El Camino College

COURSE OUTLINE OF RECORD - Official

I. GENERAL COURSE INFORMATION

Subject and Number: Computer Science 30
Descriptive Title: Advanced Programming in C++

Course Disciplines: Computer Science

Division: Mathematical Sciences

Catalog Description: This course presents an advanced coverage of the C++ programming language. Topics include templates, the Standard Template Library, data abstraction, operator overloading, inheritance, friend functions, virtual functions, multiple inheritance, and virtual base classes. An emphasis will be placed on object-oriented programming.

Conditions of Enrollment: **Prerequisite**

Computer Science 2
with a minimum grade of C
or

equivalent

Course Length: [X] Full Term [ ] Other (Specify number of weeks):
Hours Lecture: 3.00 hours per week [ ] TBA
Hours Laboratory: 3.00 hours per week [ ] TBA
Course Units: 4.00

Grading Method: Letter
Credit Status Associate Degree Credit

Transfer CSU: [X] Effective Date: Prior to July 1992
Transfer UC: [X] Effective Date: Spring 1994

General Education:
El Camino College: _____
CSU GE: _____
IGETC: _____

II. OUTCOMES AND OBJECTIVES

A. COURSE STUDENT LEARNING OUTCOMES (The course student learning outcomes are listed below, along with a representative assessment method for

each. Student learning outcomes are not subject to review, revision or approval by the College Curriculum Committee)

1. Students will design, code, compile, test and document programming solutions to problems requiring the development of C++ classes (by inheritance, by composition; templates), requiring C++ operator overloading, requiring effective use of the Standard Template Library, requiring effective use of pointers and dynamic memory allocation.
2. Students, when given a code segment involving use of a class, will be able to trace the construction of class objects, trace the destruction of class objects, verify whether memory leaks have occurred, trace object assignment operations, verify when copy constructors are invoked and when overloading of copy constructors is required.
3. Students, when given C++ code with errors, will be able to identify what those errors are and will be able to modify the C++ code to eliminate those errors.
4. Students will be able to explain the concept of C++ class templates and how they relate to the concept of generics, the concept of virtual functions and polymorphism, the concept of multiple inheritance and virtual base classes, the concept of container types and the circumstances where specific containers should or should not be used.

The above SLOs were the most recent available SLOs at the time of course review. For the most current SLO statements, visit the El Camino College SLO webpage at http://www.elcamino.edu/academics/slo/.

B. Course Student Learning Objectives (The major learning objective for students enrolled in this course are listed below, along with a representative assessment method for each)

1. Implement data abstraction and inheritance.
   Performance exams
2. Implement function and operator overloading.
   Laboratory reports
3. Use the Standard Template Library.
   Quizzes
4. Create and use friend and virtual functions.
   Objective Exams
5. Create new classes by inheritance and composition.
   Laboratory reports
6. Use multiple inheritance.
   Laboratory reports
7. Implement virtual base classes.
   Objective Exams
8. Describe how C++ implements virtual functions.
   Objective Exams

III. OUTLINE OF SUBJECT MATTER (Topics are detailed enough to enable a qualified instructor to determine the major areas that should be covered as well as ensure

consistency from instructor to instructor and semester to semester.)

Lecture or Lab	Approximate Hours	Topic Number	Major Topic
Lecture	5	I	Pointers in C++ A. Declare pointers 1. Pointer Types 2. Manipulation pointers B. Using pointers 1. Accessing variables 2. Accessing arrays 3. Const and static types 4. Passing to functions 5. Dynamic Memory Allocation
Lecture	14	II	Classes A. Inheritance 1. Base/parent class 2. Derived classes B. Modes of visibility C. Virtual functions D. Friend functions
Lecture	5	III	Overloading A. Function B. Operator
Lecture	5	IV	Templates A. Template Class B. Abstract data types
Lecture	14	V	The Standard Template Library A. Containers 1. Deque 2. List 3. Vector 4. Map B. Algorithms 1. Initialization 2. Sorting 3. Searching 4. Transforming C. Iterators
Lecture	4	VI	Advance Inheritance A. Multiple inheritance B. Virtual base classes
Lecture	7	VII	Object Oriented programming A. Designing classes B. Hierarchies
Lab	5	VIII	Pointers in C++

			A. Declare pointers 1. Pointer Types 2. Manipulation pointers B. Using pointers 1. Accessing variables 2. Accessing arrays 3. Const and static types 4. Passing to functions 5. Dynamic Memory Allocation
Lab	14	IX	A. Classes 1. Inheritance 2. Base/parent class B. Derived classes 1. Modes of visibility C. Virtual functions D. Friend functions
Lab	5	X	Overloading A. Function B. Operator
Lab	5	XI	Templates A. Template Class B. Abstract data types
Lab	14	XII	The Standard Template Library A. Containers 1. Deque 2. List 3. Vector 4. Map B. Algorithms 1. Initialization 2. Sorting 3. Searching 4. Transforming C. Iterators
Lab	4	XIII	Advanced Inheritance A. Multiple inheritance B. Virtual base classes
Lab	7	XIV	Object Oriented programming A. Designing classes B. Hierarchies
	Total Lecture Hours	54	
	Total Laboratory Hours	54	
	Total Hours	108	

