CALIFORNIA STATE UNIVERSITY CHANNEL ISLANDS

COURSE MODIFICATION PROPOSAL

Courses must be submitted by October 15, 2010,
to make the next catalog (2011-12) production

DATE (CHANGE DATE EACH TIME REVISED): 6/9/10; REV 10.11.10

PROGRAM AREA(S) : COMPUTER SCIENCE

Directions: All of sections of this form must be completed for course modifications. Use YELLOWED areas to enter data. All documents are stand alone sources of course information.

1. Course Information.

[Follow accepted catalog format.] ***(Add additional prefixes i f cross-listed)***

OLD

Prefix COMP Course# 150 Title Object-oriented programming Units (4)
4 hours lecture per week
hours blank per week

X Prerequisites: Programming experience
Consent of Instructor Required for Enrollment
Corequisites:

Catalog Description (Do not use any symbols): Introduction to algorithms, their representation, design, structuring, analysis and optimization. The course introduces the concept of object paradigm, design and implementation of algorithms as structured programs in a high level language.

General Education Categories B4
Lab Fee Requested

Graded
CR/NC
X A - F

Repeatable for up to units
Total Completions

Course Level:
X Undergraduate
Post-bac/Credential
Graduate

Optional (Student's choice)

Multiple Enrollment in same semester

NEW

Prefix COMP Course# 150 Title Object-Oriented Programming Units (4)
4 hours lecture per week
hours blank per week

X Prerequisites: Math105 or equivalent
Consent of Instructor Required for Enrollment
Corequisites:

Catalog Description (Do not use any symbols)**:** Introduction to algorithms, their representation, design, structuring, analysis and optimization. The course introduces the concept of object paradigm, design and implementation of algorithms as structured programs in a high level language.

X General Education Categories B4
Lab Fee Requested

Graded
CR/NC
X A - F

Repeatable for up to units
Total Completions

Course Level:
X Undergraduate
Post-bac/Credential
Graduate

Optional (Student's choice)

Multiple Enrollment in same semester

2. Mode of Instruction (Hours per Unit are defaulted)

Hegis Code(s)________________________________
(Provided by the Dean)

Existing

	Units	Hours Per Unit	Benchmark Enrollment	Graded
Lecture	4	1	24	y
Seminar		1		
Lab		3		
Activity		2		
Field Studies				
Indep Study				
Other blank				

Proposed

	Units	Hours Per Unit	Benchmark Enrollment	Graded	CS No. (filled out by Dean)
Lecture	4	1	24	y	
Seminar		1			
Lab		3			
Activity		2			
Field Studies					
Indep Study					
Other blank					

3. Course Attributes:

X **General Education Categories**: All courses with GE category notations (including deletions) must be submitted to the GE website: http://summit.csuci.edu/geapproval. Upon completion, the GE Committee will forward your documents to the Curriculum Committee for further processing.

A (English Language, Communication, Critical Thinking)

□ A-1 Oral Communication
□ A-2 English Writing
□ A-3 Critical Thinking

B (Mathematics, Sciences & Technology)

□ B-1 Physical Sciences
□ B-2 Life Sciences – Biology
□ B-3 Mathematics – Mathematics and Applications
X B-4 Computers and Information Technology

C (Fine Arts, Literature, Languages & Cultures)

□ C-1 Art
□ C-2 Literature Courses
□ C-3a Language
□ C-3b Multicultural
□ **D (Social Perspectives)**
□ **E (Human Psychological and Physiological Perspectives)**
□ **UDIGE/INTD Interdisciplinary**
□ **Meets University Writing Requirement**
□ **Meets University Language Requirement**

□ **American Institutions, Title V Section 40404**: □ Government □ US Constitution □ US History
Refer to website, Exec Order 405, for more information: http://senate.csuci.edu/comm/curriculum/resources.htm

□ **Service Learning Course** (Approval from the Center for Community Engagement must be received before you can request this course attribute).

4. Justification and Requirements for the Course. *[Make a brief statement to justify the need for the course]*

OLD

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

X Requirement for the Major/Minor
□ Elective for the Major/Minor
□ Free Elective

NEW

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

X Requirement for the Major/Minor
□ Elective for the Major/Minor
□ Free Elective

Submit Program Modification if this course changes your program.

