MENDOCINO COLLEGE
COURSE OUTLINE
APPROVED COURSE CSC 222

CATALOG INFORMATION
==========
Dept & Nbr: CSC 222 Title: PROGRAM & ALGORITHMS II
Full Title: Programming and Algorithms II

Units		Course Hours Per Week		Nbr of Weeks	Course Hours	Total
Max:	3.0	Lecture	2.5	17	Lecture	42.5
Min:	3.0	Lab	1.5		Lab	25.5
		Contact DHR	0.0		Contact DHR	0.0
		Contact Total	4.0		Contact Total	68.0
		Non-contact DHR	0.0		Non-contact DHR	0.0

Title 5 Category: 01 AA Degree Applic
Grading: GC Credit course for grade or CR/NC
Repeatability: 00 No repeatability allowed or defined
Also listed as:

CATALOG DESCRIPTION:
A continuation of the study of software specifications, design, and implementation. Introduction to abstract data types, objects, classes, fundamental data structures, and associated algorithms. Coverage includes dynamic memory, file I/O, linked lists, stacks, queues, binary trees, templates, inheritance, and recursion. Learn to design, implement, test, and analyze programs in an object-oriented language.

PREREQUISITES:
Completion of CSC 221.

RECOMMENDED PREPARATION:
Completion of CSC 201.

LIMITS ON ENROLLMENT:

SCHEDULE OF CLASSES INFORMATION:
Prerequisites: Completion of CSC 221.
Recommended: Completion of CSC 201.
A continuation of the study of software specifications, design, and implementation. Introduction to abstract data types, objects, classes, fundamental data structures, and associated algorithms. Coverage includes dynamic memory, file I/O, linked lists, stacks, queues, binary trees, templates, inheritance, and recursion. Learn to design, implement, test, analyze programs in an object-oriented language. (Grade or CR/NC)
Transfer Credit: CSU; UC.

ARTICULATION and CERTIFICATE INFORMATION
==========
ASSOCIATE DEGREE: Effective: FALL 2004 Inactive:
Area: D2 COMMUNICATIONS & ANALYTICAL THINKING

CSU GE: Effective: Inactive:
Transfer area:

IGETC: Effective: Inactive:
Transfer area:

CSU TRANSFER: TRANSFERABLE Effective: FALL 2003 Inactive:

UC TRANSFER: TRANSFERABLE Effective: FALL 2007 Inactive:

CAN:

CERTIFICATE APPLICABLE: N NOT CERTIFICATE/MAJOR APPLICABLE

APPROVAL AND DATES
==========
Version 01 Submitted by: CHARLES POTEET Date: 01/30/2003
Department approved: Date:
Curriculum approved: 02/19/2003 Version approved: 02/19/2003
Prerequisites approved: 02/19/2003 Last reviewed: 02/19/2003
Term effective: FALL 2003 Last taught: SPRING 2008 Inactive:

COURSE CONTENT
==========
OUTCOME AND OBJECTIVES:
1. Learn and adopt a disciplined approach to algorithm design and program implementation, fundamental areas of structured problem solving, object-oriented programming, data abstraction, and comparative analysis of algorithms
2. Understand the basic principles and techniques in specification, design, implementation, and application of basic data types to real-world problems
3. Usage of basic tools to develop correct, efficient, and well-structured programs. Build a foundation for further studies in computing sciences.
4. Understand and use object-oriented concepts, terminology, and techniques in conversation and problem-solving.
5. Create procedural and object-oriented algorithms, designs, and specifications for software to achieve a desired result.
6. Use procedural and object-oriented techniques in the C++ language to write programs that fulfill a specification contract.
7. Select the correct data structure for a given application design or implementation, and understand the differences between them.
8. Create and use techniques such as overloading, templates, inheritance, and standard libraries to achieve code reusability
9. Create C++ programs of moderate complexity using object-oriented and procedural techniques.
10. Proficiently use C++ programming tools and debugging techniques.
11. Successfully move on to the study of advanced data structures and their use in complex software.

