College of San Mateo
Course Outline

☐ New Course
☐ Update/No change
X Course Revision (Minor)
☐ Course Revision (Major)

Date: April 21, 2010

Department: **CIS** **Number:** **278**

Course Title: **Programming Methods: C++** **Units: 4**

Total Semester Hours **Lecture:** **48** **Lab:** 48 **Homework:** 96 **By Arrangement: 0**

Length of Course

X Semester-long
☐ Short course (Number of weeks___)
☐ Open entry/Open exit

Grading

☐ Letter
☐ Pass/No Pass
X Grade Option (letter or Pass/No Pass)

Faculty Load Credit (To be completed by Division Office; show calculations.)**:**
(3.0 X 16)/16 + (3.0 X 16)/16 X 0.7 = 5.1 FLC

1. **Prerequisite** (Attach Enrollment Limitation Validation Form.)
Math 120 or equivalent and CIS 254 or equivalent

2. **Corequisite** (Attach Enrollment Limitation Validation Form.)
N/A

3. **Recommended Preparation** (Attach Enrollment Validation Form.)
Eligibility for English 838/848

4. **Catalog Description** (Include prerequisites/corequisites/recommended preparation. For format, please see model course outline.) **CIS 278 (CS1) Programming Methods: C++** (4)
(Pass/No Pass or letter grade option.) Minimum of 48 lecture and 48 lab hours per term. Prerequisite: MATH 120 or equivalent; CIS 254 or equivalent. Recommended Preparation: eligibility for ENGL 838/848.. Continuation of CIS 254. Object-oriented programming methodology for both computer science majors and computer professionals. Systematic approach to design, construction, and management of computer programs; emphasizing program documentation, testing, debugging, maintenance and software reuse. Also includes UML, virtual machines, exception handling, sorting and searching algorithms, recursion, fundamental graphics, and computer ethics. This course conforms to the ACM CS1 standards. A materials fee in the amount shown in the *Schedule of Classes* is payable upon registration. (AA: Area C1/Area E2c, CSU: Area B4, UC)

5. **Class Schedule Description** (Include prerequisites/corequisites/recommended preparation. For format, please see model course outline.)
CIS 278 (CS1) PROGRAMMING METHODS: C++ Continuation of CIS 254. Object-oriented programming methodology for both computer science majors and computer professionals. Systematic approach to design, construction, and management of computer programs; emphasizing program documentation, testing, debugging, maintenance and software reuse. Also includes UML, virtual machines, exception handling, sorting and searching algorithms, recursion, fundamental graphics, and computer ethics. This course conforms to the ACM CS1 standards. A $____ materials fee is

payable upon registration. ***Prerequisite***: MATH 120 or equivalent; CIS 254 or equivalent. ***Recommended Preparation***: eligibility for ENGL 838/848. Pass/No Pass or letter grade option. (AA: Area C1/Area E2c, CSU: Area B4, UC)

6. Student Learning Outcomes (Identify 1-6 expected learner outcomes using active verbs.)

Upon successful completion of the course, the student will be able to:

- Demonstrate knowledge and understanding of the principal object-oriented programming concepts.
- Employ Unified Modeling Language (UML) notation to model the object-oriented design of a non-trivial computer program.
- Implement a medium-size computer program that is stylistically and functionally correct, based on an object-oriented design model.
- Reuse existing components through inheritance and polymorphism.
- Implement, test, and debug simple recursive functions.
- Understand and employ basic sorting and searching algorithms.
- Create dynamically allocated variables.
- Employ components in the C++ Standard Template Library (STL).

7. Course Objectives (Identify specific teaching objectives detailing course content and activities. *For some courses, the course objectives will be the same as the student learning outcomes. In this case, "Same as Student Learning Outcomes" is appropriate here.)*

Same as above

8. Course Content (Brief but complete topical outline of the course that includes major subject areas [1-2 pages]. Should reflect all course objectives listed above. In addition, a sample course syllabus with timeline may be attached.)

I. C++ Basics
   A. I/O Streams
   B. Control Structures
   C. Algorithm development; design patterns
   D. Structured programming methodology

II. Functions
   A. Predefined functions
   B. STL template libraries
   C. User defined functions
      1. Call by reference/value parameter passing

2. Function overloads and default arguments

III. Fundamental Data Structures
   A. Array declaration and manipulation
   B. Partially filled arrays
   C. C-strings
   D. Dynamically allocated arrays
   E. String class
   F. Simple text files

IV. Object Oriented Programming
   A. Review of object oriented methodology and design
      1. UML class diagrams
   B. Class Development
      1. Object construction
      2. Use of const keyword
      3. Static class members and functions
      4. Operator overloading
      5. Copy constructors, destructors, assignment overload
   C. Derived Classes
      1. Polymorphic behavior
      2. Abstract classes

V. Fundamental Computing Algorithms
   A. Sorting
   B. Searching
   C. Recursive algorithms

VI. Exception handling

VII. Foundations of Human-Computer Interaction
   A. Human-centered development and evaluation
   B. Principles of good designs and good designers
   C. Engineering tradeoffs
   D. Introduction to usability testing

VIII. Software Engineering Issues
   A. Tools
   B. Processes
   C. Requirements and specifications
   D. Design and testing
   E. Design for reuse
   F. Risks and Liabilities of computer-based systems

IX. Language Translation
   A. Comparison of interpreters and compilers
   B. Portability

9. Representative Instructional Methods (Describe instructor-initiated teaching strategies that will assist students in meeting course objectives. Describe out-of-class assignments, required reading and writing assignments, and methods for teaching critical thinking skills. **If hours by arrangement are required, please indicate the additional instructional activity which will be provided during these hours, where the activity will take place, and how the activity will be supervised.**)

The course includes the following instructional methods as determined appropriate by the professor:

Lectures, to introduce new topics;
"Models" for problem-solving techniques;
Class (group) problem solving , each person contributing a potential "next step";
Student participation in short in-class projects;
Q/A sessions with students providing both the questions AND the answers;
Students working in small groups to solve significant programming assignments.
Live code development/debugging demonstrations

10. Representative Methods of Evaluation (Describe measurement of student progress toward course objectives. Courses with required writing component and/or problem-solving emphasis must reflect critical thinking component. If skills class, then applied skills.)

Assessment of student contributions during class discussion and project time;
Individual programming assignments ;
Midterm and Final exams (short answer (textbook material), general problem solving (similar to in-class work), short program segments (similar to programming assignments) ;
Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort.

11. Representative Text Materials (With few exceptions, texts need to be current. Include publication dates.)

Savitch, Walter, *Absolute C++* 4th ed., Addison-Wesley, San Francisco, 2010.

Deitel & Deitel, *C++: How To Program* , 7th ed., Prentice Hall, New Jersey, 2010.

Prepared by: _____
(Signature)

Email address: grassos@smccd.edu

Submission Date: 4/16/2010