William Paterson University of New Jersey
College of Science and Health
Department of Computer Science

COURSE OUTLINE

I. Course Title: CS-230 Computer Science I 4 credits

II. Course Description:
This is the introductory course offered to all majors to gain fundamental knowledge of computer and computing and their applications. The topics include algorithmic approach to computer problem solving and programming methodology: Analysis, design, writing, compilation, execution, documentation, implementation, debugging, and evaluation of a computer program with procedural abstraction and basic data representation. Substantial programming assignments (in ANSI C/C++ language) is emphasized , including problem solving in numerical, applied to mathematical, science, business and other area as well as non-numerical applications. .

III. Prerequisite:
High School Algebra Proficiency Equivalent or MATH-115, or MATH-135

IV. Course Objectives:
1. To learn algorithmic problem solving methodologies.
2. To learn structured programming techniques and development.
3. To learn the basic features of the C/C++ programming language.
4. To become familiar with a program development environment and to acquire computer problem solving and programming skills through extensive computer project assignments, applied to mathematics, various scientific fields, and business as well as other numerical and non-numerical areas.
5. To become familiar with evolution of computer hardware, software and systems including computer ethics, security as well as social impact of computer and information technology

V. Student Learning Outcomes (for TI, Course Specific and UCC general)
After the completion of this course, a successful student will be able to do the following:
1. To meet Technology Intensive Outcomes students will be able to:
   - T 1. Demonstrate a sound understanding of technology concepts, systems integration (Software, hardware and systems) and operations. This is the foundation of Computer Science program which implanted the seed for the current magnitude of computer technology and application in all fields.
   - T 2. Use a variety of technologies, hardware technology, input, output, operating systems, programming language (including compilation, debugging, linkage and execution process, etc) to access, evaluate, collect and manage data, information and database.

T3. Understand the impact of technology. CS Students will gain proficiency in foundation of the computer science studies to prepare them to advance to higher level of computer science. Non-CS majors will acquire the basic skills to create their own computer programs which can be applied for problem solving in their respective field of studies.

T 4. Practice legal and ethical behaviors in the context of technology, Students are knowledgeable in and practice:

a. University Student Code of Conduct,
b. Computer Project Policy (departmental)
c. ACM Code of Ethic and Professional Conduct
d. IEEE Code of Ethics

The above consist of extensive directives regarding legal and ethical behaviors in the context of technology

2. Course Specific SLOS
   a. Describe the evolution of Computer hardware, Software and system, including security, privacy, and its impact on society
   b. Describe major components of a digital computer system with their functions.
   c. Distinguish different categories of software
   d. Explain the difference between high-level languages and low-level languages
   e. Describe the concept of a variable, and evaluate arithmetic expressions.
   f. Provide an algorithmic solution to a problem that requires simple selection, simple repetitive execution, or simple conditional execution.
   g. Provide an algorithmic solution to a problem that requires multi-way selection, nested selections, and nested repetitive execution, and a combination of these behaviors.
   h. Specify simple selections, multi-way selections, nested selections, simple repetitive executions, nested repetitive executions and conditional executions in a high-level language.
   i. Trace the execution of a program segment with simple selections, multi-way selections, nested selections, simple repetitive executions, nested repetitive executions and conditional executions.
   j. Write, document, execute, debug and test a program in every program development environment.
   k. Write programs that manipulate a one-dimensional and a two-dimensional array, and also explain and the concept and use of a pointer variable.
   l. Write, declare, and call functions that return a value or not, and with value and/or reference parameters
   m. Write programs with four or more functions.
   n. Write program for problem derived from mathematics, Science, business and other areas of computer applications
3. The course will also reinforce the following students learning outcomes of the university: General UCC SLOS:
   a. Effectively express themselves in written and oral form. Students will be

expected to complete written assignments on programming and describe the behavior of programs in exams and home works. Measure: homework and exam questions. (Program SLO 1)

b. Demonstrate ability to think critically. Critical thinking is required in building models for addressing programming problems, constructing programs, choosing the correct procedural statements/models, verifying their correctness (via the most exacting standard, debugging), and judging them in terms of efficiency. Measure: exams, surveys, and projects. Locate and use information. (Program SLOS 4 and 8)

c. Students will be expected to find formulas, equations, and facts on the Internet to support the building of those models. Measure: project-work requiring the Internet. On less frequent occasions, the library will be used as a resource for information needed in programming. (Program SLOS 2, 4, 5, and 8)

d. Demonstrate ability to integrate knowledge and idea in a coherent and meaningful manner. Programming requires the effective merging of application-specific knowledge, procedural representations in C/C++, algebraic formulas, and other discipline specific information. Coherence and meaning are judged by the strictest standard, that of the compiler (linguistically) and execution (for precise exact goals). Measure: programming projects, exams, in class programming writing practice effort and in class presentation of homework level. (Program SLOS 4, 5, and 8)

