

LSC Use Only Proposal No:	UWUCC Use Only Proposal No: 11-124m.
LSC Action-Date:	UWUCC Action-Date: AP-4/3/12 Senate Action Date: App-5/01/12

Curriculum Proposal Cover Sheet - University-Wide Undergraduate Curriculum Committee

Contact Person(s) David T. Smith	Email Address dtsmith@iup.edu
Proposing Department/Unit Computer Science	Phone 7-4478

Check all appropriate lines and complete all information. Use a separate cover sheet for each course proposal and/or program proposal.

1. Course Proposals (check all that apply)

New Course Course Prefix Change Course Deletion
 Course Revision Course Number and/or Title Change Catalog Description Change

Current course prefix, number and full title: **COSC 320 Software Engineering Practice**

Proposed course prefix, number and full title, if changing: **COSC 473 Software Engineering Practice**

2. Liberal Studies Course Designations, as appropriate
 This course is also proposed as a Liberal Studies Course (please mark the appropriate categories below)

Learning Skills Knowledge Area Global and Multicultural Awareness Writing Across the Curriculum (W Course)
 Liberal Studies Elective (please mark the designation(s) that applies – must meet at least one)

Global Citizenship Information Literacy Oral Communication
 Quantitative Reasoning Scientific Literacy Technological Literacy

3. Other Designations, as appropriate

Honors College Course Other: (e.g. Women's Studies, Pan African)

4. Program Proposals

Catalog Description Change Program Revision Program Title Change New Track
 New Degree Program New Minor Program Liberal Studies Requirement Changes Other

Current program name: _____

Proposed program name, if changing: _____

5. Approvals	Signature	Date
Department Curriculum Committee Chair(s)	<i>[Signature]</i>	2/6/12
Department Chairperson(s)	<i>[Signature]</i>	2/10/2012
College Curriculum Committee Chair	<i>[Signature]</i>	3/7/12
College Dean	<i>[Signature]</i>	3/12/12
Director of Liberal Studies (as needed)		
Director of Honors College (as needed)		
Provost (as needed)		
Additional signature (with title) as appropriate		
UWUCC Co-Chairs	<i>[Signature]</i>	4/24/12

Received

APR 24 2012

Liberal Studies

Received

MAR 12 2012

Liberal Studies

Course Revision and Number Change - Computer Science Curriculum

Part II. Description of Curriculum Change

1. New Syllabus of Record

I. Course Description

COSC 473 Software Engineering Practice

3c-01-3cr

Prerequisite: COSC 319 or instructor permission.

Planning, design, and implementation of large software systems using software engineering techniques. Students work on project teams on real or realistic software development projects. Credit for either COSC 473 or 493, but not both, may count toward computer science major requirements for graduation; the other course credits will be counted as free electives.

II. Course Outcomes

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

1. Apply what they have learned in the computer science curriculum to a medium sized real world software development project.
2. Explain concepts in the computing field that are difficult to teach in instructor led course, such as user interaction, and testing, etc.
3. Experience a simulated business and industrial environment in which a computer professional must learn to function.
4. Assess his/her professional, emotional, social and intellectual growth.
5. Organize career goals and formulate course selection to attain those goals.
6. Develop writing skills that are necessary in the professional world of computing.

III. Detailed Course Outline

- | | |
|---|---------|
| 1. Introduction | 3 hours |
| 2. Project Plan | |
| a. We have been presented with a Problem statement from the client. You need to have a clear understanding of the presented specification, read through and understand what the system is supposed to/not do and any problems understanding the requirements. | |
| b. Produce a formal problem statement document. | |
| c. Produce a project plan- a project planning document (template provided) is a document that estimate the effort required to produce your proposed system. | |
| 3. Requirements Analysis | 6 hours |
| a. Meeting with the client to clarify/confirm the requirements. Please email the client the first week of classes and schedule a meeting. | |
| b. At the meeting, you are expected to present in REPORT form and in PP | |
| i. Ultimate goal of the project | |
| ii. What the system is suppose to do / not do. | |
| iii. Your primary concerns with the requirements. | |
| iv. Process model | |
| v. Organizational structure and responsibilities. | |
| vi. Based-line effort schedule | |
| vii. Range of skills and experience | |
| viii. Overview of data requirements | |
| ix. User view of the system | |

Course Revision and Number Change - Computer Science Curriculum

Most of the items covered are coming from the Project Plan report.

- c. Start the requirements engineering process
 - i. Review of the UML models for requirements engineering
 - ii. Develop a requirements specification document.
 - iii. Once we all know what the system is suppose to do, then Prototyping should start there and then.

