HOSTOS COMMYNITY COLLEGE
DEPARTMENT OF MATHEMATICS

COURSE: CSC 275 OBJECT ORIENTED PROGRAMMING

CREDIT HOURS: 3.0

EQUATED HOURS: 3.0

CLASS HOURS: 3.0

PRE/COREQUISITE: ENG 93/ESL 91/ESL 93
PREREQUISITE: CSC 215 Modern Programming

RECOMMENDED TEXT: *C++ How to Program Late Objects Version*, by Harvey & Paul Deitel (7th Edition) Pearson:

DESCRIPTION: This course is a continuation of algorithmic problem solving designed to promote object-oriented programming concepts, techniques, and applications. It introduces more advanced methods, particularly object-oriented design. Topics include procedural abstraction, user defined static, dynamic and generic data types, linked structures, sorting and searching, event-driven programming and recursion. Abstract data types, inheritance and polymorphism are examined. Principles of rigorous programming practice and software development are emphasized.

EXAMINATIONS: A minimum of three quizzes (30%), midterm/project in computer laboratory (40%), and comprehensive final exam (30%).

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

Specific Student Learning Objectives within the C++ programming environment:
Students will demonstrate proficiency with:

1) Designing and implementing programs with a modern programming language such as C++
2) Modeling real life object with a software object
3) Pointers for functions, objects, arrays etc.
3) Classes of objects, functions and functions as members of classes, demonstrating ability to employ hierarchies of functions and objects within classes.

4) Overload operators for binary and unary, operators as member or non-member functions and conversion between types of operators.
5) Programming with Public, protected and private inheritance and the polymorphism of class-objects and functions.

General Student Programming Objectives

1) Describe the process by which high level process is deconstructed into a sequence of atomic logical steps and then transformed in a manner compatible with the syntactic rules of a programming language into an executable computer program. Think critically in linking theory, research and implementation.
2) Write assignments consisting of design, documented code, and description of testing \ methodology, critical contemplation of the difficulties faced, possible improvement
3) Accomplish a specific purpose, ethnically and legally (e.g., demonstrate critical Interpretation of required reading; and/or primary data gathering by observation and experimentation; and/or finding and evaluating internet resources)
4) Specify simple abstract data types and design implementations, using abstraction functions to document them.
5) Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
6) Name and apply some common object-oriented design patterns and give examples of their use

COURSE OUTLINE:

Week 1) Review of writing codes in a modern programming language such as C++: Structures-logical operators, functions, arrays etc.

Week 2&3) Pointers (Ch. 7) (7.1-7.12) Addresses and pointers. Pointer variable definitions and initialization, pointer operators, using pointers with constant qualifies and non-constant data, pointer expressions and pointer arithmetic, relationship between pointers and arrays, pointers to functions.

Week 4) Functions and an Introduction to Recursion (Ch. 5)(5.1-5.22) Returning values from functions. Reference arguments overloaded functions. Inline function. Default arguments. Returning by reference and recursive functions.

Week 5 (Ch. 9) (9.1-9.13) Introduction to class, objects, class variables, class methods. Defining a class with a member function, defining a member function with a parameter, placing a class with a separate file for reusability, validating data with set functions.

Week 6) Classes In depth (Ch. 10, 10.1-10.7) Constant objects and constant member functions, Composition-objects as members of classes, friend functions and classes, using the 'This' pointer, static class members, proxy classes, class hierarchy.

Week 7 &8) Operator Overload (Ch. 11, 11.1-11.15) Using overload operator and the standard library of functions, overloading binary operators, overloading stream insertion and extractions operators. Overloading Unary operators, Dynamic memory management, data and array classes, and operators as member functions versus non-member functions. Converting between types, explicit constructors, building a string class.

Week 9 & 10) Object Oriented Programming-Inheritance (Ch.12, 12.1-12.7) Basic and derived classes, protected members, relationship between base classes and derived classes, constructors and destructors in derived classes. Public, protected and private inheritance.

Week 11) Midterm Project

Week 12&13) Polymorphism (Overloaded Methods-Run time and Inheritance (Ch.13, 13.1-13.9) Relationships among objects in an inheritance hierarchy: Base-class functions from derived class objects, aiming derived class pointers at derived class objects, derived class member functions calls via base class pointers. Type fields and switch statements, abstract classes and pure virtual functions, demonstrating polymorphic processing. Polymorphism, Type Conversion, Casting, Etc.

Week 14) Review for final exam

(Optional as time allows) Templates, Function templates, overloading function templates, class templates, non-type and default type parameters.

Resources

Recommended free download-MIT OCW 6-096 Introduction to C++
http://dspace.mit.edu/handle/1721.1/74125

Recommended alternative textbook: C++ Primer C++ Primer (5th Edition) 5th Edition by Stanley B. Lippman, Josée Lajoie, Barbara E. Moo
https://www.amazon.com/Primer-5th-Stanley-B-Lippman/dp/0321714113

Simulation work with application software will be conducted in computer Lab facility during and outside of class time.