VI. Course Contents:

1 Introduction to Computers and Programming
   a. Evolution of computers and systems, including security, privacy and impact on Society
   b. Computer Organization and program execution
   c. Computer problem solving steps and the algorithm
   d. Principle of structured programming: top-down modular design
   e. Introduction to C++ language
   f. Lab procedures and ethics, and project policy, including project originality
2. Data Types, Declarations, and Simple Input/Output
   a. Elementary data types: integer, floating point, and character type
   b. Variables: declarations, values, addresses
   c. Arithmetic and assignment operators
   d. cin and cout for I/O.
3. The Selection Structure
   a. Relational and logical operators
   b. If-else statement: simple and nested
   c. Compound statement
   d. Switch statement
4. The Repetition Structure
   a. While statement
   b. For statement

   c. Do-while statement
   d. Nested loops
5. Structured Program Development
   a. Top-down program development
   b. Modularity, top-down design, and step-wise refinement
   c. Structure diagrams
6. Function in ANSI C++
   a. Function definitions and function calls
   b. Function prototypes
   c. Library functions and code reuse
   d. Variable scope, life time, and type
7. Arrays, Addresses, and Pointers
   a. Array declaration and initialization
   b. Arrays and loops
   c. Passing individual array elements to function
   d. Passing entire array to functions
   e. Call-by-reference versus call-by-value
   f. Pointer variables
   g. Array names versus pointer variables
   i. Higher dimensional arrays

VII. Teaching Methods:

1. Classroom lectures, discussions, and problem-solving sessions.
2. Homework and pre-exam reviews.
3. Extensive program practice sessions

VIII. Evaluation:

How the SLO will be evaluated:

1. Student knowledge will be assessed by periodic exams, quizzes, lab projects, homework, and both written and oral presentations. There will be additional assessments of their computer programming, compilation, trouble-shooting, and debugging abilities during the execution cycle, and their ability to interpret their results.
2. Students will be evaluated by the by their ability o to carry out problem-solving for various application problems in science-related subjects. For example, use of gravitational acceleration, speed of light, temperature system conversion, one and two dimensional linear systems analysis and other application to investigate the objective universe.
3. Students will be evaluated by the degree of their achievement in data collection (input structure), data organization (using scalar reference, array and pointer variables) to construct algorithm and programs, as well as their ability to trouble-shoot during compilation and execution of C/C++ code.
4. Students will be evaluated by their ability to solve problems related to science using algebra, trigonometry, analytic geometry, probability, and statistics. For

example, students use a random number generator to analyze political election results.

For Technology Intensive Course:

T1. Students will be evaluated by their performance and knowledge acquired base on the periodic class room examinations, quizzes, homework assignments effort on lab practice projects (collectively) and individual projects.

T2. Students will be evaluated these skills by presentation of homework solution orally in class and in writing as well as the actual design and construction of algorithms which lead to computer programs applied to mathematics, and scientific related areas. For example, program design and analysis involving solution to problem related to of gravitational acceleration, conversion of temperature system, which are of skills necessary to investigate the objective universe.

T3. Students will be evaluated by their ability to collect, organize, and to evaluate scientific information by identifying needed input data for the program, arrange the data in the form of integer, floating point and/or characters, arrays and pointers etc., then interpret the degree of success by the outcome obtained for the solved problems.

T4. Students will be evaluated in their quantitative skill by demonstrating ability to use appropriate mathematical tools to solve problems related to mathematic, engineering, science, business and other areas of applications. In particular, extensive algebra and trigonometry, some analytic geometry, probability and statistics are employed for program constructs in C/C++ code .

IX. Textbook (Currently in use):

"First Book of C++: From Here to There", 3rd Ed., by Gary Bronson, Course Technology /Thomson Company, 2006.

"Starting Out with C++", Tony Gaddis, Judy Walters, Godfrey Muganda. 6th Edition, Addison Wesley, 2008

X. Bibliography:

1. C++: How to Program (Fifth Edition), C, H.M. Deitel and P.J. Deitel, Prentice Hall 2006
2. C by Dissection, Al Kelley and Ira Pohl, Benjamin Cummings 1992
3. Learning C++, Neill Graham, McGraw Hill 1991
4. C++ The Computer Wizard Series, John Mason, Prentice Hall, 2006
5. C++ Programming: From Problem Analysis to Program Design, Second Edition, D. Malik, Prentice Hall, 2004.
6. Computer Science: A Structured Approach Using C++, Second Edition, Behrouz Forouzan, Prentice Hall, 2003.
7. Programming with C++, Concept and Applications, Gibert Ndjatou, Thomson CP., 2003
8. "Absolute C++" Walter Savitch, 3rd Edition, Addson Wesley, 2008

XI Prepared by Dr. Li-hsiang (Aria) S. Cheo, Nov. 2010

XII. Original Author and Department Approval Date: Dr. L. Cheo, Spring 1973.

XIII. Revised by: Dr. L. Cheo, 1979 and 1981
Drs. .L. Cheo, J. Najarina and Prof. Jololian, 1986
Dr.s L. Cheo and J. Najarian, 1992
Drs. E. Hu and .Ndjatou 1997
Dr. G. Ndjatou on April, 2000.

XIV. Department Current Revision Approval Date: November, 2010.