### CALIFORNIA STATE UNIVERSITY CHANNEL ISLANDS COURSE MODIFICATION PROPOSAL Courses must be submitted by October 15, 2012, and finalized by the end of the fall semester to make the next catalog (2013-14) production

DATE (CHANGE DATE EACH TIME REVISED): 8/28/12; 11/21/12; 11/26/12GE; REV 11.29.12 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 all change areas 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 Outcomes References GE Other Reactivate Course

**Justification:** Currently nothing to prevent students from taking COMP 105 and COMP 150 in the same semester. Appropriate to take one or the other. Added a more general learning outcome.

#### 2. 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: MATH 105 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.

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: X Undergraduate Post-Baccalaureate Graduate

#### NEW

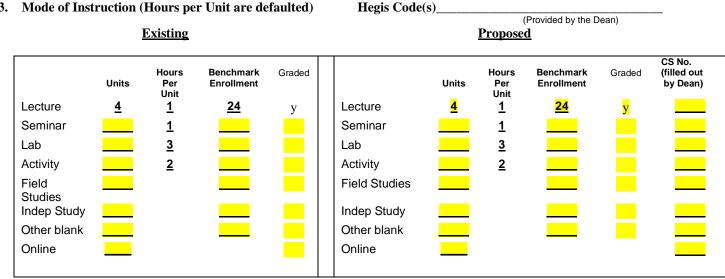
Prefix COMP Course# 150 Title Object-oriented Programming Units (4) 4 hours lecture per week hours blank per week

X 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

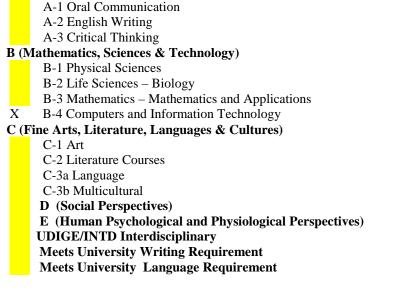
Grading Scheme (Select one below): X A – F Credit/No Credit Optional (Student's Choice) Repeatable for up to units Total Completions Multiple Enrollment in Same Semester Y/N Course Level: X Undergraduate Post-Baccalaureate Graduate



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

#### 4. 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)

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).

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

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

OLD

OLD

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

**NEW** 

- □ Apply the core concepts of the object oriented programming. □ Apply the core concepts of the object oriented programming. • Analyze, design, implement and test programs, organized around the central idea of the Object. around the central idea of the Object. •Use Object oriented analysis and design methodology to build models of the simple objects 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 prelanguage level modeling.
- Organize and express ideas clearly and convincingly in oral and written forms.
- 7. Course Content in Outline Form. (Be as brief as possible, but use as much space as necessary) OLD 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.

#### 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

· Analyze, design, implement and test programs, organized •Use Object oriented analysis and design methodology to build • 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 prelanguage level modeling. • Organize and express ideas clearly and convincingly in oral and written forms. • Reason inductively and deductively. (GE 2.1) □ 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.

> > 3

| <ul> <li>Engineering benefits of OOAD.</li> <li>Subclasses. Inheritance as the refinement, and<br/>enhancement of the functionality. Thin wrappers.</li> <li>Inheritance and polymorphism of the behavior as the<br/>enrichment of the data type. Casting.</li> <li>Sorting algorithms.</li> <li>Binary searches. Recursion.</li> </ul> | <ul> <li>Engineering benefits of OOAD.</li> <li>Subclasses. Inheritance as the refinement, and<br/>enhancement of the functionality. Thin wrappers.</li> <li>Inheritance and polymorphism of the behavior as the<br/>enrichment of the data type. Casting.</li> <li>Sorting algorithms.</li> <li>Binary searches. Recursion.</li> </ul> |  |  |  |
|---|---|--|--|--|
| Binary searches. Recursion.     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 over  | erlap   |  |  |  |

- 8. 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:

#### 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 2013

- 12. 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.
- 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 1, 2012 of preceding year. Priority deadline for Course Proposals and Modifications: October 15, 2012.

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

| Peter Smith |  |
|-------------|--|
|-------------|--|

<mark>8/28/12</mark>

Date

Proposer(s) of Course Modification Type in name. Signatures will be collected after Curriculum approval. GE Committee response to your request have COMP150: Object-oriented Programming added to B4: Computers and Information Technology

Approved by committee on 09-24-2010 Approved by committee on 11-26-12

\_\_\_\_\_

Request Submitted

Courses COMD150 Object oriented

Course: COMP150 Object-oriented Programming Area: B4 Computers and Information Technology Date Submitted: 8/27/2010 12:00:00 AM Date Approved: 11/26/2012 10:34:03 AM

1. 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 nonmathematical problems with an object-oriented approach.

2. 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.

3. 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.

4. Include use of computers or information technology to solve problems as appropriate

This course makes extensive use of computers in solving problems.

# **Approval Sheet**

## 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.

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<br>Director                   |           |      |
|  | Signature | Date |
| Center for Multicultural<br>Engagement Director              |           |      |
|  | Signature | Date |
| Center for Civic Engagement<br>and Service Learning Director |           |      |
|  | Signature | Date |
| Curriculum Chair   |           |      |
|  | Signature | Date |
| AVP  |           |      |
|  |           |      |