Course Outline of Record

Number: CS G179  Title: C++ Programming, Advanced

Originator: Donavan Nielsen  Eff Term: Fall 2012
Formerly Known As:

Cross Listed Course:

Semester Units: 4.0
Hrs Lec: 54.0  Hrs Lab: 54.0  Hrs Other: 0.0
Contact Hrs Total: 108.0
Study Non-Contact Hrs Recommended: 108.0

Catalog Description:
This course covers advanced features of software development using the C++ language. Topics covered will include input/output streams, file input and output, exception handling, Standard Template Library (STL) including string class, sequential and associative containers. Understanding function objects, STL algorithms, adaptive containers, bitset class and smart pointers will also be discussed in lectures and practiced through lab projects. Advisory: Experience with Object Oriented Programming in C++, including templates.

Justification for Course:

Prerequisites:

Corequisites:

Advisories:
- Experience with Object Oriented Programming in C++, including templates.

Assigned Disciplines:
Computer science

Material Fee: Yes [ ] No [X] Amount: $3.00
Credit Status: Noncredit [ ] Credit - Degree Applicable [X] Credit - Not Degree Applicable [ ]
Open Entry/Open Exit: Yes [ ] No [X]
Transfer Status: CSU Transferable [ ] UC/CSU Transferable [X] Not Transferable [ ]
Basic Skills Status: Yes [ ] No [X] Levels Below Transfer: Not Applicable
California Classification Codes: Y - Not Applicable
Non Credit Course Category: Y - Not applicable, Credit Course
Occupational (SAM) Code: D
Repeatable According to State Guidelines: No [X] Yes [ ] Number Repeats:
Required for Degree or Certificate: No [ ] Yes [X]
AA in Liberal Arts with an Emphasis in Computer Science and Technology

GE and Transfer Requirements Met:

Course Level Student Learning Outcome(s) Supported by this course:

Course Objectives:
1. Discuss advanced Object Oriented Programming concepts in C++.
2. Create and use modules including containers from the Standard Template Library (STL).
3. Discuss and use iterators and algorithms to find, remove, reverse, and edit elements in containers.
4. Create and implement a method to work with bit flags to organize and manipulate bitwise information.
5. Implement exception handling.
6. Use software development methodologies and debugging techniques.

**COURSE CONTENT:**

**LECTURE CONTENT:**

A. Input / Output Streams.
B. File Input / Output.
C. Exception handling.
D. Intro to the Standard Template Library (STL).
E. Iterators and Algorithms.
F. STL String Class.
G. STL Dynamic Array classes (vector and deque classes).
H. Doubly linked list (STL list class).
I. Efficient searches using STL associative containers (set, multiset, map and multimap).
J. Supplying Custom Sort Predicates (unary and binary).
K. Concept and usage function objects.
L. Adaptive Containers (STL stack, queue and priority_queue classes).
M. Organize and manipulate bitwise information using STL bitset and vector<bool> classes.
N. Smart Pointers.

**LABORATORY CONTENT:**

A. Weekly assignments will be given such as: given a set of requirements for a small business or scientific problem, need code using the various classes being studied in their solution.
B. Create a project with the correct file structure.
C. Break the programs into appropriate classes.
D. Design a simple user interface to satisfy the user interactions.
E. Code all the necessary expressions, branches, loops, functions, and classes.
F. Add the appropriate error handling routines.

**METHODS OF INSTRUCTION:**

A. Lecture:
B. Lab:
C. Online:
D. Independent Study:

**INSTRUCTIONAL TECHNIQUES:**

**COURSE ASSIGNMENTS:**

Out-of-class Assignments

Reading Assignments

Required reading:

Textbook
Websites
Optional library research paper promoting study in current advanced programming

Writing Assignments

Weekly lab assignments will be posted on the class web site. Lab solutions shall be submitted in person by the student in the lab during lab hours. Students will be required to write documentation under the form of comments for their projects. Detailed explanations may be requested by the
METHODS OF STUDENT EVALUATION:
Midterm Exam
Final Exam
Short Quizzes
Written Assignments
Objective Examinations
Projects (ind/group)
Problem Solving Exercises
Skills Demonstration

Demonstration of Critical Thinking:
Students will be demonstrating their laboratory projects. Optional research in current advanced programming or other related topics selected by the student and approved by the instructor.

Required Writing, Problem Solving, Skills Demonstration:
Weekly lab assignments will be posted on the class web site. Lab solutions shall be submitted in person by the student in the lab during lab hours. Students will be required to write documentation under the form of comments for their projects. Detailed explanations may be requested by the instructor.

TEXTS, READINGS, AND RESOURCES:

TextBooks:

LIBRARY:

Adequate library resources include:

Comments:

Attachments:

[Attached Files]