicit
3. Constructor Overloading
4. Destructors
5. Constructor calling sequence
6. Copy Constructors

8 File I/O Storage:
1. Input / Output Streams
2. File Output Formatting
3. Records & Structures
4. Parameters to functions
5. Error Testing
6. Reading and Writing
7. Binary Files
8. Random Access Files

13 Operator Overloading:
1. Fundamentals
2. Operator overloading
3. Binary and Unary operators
4. Input and Output operators

16 Standard Template Library:
1. Templates – Introduction
2. Function Template
3. Fundamentals of STL
4. STL – Vectors
5. STL – Lists

9 Advanced Programming Techniques:
1. Command Line Arguments
2. Conditional Operator
3. Switch Statement
4. Enumerated Data types
5. Default arguments
6. Overloading functions
7. Dynamic Allocation I

17 Inheritance:
1. Fundamentals
2. Class access
3. Constructors, Destructors and Inheritance
4. Overriding Base Functions
5. Hierarchies

11 C++ Classes:
1. OOP Introduction
2. C++ Classes
3. Member functions
4. Private members
5. Protected members
6. Passing objects
7. Aggregation and Composition
8. Dynamic Allocation II

18 Polymorphism:
1. Virtual member functions
2. Abstract Base Classes
3. Composition versus Inheritance
4. Virtual Destructors

Topics to be covered as time allows:
- Pre-Processor Directives - Appendix D
- Creating Dynamic Link Libraries

Fall 2017
RM [Math Syllabus CS-203F17]