

CALIFORNIA STATE UNIVERSITY CHANNEL ISLANDS

COURSE MODIFICATION PROPOSAL

Courses must be submitted by October 15, 2013, and finalized by the end of the fall semester to make the next catalog (2014-15) production

DATE (CHANGE DATE EACH TIME REVISED): 3/20/2014, REV: 4/22/2014

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. Indicate Changes and Justification for Each. [Mark an X by all change areas that apply then please follow-up your X's with justification(s) for each marked item. Be as brief as possible but, use as much space as necessary.]

- | | |
|---|---|
| <input type="checkbox"/> Course title | <input type="checkbox"/> Course Content |
| <input type="checkbox"/> Prefix/suffix | <input type="checkbox"/> Course Learning Outcomes |
| <input type="checkbox"/> Course number | <input type="checkbox"/> References |
| <input checked="" type="checkbox"/> Units | <input type="checkbox"/> GE |
| <input type="checkbox"/> Staffing formula and enrollment limits | <input type="checkbox"/> Other <input type="checkbox"/> |
| <input type="checkbox"/> Prerequisites/Corequisites | <input type="checkbox"/> Reactivate Course |
| <input type="checkbox"/> Catalog description | |
| <input checked="" type="checkbox"/> Mode of Instruction | |

Justification: Our experience with adding lab sections to our programming courses executed a few semesters ago has proven extremely successful as evidenced by visibly better programming skills, and therefore better preparation for further courses as well as – ultimately – for the marketplace, of our students. It has been also praised a lot by the students in their evaluations of the instructors. Hence, we believe that the students of COMP 150 should also benefit from the expanded hands-on element of the course. No other changes are proposed here; just the change from 4 units of lecturing to the 3+1 formula: 3 hours of lecturing and 1 unit of lab (i.e., three hours in the lab for the students).

2. Course Information.

[Follow accepted catalog format.] (Add additional prefixes if cross-listed)

OLD

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

☐ Prerequisites: MATH 105 or equivalent and COMP 105 or permission of the instructor

☐ 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

Grading Scheme (Select one below):

x A – F

☐ Credit/No Credit

☐ Optional (Student's Choice)

Repeatable for up to ☐ units

NEW

Prefix COMP Course# 150
 Title Introduction to Object-Oriented Programming
 Units (3+1)
 3 hours lecture per week
 3 hours lab per week

☐ Prerequisites: MATH 105 or equivalent and COMP 105 or permission of the instructor

☐ Consent of Instructor Required for Enrollment

☐ Corequisites: ☐

Catalog Description (Do not use any symbols):

Hours: Three hours lecture and three hours lab per week
 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

Grading Scheme (Select one below):

x A – F

☐ Credit/No Credit

☐ Optional (Student's Choice)

Repeatable for up to ☐ units

Total Completions
 Multiple Enrollment in Same Semester Y/N
 Course Level:
☒ Undergraduate
☐ Post-Baccalaureate
☐ Graduate

Total Completions
 Multiple Enrollment in Same Semester Y/N
 Course Level:
☒ Undergraduate
☐ Post-Baccalaureate
☐ Graduate

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

Hegis Code(s) _____
 (Provided by the Provost Office)

Existing

Proposed

	Units	Hours Per Unit	Default Section Size	Graded		Units	Hours Per Unit	Default Section Size	Graded	CS No. (filled out by Provost Office)
Lecture	<u>4</u>	<u>1</u>	<u>24</u>	<input type="checkbox"/>	Lecture	<u>3</u>	<u>1</u>	<u>24</u>	<input type="checkbox"/>	<input type="checkbox"/>
Seminar	<input type="text"/>	<u>1</u>	<input type="text"/>	<input type="checkbox"/>	Seminar	<input type="text"/>	<u>1</u>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lab	<input type="text"/>	<u>3</u>	<input type="text"/>	<input type="checkbox"/>	Lab	<u>1</u>	<u>3</u>	<u>24</u>	<input type="checkbox"/>	<input type="checkbox"/>
Activity	<input type="text"/>	<u>2</u>	<input type="text"/>	<input type="checkbox"/>	Activity	<input type="text"/>	<u>2</u>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Field Studies	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	Field Studies	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Indep Study	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	Indep Study	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other blank	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	Other blank	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
Online	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	Online	<input type="text"/>		<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

4. Course Attributes:

☐ **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
- ☒ 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 (Graduation Writing Assessment Requirement)

Meets University Language Requirement

American Institutions, Title V Section 40404: Government US Constitution US History
Regarding 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).

Online Course (Answer YES if the course is ALWAYS delivered online).

5. 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.

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

x Requirement for the Major/Minor
Elective for the Major/Minor
Free Elective

Submit Program Modification if this course changes your program.

6. Student Learning Outcomes. (List in numerical order. Please refer to the Curriculum Committee's "Learning Outcomes" guideline for measurable outcomes that reflect elements of Bloom's Taxonomy: <http://senate.csuci.edu/comm/curriculum/resources.htm>. The committee recommends 4 to 8 student learning outcomes, unless governed by an external agency (e.g., Nursing).

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.
•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.
- 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.
- Reason inductively and deductively. (GE 2.1)

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.
- 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.
- Reason inductively and deductively. (GE 2.1)

7. 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.

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" data structures. Lists.
- OOD: from the requirement to the api. Separation of the api and the implementation.

Programming for the contract.

Engineering benefits of OOAD.

- Subclasses. Inheritance as the refinement, and enhancement of the functionality. Thin wrappers.
- Inheritance and polymorphism of the behavior as the enrichment of the data type. Casting.
- Sorting algorithms.
- Binary searches. Recursion.

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" data structures. Lists.
- OOD: from the requirement to the api. Separation of the api and the implementation.

Programming for the contract.

Engineering benefits of OOAD.

- Subclasses. Inheritance as the refinement, and enhancement of the functionality. Thin wrappers.
- Inheritance and polymorphism 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 ☒

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 ☒

If YES, what course(s) and provide a justification of the overlap.

Overlapping courses require Chairs' signatures.

8. Cross-listed Courses (Please note each prefix in item No. 1) *Beyond three disciplines consult with the Curriculum Committee.*

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

9. References. [Provide 3-5 references]

OLD 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)

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)

10. Tenure Track Faculty qualified to teach this course.

All Computer Science Faculty

11. Requested Effective Date or First Semester offered: Fall 2014

12. New Resource Requested: Yes ☐ No ☒

If YES, list the resources needed.

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

None

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

None

C. Facility/Space/Transportation Needs:

None

D. Lab Fee Requested: Yes ☐ No ☒ (Lab fee requests should be directed to the Student Fee Committee)

E. Other. None

13. Will this course modification alter any degree, credential, certificate, or minor in your program? Yes ☐ No ☒

If, YES attach a program update or program modification form for all programs affected.

Priority deadline for New Minors and Programs: October 1, 2013 of preceding year.

Priority deadline for Course Proposals and Modifications: October 15, 2013.

Last day to submit forms to be considered during the current academic year: April 15th.

AJ Bieszczad

3/20/2014

Proposer(s) of Course Modification

Date

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

Approval Sheet

Course: [REDACTED]

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.

The CI program review process includes a report from the respective department/program on its progress toward accessibility requirement compliance. By signing below, I acknowledge the importance of incorporating accessibility in course design.

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
AVP		
Signature		Date