5. Student Learning Outcomes. (List in numerical order. You may wish to visit resource information at the following website: http://senate.csuci.edu/comm/curriculum/resources.htm)

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

OLD

□ Apply the core concepts of the object oriented programming.
• Analyze, design, implement and test programs, organized around the central idea of the Object.
• Discuss the central idea of programming
•Use Object oriented analysis and design methodology to build models of the simple objects
• Discuss code Encapsulation as the engineering tool for ensuring code reuse, and stability..
• Participate in the programming activities, as a team member.
• Discuss their ideas on the proposed solutions of the

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

NEW

□ Apply the core concepts of the object oriented programming.
• Analyze, design, implement and test programs, organized around the central idea of the Object.
•Use Object oriented analysis and design methodology to build models of the simple objects
• Apply code Encapsulation as the engineering tool for ensuring code reuse, and stability..
• Actively participate as a team member in the programming activities.
• Organize and express their ideas on the proposed solutions of the assignments clearly in written form.

assignments.
• Write English language comments in the source code
• Realize similarities and differences between programming and natural languages.
• Use diagrams and charts as powerful form of the pre-language level modeling.
• Organize and express ideas clearly and convincingly in oral and written forms.

• Write English language comments in the source code
• Use diagrams and charts as powerful form of the pre-language level modeling.
• Organize and express ideas clearly and convincingly in oral and written forms.

6. Course Content in Outline Form. (Be as brief as possible, but use as much space as necessary)

OLD

□ Introduction to IDE, and a first program.
• Testing as a programming activity. Client use of the Object's Api, first use of Objects.
• Variables and assignments. Data typing. Primitive data types.
• Flow of the execution. Variables, Boolean conditions and control structures.
Automation of the repetitious task and self-referencing.
• Block structure of the code. Scope of the name.
• Objects as statefull, dynamic models. Member variables and methods as modeling ingredients.
Classes. Constructors. Programming as modeling state and behavior of the Entity.
• Object Encapsulation and implementation hiding, role of the api. OO structure of the code.
• Method's definitions and calls. Chaining. Overloading. More on Constructors.
• Indexed data types, arrays.
• Object view on the "smart" date structures. Lists.
• OOD: from the requirement to the api. Separation of the api and the implementation.
Programming for the contract.
• Engineering benefits of OOAD.
• Sublasses. Inheritance as the refinement, and enhancement of the functionality. Thin wrappers.
• Inheritance and polymorpism of the behavior as the enrichment of the data type. Casting.
• Sorting algorithms.
• Binary searches. Recursion.

NEW

□ Introduction to IDE, and a first program.
• Testing as a programming activity. Client use of the Object's Api, first use of Objects.
• Variables and assignments. Data typing. Primitive data types.
• Flow of the execution. Variables, Boolean conditions and control structures.
Automation of the repetitious task and self-referencing.
• Block structure of the code. Scope of the name.
• Objects as statefull, dynamic models. Member variables and methods as modeling ingredients.
Classes. Constructors. Programming as modeling state and behavior of the Entity.
• Object Encapsulation and implementation hiding, role of the api. OO structure of the code.
• Method's definitions and calls. Chaining. Overloading. More on Constructors.
• Indexed data types, arrays.
• Object view on the "smart" date structures. Lists.
• OOD: from the requirement to the api. Separation of the api and the implementation.
Programming for the contract.
• Engineering benefits of OOAD.
• Sublasses. Inheritance as the refinement, and enhancement of the functionality. Thin wrappers.
• Inheritance and polymorpism of the behavior as the enrichment of the data type. Casting.
• Sorting algorithms.
• Binary searches. Recursion.

Does this course content overlap with a course offered in your academic program? Yes **No** X
If YES, what course(s) and provide a justification of the overlap.

Does this course content overlap a course offered in another academic area? Yes **No** X
If YES, what course(s) and provide a justification of the overlap.

Overlapping courses require Chairs' signatures.

