

CALIFORNIA STATE UNIVERSITY CHANNEL ISLANDS

NEW COURSE PROPOSAL

PROGRAM AREAS BIOLOGICAL AND PHYSICAL SCIENCES, MATH AND COMPUTER SCIENCE

1. Catalog Description of the Course. [Include the course prefix, number, full title, and units. Provide a course narrative including prerequisites and corequisites. If any of the following apply, include in the description: Repeatability (May be repeated to a maximum of ___ units); time distribution (Lecture ___ hours, laboratory ___ hours); non-traditional grading system (Graded CR/NC, ABC/NC). Follow accepted catalog format.]

COMP 462. ADVANCED OBJECT-ORIENTED PROGRAMMING (3)

Three hours of lecture in the lab per week.

Prerequisite: COMP. 350.

Principles of object-oriented design and programming based on languages such as JAVA, C++ and Smalltalk will be presented. Understanding of the role of objects, methods, message passing, encapsulation, and inheritance for effective programming will be stressed. Language structure versus particular engineering objectives will be analyzed. Design Patterns techniques will be an unifying theme.

2. Mode of Instruction.

	Units	Hours per Unit	Benchmark Enrollment
Lecture	<u>3</u>	<u>1</u>	<u>24</u>
Seminar	_____	_____	_____
Laboratory	_____	_____	_____
Activity	_____	_____	_____

3. Justification and Learning Objectives for the Course. (Indicate whether required or elective, and whether it meets University Writing, and/or Language requirements) [Use as much space as necessary]

The course is a required course for Computer Science majors according to accreditation guidelines.

Through this course, students will be able to

- Design object oriented algorithms in high-level language,
- Explain optimization problems and performance tradeoffs.
- Implement algorithms
- Use and cultivate sound engineering practices
- Use OOAD methodology to solve software engineering problems
- Use Rational Approach to carry on OOAD.
- Think and express design in the “design patterns”.
- Use testing strategies and modern refactoring techniques to provide flexible, and robust software solutions.
- Write tests plans and documentation.
- Design and implement a comprehensive selfstanding software solution - final project
- Organize and express ideas clearly and convincingly in oral and written forms.

This course is not designed to satisfy the University Writing or Language requirements.

4. Is this a General Education Course NO

If Yes, indicate GE category:

A (English Language, Communication, Critical Thinking)	
B (Mathematics & Sciences)	

C (Fine Arts, Literature, Languages & Cultures)	
D (Social Perspectives)	
E (Human Psychological and Physiological Perspectives)	

5. Course Content in Outline Form. *[Be as brief as possible, but use as much space as necessary]*

1. Introduction; OO; UML; Rational Process
2. Software Design Principles I
3. Software Design Principles II; Introduction to Design Patterns
4. Creational Design Patterns
5. Structural Design Patterns
6. Behavioral Design Patterns
7. Introduction to Object Components and Data Objects
8. Refactoring and testing
9. Introduction to the EAI

6. References. *[Provide 3 - 5 references on which this course is based and/or support*

Component Software: Beyond Object-Oriented Programming, by C. Szyperski, ISBN: 0-201-74572-0, Addison Wesley Professional (2000).

Design Patterns : Elements of Reusable Object-Oriented Software by Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John, ISBN 0201634988 (1998), Addison-Wesley Pub Co.

Refactoring, by Martin Fowler, (1999) Addison-Wesley Pub Co.

7. List Faculty Qualified to Teach This Course.

All Computer Science faculty.

8. Frequency.

a. Projected semesters to be offered: Fall Spring Summer

9. New Resources Required.

a. Computer (data processing), audio visual, broadcasting needs, other equipment

Use of existing computer lab.

b. Library needs

none

c. Facility/space needs

none

10. Consultation.

Attach consultation sheet from all program areas, Library, and others (if necessary)

11. If this new course will alter any degree, credential, certificate, or minor in your program, attach a program modification.

Proposer of Course

Date