TOPICS AND SCOPE:
1. O-O Terminology

2. Standard Template Library (STL)
3. Make files
4. Data Abstraction
5. Preprocessor Directives
6. Classes & Structs
7. Member Functions
8. Access Modifiers
9. Arrays & Strings
10. Stacks
11. Searching
12. Paramenter Passing
13. Templates
14. Big-O notation
15. Recursion
16. Effective Documentation
17. Conventions
18. Linking
19. Unix Environment
20. Basic Data Structures
21. Consturctors & Destructors
22. Header files & Libraries
23. Linked Lists
24. Queues
25. Sorting
26. Stream I/O
27. Inheritance
28. Overloading

ASSIGNMENTS:

READING ASSIGNMENTS:
Assigned readings including chapters from the text and may include articles from computer magazines and/or journals, software tutorials, and related web-based materials.

WRITING ASSIGNMENTS:
Write, debug, test, and document several computer applications or applets using an object-oriented language and development environment.

OUTSIDE ASSIGNMENTS:
1. Reading assigned chapters or other materials
2. Completion of assigned exercises and problem sets
3. The development, implementation, and troubleshooting of algorithms and computer programs that provide solutions to programming problems

METHOD OF INSTRUCTION:

METHODS OF EVALUATION:

Computational or non-computational problem-solving demonstrations including: development and refinement of algorithms for creating computer programs; development and troubleshooting of programs using an object- oriented language such as C++ or Java. Examinations including quizzes, midterm, and final. Students may be required to create a portfolio of their work and

periodically submit it for evaluation.

BASIS FOR GRADING:

The assignment of a grade is based on the level of achievement of the outcomes and objectives of the course outline and is reflected in quantifiable terms in the course syllabus.

REPRESENTATIVE TEXTBOOKS:
Data Structures and Other Objects Using C++; M.Main, W.Savitch, 2nd Ed., Addison Wesley Longman, 2000.
Data Abstraction and Problem Solving with C++: Walls and Mirrors; F.M. Carrano, J.J.Prichard, 3rd Ed., Pearson Education, 2002.
C++ How to Program; Dietel and Dietel, 3rd Ed., Prentiss-Hall, 2001.

RATIONALE
=========

RESOURCES REQUIRED
==================

MISCELLANEOUS
=============

Advisory generate desc:	Y	YES
Area department:	CSC	COMPUTER SCIENCE
Audit flag:	N	NOT AUDITABLE
Basic skills:	X	NOT BASIC SKILLS
Classification:	I	Career-Technical Education
Cost level:	00	VALUE NOT FOUND
Disciplines:		COMPUTER SCIENCE
Division:	00	VALUE NOT FOUND
Faculty service areas:		COMPUTER SCIENCE
Fee:	$0.00	
In-service:	X	NOT IN-SERVICE
Level below transfer:	X	NOT APPLICABLE
Matric-requiring:	X	Exempt from assessment
Maximum class size:	0	
Maximum wait list:	0	
Method of instruction:	03	LECTURE/LABORATORY
Non-credit category:	X	NOT APPLICABLE, CREDIT COURSE
Open entry/exit:	N	Not open entry/exit
Pacs activity:	1701	MATHEMATICS GENERAL
Pacs program project:	0000	
Preq/coreq generate desc:	Y	YES
Preq/coreq provisional:	N	NO
Preq/coreq reg check:	Y	PREREQUISITE RULES EXIST
Repeat group id:		
Requires instructor sig:	N	INSTRUCTOR'S SIGNATURE NOT REQUIRED
SAM classification:	B	Advanced occupational
Selected/special topic:	N	NOT A SELECTED TOPIC COURSE
Special class:	X	NOT A SPECIAL COURSE
TOP code:	0707.10	COMPUTER PROGRAMMING
Workload:	0.0000	