LAGUARDIA COMMUNITY COLLEGE
CITY UNIVERSITY OF NEW YORK
DEPARTMENT OF MATHEMATICS, ENGINEERING, AND COMPUTER SCIENCE

MAC125 – Advanced C++ Programming

4 hours (3h lecture, 1h lab), 3 credits
Prerequisite: MAC101 – Introduction to Computer Science

Catalog Description:
This course presents object-oriented algorithmic problem solving using C++. Topics include pointers and pointer arithmetic, linked lists, memory management, recursion, operator overloading, inheritance and polymorphism, stream and file I/O, exceptions and exception handling, templates and STL, applications of simple data structures and testing and debugging techniques.

Instructional Objectives:

1. Introduce students to the syntax of the C++ programming language.
2. Reinforce students' ability to solve problems using C++.
3. Familiarize students with classes.
4. Familiarize students with constructors and instantiation of objects.
5. Enable students to use objects and object-oriented concepts such as encapsulation, inheritance and polymorphism.
6. Introduce students to exception handling.
7. Familiarize students with the different aspects of the C++ programming language and programming techniques (e.g. user defined classes).

Performance Objectives:

1. Write syntactically correct C++ programs.
2. Solve problems using C++.
3. Write programs to implement classes.
4. Write constructions in classes.
5. Write programs that utilize objects and the concepts of encapsulation, inheritance and polymorphism.
6. Write programs with exception handling.
7. Compare and contrast the various programming techniques and features of the C++ programming language.

Required Textbook :
Walter Savitch, ***Absolute C++*, Fifth Edition,** Addison Wesley, 2013, ISBN13: 9780132830713

Some additional references:

www.cplusplus.com
Free electronic Book by Bruce Eckel (Thinking in C++)
http://www.ibiblio.org/pub/docs/books/eckel/

Grading Standards:

Quizzes	10%
Laboratory/HWs	20%
Project	15%
Midterm Exam	25%
Final Exam	30%
Total	100%

Note: Your labs will be graded according the following rubric:

1	2	3	4	5
The student has entered a code that does not solve the problem and may or may not run. The code has no relation at all to the solution of the problem.	The program has a sense of the solution but is lacking some key logic.	The structure of the program is good but there is one small logic or syntax error.	The program solves the problem but it is not documented and written concisely and lacks good programming style.	The program solves the problem correctly, is documented, and tested for all cases.

Academic Integrity:

This class will be conducted in compliance with LaGuardia Community College's academic integrity policy.

Attendance:

The maximum number of unexcused absences allowed is 15% of the total class meetings (about 7 hours). Unexcused absences beyond this maximum will result in a grade of WU or F.

Comments:

The grading standards listed above and the suggested homework problems listed in the course outline are both subject to modification by the instructor.

COURSE OUTLINE

Weeks	Topic	Chapter	Suggested Homework Assignments and Programming projects
Week 1	C++ Basics: data types, expressions, assignments, flow control. Strings, streams and file I/O.	1, 2, 9, 12	2.2, 2.8, 9.2 Project 9.6
	Function basics, parameters and overloading. Arrays and structures.	3-5, 6.1	3.6, 4.15, 5.9

Week 2	Introduction to classes, classes and functions, public and private members.	6.2	
	Constructors and destructors, static functions and variables.	7.1, 7.2	6.5, 6.7, 6.8, 7.6, 7.7
Week 3	Inline function definitions, constants. Constants as parameters.	7.2	
	Declare-Define-Use approach to writing classes, .h and .cpp files, stages of compilation, macros and preprocessor directives.	11.1	11.4
Week 4	Introduction to operator overloading.	8.1	8.2, 8.7
	Overloading operators as members and as non-members. Friend functions.	8.1, 8.2, 8.3	
Week 5	Pointers and references. Arrays and pointers.	10.1	10.3 Project 10.6
	Functions and pointers. Dynamic allocation.	10.2	
Week 6	Classes and dynamic allocation.	10.3	
	Exception handling, throwing an exception, try-catch blocks.	18	18.1, 18.2, 18.5
Week 7	Midterm Exam (Covers weeks 1-6)		
	Introduction to inheritance.	14.1	14.3, 14.4, 15.3
Week 8	Pointers and inheritance, casting, polymorphism, virtual functions, pure virtual functions, abstract classes.	14.2, 15.1	
	Inheritance and dynamic allocation, constructors, copy constructors, "=" overloads and inheritance, multiple inheritance.	15.1	
Week 9	Exceptions and inheritance, protected/private inheritance.	15.2	Project 14.9, 15.4
	Introduction to templates, function templates, class templates. Templates and inheritance.	16.1, 16.2	16.4, 16.6
Week 10	Introduction to Standard Template Library (STL): vector and basic_string. Selection sort. Linear search. Binary search.	16.3, 7.3	
	Recursions.	13	13.1, 13.8
Week 11	Linked lists.	17.1	
	More STL: iterators, containers, stacks, queues.	17.2,17.3,19.1,19.2	17.2, 19.3
Week 12	More STL.	17.2,17.3,19.1,19.2	
	Final Exam Review		
Week 13	Final Exam (Covers weeks 1-12)		