HOSTOS COMMYNITY COLLEGE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

COURSE: CSC 140 INTRODUCTION TO COMPUTER SCIENCE & PROGRAMMING

CREDIT HOURS: 3.0

EQUATED HOURS: 3.0

CLASS HOURS: 3.0

PRE/COREQUISITE: ENG 10/110 ALP AND MAT 150 (Coreq)

RECOMMENDED TEXT: C++ How to Program (10th Edition) By Paul Deitel & Harvey Deitel ISBN-13: 978-0-13-444823-7

REQUIRED SOFTWARE:
MyLab Programming with Pearson eText -- Access Code Card -- for C++ How to Program (Early Objects Version)

REFERENCE MATERIAL
C++ Reference:
http://en.cppreference.com/w/cpp
HomeWork:
MyLab Programming:
MyLab Programming with Pearson eText -- Access Code Card -- for C++ How to Program (Early Objects Version)

DESCRIPTION: Computer problem solving and programming in a high level language such as C++ are introduced. Algorithmic problem solving and basic programming techniques are emphasized. Problems are solved using methods such as top-down design and stepwise iterative refinement. Programming topics include basic data types, operators and expressions, control structures, functions, arrays and pointers. Students are introduced to a modern program development environment in the computer lab and are expected to complete several programming projects.

EXAMS /PROJECTS: 2 Exams, 2 projects and comprehensive final exam

GRADES: A, A-, B+, B, B-, C+, C, D, I, F.

GRADING

Homework	10%
Projects (2)	20%
Exams (2)	40%
Final Exam	30%

LEARNING OUTCOMES:

1. Explain the major logical units of a computer. Explain the difference between machine, assembly and high-level computer languages. Explain the process for computer code to become executable program.
2. Apply the appropriate syntax to design programs to solve simple and complex real world problems. Understand data storage, addresses, arrays and pointers in C++. Explain the difference between sequence, selection and repetition control structures and demonstrate the ability to use each to implement algorithms
3. Utilize operators to perform arithmetic calculations and assign values to variables and explain operator precedence.
4. Apply Boolean logical operators, in selection control structures to build simple and complex conditional statements.
5. Explain the rationale behind dividing program tasks into separate functions, the operation of the function call stack and the difference between passing arguments by value and by reference, and demonstrate the ability to construct and use functions.
6. Apply single and multi-dimensional arrays, within a function or passed between functions. Explain what a pointer is, how pointers relate to arrays, and apply the use of pointers and pointer operators

COURSE OUTLINE:

Week Topics

1 Week 1

1. Computer hardware and software
2. Data hierarchy
3. Machine, assembly and high-level languages
4. Object and Software Development Technology

2 Week 2

1. C++ development environment and debugging
2. Operating Systems, Internet
3. Basic output
4. Variable declarations and basic input

3 Week 3

1. Arithmetic operators and operator precedence
2. Equality and relational operators (AND, OR, XOR, NOT)
3. Describing algorithms
4. Control structures: sequence, selection and repetition

4 Week 4

1. if / if … else statements
2. Counter-controlled repetition vs. sentinel-controlled repetition
3. Nested control structures
4. Assignment operators and increment/decrement operators

5 Week 5

1. counter-controlled repetition, for loops
2. do … while loops
3. multiple selection with switch statements
4. break and continue statements

6 Week 6

1. Boolean logical operators (AND / OR / NOT)
2. Defining functions and function prototypes
3. C++ standard library libraries, including math
4. Argument coercion and promotion rules

7 Week 7

1. Exam 1

2. Random number generation
3. User-defined enumeration type
4. Scope rules

8 Week 8

1. Function Call Stack
2. Inline functions
3. Pass by value vs. pass by reference

9 Week 9

1. Project 1
2. Default arguments
3. Unary scope resolution operator

10 Week 10

1. Function overloading
2. Introduction to recursion
3. Declaring and using one-dimensional arrays
4. Searching and sorting arrays

11 Week 11

1. Multi-dimensional arrays
2. Pointer declaration and initialization
3. Address and dereference operators

12 Week 12

1. Using pointers with functions
2. Using const with pointers
3. Pointer expressions and arithmetic

13 Week 13

1. Pointers and arrays

2. Exam 2

14 Week 14

1. Review for Final
2. Project 2