IV. PRIMARY METHOD OF EVALUATION AND SAMPLE ASSIGNMENTS

A. PRIMARY METHOD OF EVALUATION:

Problem solving demonstrations (computational or non-computational)

B. TYPICAL ASSIGNMENT USING PRIMARY METHOD OF EVALUATION:

Create a class, MyVector, to model vectors. The private data of the class should include a pointer to a double (storage for the elements of the vector) and an integer (the dimension of the vector). Remember that for classes with dynamic memory allocation, you will need to implement The Big Three. You are to overload the I/O operators, operators + and -, and overload the operator * to model the vector dot product. You are to write a test program to make sure your MyVector class does what it is supposed to do.

C. COLLEGE-LEVEL CRITICAL THINKING ASSIGNMENTS:

1. Develop a class, MyMatrix, to model matrices. The initial implementation of the class will utilize a double pointer to a double data type to store the elements of the matrix, and integer values to hold the number of rows and columns. Since you will have dynamic memory allocation, The Big Three must be implemented. You will overload the I/O operators, operators +, -, and * (matrix multiplication). You will implement a public member function Determinant to return the determinant of the matrix if it is a square matrix. You are to use function recursion in the definition of the Determinant function. Finally, you are to write a test program to verify that your MyMatrix class is functioning in a manner consistent with a matrix object, and with the good programming practices you have been learning.
2. Modeling the work we have been studying in the Vehicle inheritance class hierarchy, you are to create a Person inheritance class hierarchy. The classes Student, Voter, and Faculty will inherit directly from Person. The class GradStudent will inherit from Student. The class StudentVoter will inherit from both the Student class and the Voter class. You will be making use of at least one virtual base class, virtual functions, virtual inheritance, and multiple inheritance. Again, you are to write test code to verify that the classes of your hierarchy are working properly. In particular, you will verify that pointers to the base class Person may be used to properly create, dynamically, objects of any class in your hierarchy.

D. **OTHER TYPICAL ASSESSMENT AND EVALUATION METHODS:**

Other exams

Quizzes

Laboratory reports

V. INSTRUCTIONAL METHODS

Laboratory

Lecture

Note: In compliance with Board Policies 1600 and 3410, Title 5 California Code of Regulations, the Rehabilitation Act of 1973, and Sections 504 and 508 of the Americans with Disabilities Act, instruction delivery shall provide access, full inclusion, and effective communication for students with disabilities.

VI. WORK OUTSIDE OF CLASS

Study

Required reading

Problem solving activities

Estimated Independent Study Hours per Week: 6

VII. TEXTS AND MATERIALS

A. **UP-TO-DATE REPRESENTATIVE TEXTBOOKS**
Stanley B. Lippman, Josee Lajoie, Barbara E. Moo. C++ Primer. 5th ed. Addison Wesley, 2013.

B. **ALTERNATIVE TEXTBOOKS**

C. **REQUIRED SUPPLEMENTARY READINGS**

D. **OTHER REQUIRED MATERIALS**

VIII. CONDITIONS OF ENROLLMENT

A. Requisites (Course and Non-Course Prerequisites and Corequisites)

Requisites	Category and Justification
Course Prerequisite Computer Science-2 or	Sequential
Non-Course Prerequisite	The student will be exempted from the prerequisite if she/he can demonstrate sufficient object-oriented programming knowledge in a language (such as C/C++, C#, Java, PHP,...) through work portfolio or oral and/or written examination by the department of computer science faculty.

B. Requisite Skills

Requisite Skills

1. Write C++ code using programmer-defined classes. CSCI 2 - Explain and implement basic data structure techniques: pointers, classes, recursion, searching, sorting, templates and dynamic memory allocation.
2. Write C++ code using function pointers. CSCI 2 - Explain and implement basic data structure techniques: pointers, classes, recursion, searching, sorting, templates and dynamic memory allocation.
3. Write C++ code involving dynamic memory allocation and de-allocation. CSCI 2 - Explain and implement basic data structure techniques: pointers, classes, recursion, searching, sorting, templates and dynamic memory allocation.

C. Recommended Preparations (Course and Non-Course)

Recommended Preparation	Category and Justification

D. Recommended Skills

Recommended Skills

E. Enrollment Limitations

Enrollment Limitations and Category	Enrollment Limitations Impact

Course created by Joseph Hyman on 10/05/1990.

BOARD APPROVAL DATE: 12/09/1991

LAST BOARD APPROVAL DATE: 11/21/2016

Last Reviewed and/or Revised by Ralph Taylor on 03/28/2016

19378