7. Cross-listed Courses (Please note each prefix in item No. 1)

A. List cross-listed courses (Signature of Academic Chair(s) of the other academic area(s) is required).
B. List each cross-listed prefix for the course:
C. Program responsible for staffing:

8. References. *[Provide 3-5 references]*

OLD 1) JAVA, An Intro to Computer Science and Programming by Walter Savitch : (current edit.) Prentice Hall, ISBN 0-13-031697-0
2) Programming.Java: An Introduction to Programming Using Java by Rick Decker, Stuart Hirshfield , Brooks/Cole Pub Co; ISBN: 0534371094 ; 2 edition (1999)
3) Java Software Solutions: Foundations of Program Design, Update JavaPlace
by John Lewis , William Loftus, Addison-Wesley Publishing; ISBN: 0201781298 ; 3rd edition (2002)

NEW 1) JAVA, An Intro to Computer Science and Programming by Walter Savitch : (current edit.) Prent ceHall, ISBN 0-13-031697-0
2) Programming.Java: An Introduction to Programming Using Java by Rick Decker, Stuart Hirshfield , Brooks/Cole Pub Co; ISBN: 0534371094 ; 2 edition (1999)
3) Java Software Solutions: Foundations of Program Design, Update JavaPlace
by John Lewis , William Loftus, Addison-Wesley Publishing; ISBN: 0201781298 ; 3rd edition (2002)

9. Tenure Track Faculty qualified to teach this course.
All Computer Science faculty

10. Requested Effective Date or First Semester offered: Fall 2011

11. New Resource Requested: Yes No X
If YES, list the resources needed.

A. Computer Needs (data processing, audio visual, broadcasting, other equipment, etc.)

B. Library Needs (streaming media, video hosting, databases, exhibit space, etc.)

C. Facility/Space/Transportation Needs:

D. Lab Fee Requested: Yes No (Refer to the Dean's Office for additional processing)
E. Other.

12. Indicate Changes and Justification for Each. *[Check all that apply and follow with justification. Be as brief as possible but, use as much space as necessary.]*

Course title
Prefix/suffix
Course number
Units
Staffing formula and enrollment limits
X Prerequisites/Corequisites
Catalog description
Mode of Instruction

Course Content
X Course Learning Objectives
References
GE
Other
Reactivate Course

Justification: The prerequisite change is because instructors have found that mathematical competence is more important than prior programming experience in achieving success in the course. The learning outcomes are modified to make them assessable.

13. Will this course modification alter any degree, credential, certificate, or minor in your program? Yes No X
If, YES attach a program update or program modification form for all programs affected.
Priority deadline for New Minors and Programs: October 4, 2010 of preceding year.
Priority deadline for Course Proposals and Modifications: October 15, 2010.
Last day to submit forms to be considered during the current academic year: April 15th.

Peter Smith   6/9/10

Proposer(s) of Course Modification   Date

Type in name. Signatures will be collected after Curriculum approval.

<u>Request for COMP 150: Object-oriented Programming to be added to GE Category B4: Computers and Information Technology.</u>

Committee Response:
<u>Approved by committee on 09-24-2010</u>

Criteria and Justifications Submitted:

- *Promote the understanding and appreciation of the methodologies of math or science as investigative tools and the limitations of mathematical or scientific endeavors*
  This course covers the development, implementation and testing of algorithms to solve mathematical and non-mathematical problems with an object-oriented approach.
- *Present mathematical or scientific knowledge in a historical prespective and the influences of math and science on the development of world civilizations, both past and present*
  This course covers the development of programming from simple machine code to modern programming environments.
- *Apply inductive and deductive reasoning processes and explore fallacies and misconceptions in the mathematical or scientific areas*
  An important component of the course is the interaction between instructor and student in determining whether a solution to an assignment is in fact correct.
- *Include use of computers or information technology to solve problems as appropriate*
  This course makes extensive use of computers in solving problems.

Course: COMP 150
If your course has a General Education Component or involves Center affiliation, the Center will also sign off during the approval process.

Multiple Chair fields are available for cross-listed courses.

Program Chair		
	Signature	Date
Program Chair		
	Signature	Date
Program Chair		
	Signature	Date
General Education Chair		
	Signature	Date
Center for Intl Affairs Director		
	Signature	Date
Center for Integrative Studies Director		
	Signature	Date
Center for Multicultural Engagement Director		
	Signature	Date
Center for Civic Engagement and Service Learning Director		
	Signature	Date
Curriculum Chair		
	Signature	Date
Dean of Faculty		
	Signature	Date