- 4. Design and Implementation 30 hours
 - a. Design
 - i. Architectural design
 - ii. Detailed design
 - iii. Develop a design document
 - b. Rest of the implementation and testing of the system.
 - c. Develop a User Manual.
 - d. Produce a systems manual which consists of a series of all reports
 - i. Planning document
 - ii. Software Requirements Specification
 - iii. Software Design Specification
 - iv. Validation and Verification plan

- 5. Testing and Deployment 3 hours
 - a. Complete systems testing
 - b. Package system for deployment and present demo to class (Burn the whole system and manuals onto a CD).
 - c. You are also required to produce an individual report on your experiences of working on the system. This has to be submitted on the final day of classes.

- Total: 42 hours

- 6. Finals Week 2 hours
 - a. Deliver complete and packaged system, including the user manual to client
 - b. Do a final presentation to the client

IV. Evaluation Methods

Group Journals	10 %
Group Reports	50 %
Individual report	10 %
Presentations	15 %
Class Attendance grade	15 %

Grading Scale: 90-100% A, 80-89% B, 70-79% C, 60-69% D, 0-59% F

V. Attendance policy

The attendance policy will conform to the University wide attendance criteria.

VI. Textbooks, Readings, and other Readings

Mainly Internet resources will be used for this course including class notes from the COSC 319 class. Students are encouraged to acquired a reference book in software engineering The following are recommended, but not required:

- Ian Sommerville, *Software Engineering*, 9th Edition: Addison Wesley, 2010
- J. Fernando Naveda and Stephen B. Seidman, *IEEE Computer Society Real-World Software Engineering Problems: A Self-Study Guide for Today's Software Professional*, Wiley Interscience, 2006

Course Revision and Number Change - Computer Science Curriculum

VII. Special Resource Requirements

None.

VIII. Bibliography

Braude, Eric and Michael E. Bernstein, *Software Engineering: Modern Approaches*, 2nd Edition, Wiley, 2010.

Bruegge, Bernd and Allen H. Dutoit, *Object-Oriented Software Engineering Using UML, Patterns, and Java*, 3rd Edition, Prentice Hall, 2009.

Jones, Capers, *Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies*, McGraw-Hill, 2009.

Pfleeger, Shari Lawrence and Joanne M. Atlee, *Software Engineering: Theory and Practice*, 4th Edition, Prentice Hall, 2009.

Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw-Hill, 2010.

Schach, Stephen, *Object-Oriented and Classical Software Engineering*, 8th Edition, McGraw-Hill, 2010.

van Vliet, Hans, *Software Engineering: Principles and Practice*, 3rd Edition, Wiley, 2008.

2. Current and Proposed Course Number and Title

Current (old) Course Number and Title:

COSC 320 Software Engineering Practice

Proposed (new) Course Number and Title:

COSC 473 Software Engineering Practice

3. Justification/Rationale

Software Engineering Practice is the practicum that is intended to be a capstone of the Computer Science program where student apply their skills and knowledge to sizable projects that are representative to industry. It is the on-campus counterpart to an internship experience. Given this, it is being renumbered to 473 to indicate a senior level course that is on par with COSC 493 Internship experience. Furthermore, this change provides alignment with the department's ABET accreditation plan in which COSC 493 or COSC 473 are the designated capstone courses.

4. Old Syllabus of Record

I. Course Description

COSC 320 Software Engineering Practice

3c-0l-3cr

Prerequisite: COSC 319 or instructor permission.

Planning, design, and implementation of large software systems using software engineering techniques. Students work on project teams on real or realistic software development projects. Credit for either COSC 473 or 493, but not both, may count toward computer science major requirements for graduation; the other course credits are free electives.

II. Course Outcomes

Course Revision and Number Change - Computer Science Curriculum

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

1. Apply what they have learned in the in the computer science curriculum to a medium sized real world software development project.
2. Learn concepts in the computing field that are difficult to teach in instructor led course, such as user interaction, and testing, etc.
3. Experience a simulated business and industrial environment in which a computer professional must learn to function.
4. Grow professionally, emotionally, socially and intellectually.
5. Sharpen their focus on career goals and course selection to reach those goals.
6. Develop writing skills that are necessary in the professional world of computing.

III. Detailed Course Outline

1. Week One 3 hours
2.
 - a. We have been presented with a Problem statement from the client. You need to have a clear understanding of the presented specification, read through and understand what the system is supposed to/not do and any problems understanding the requirements.
 - b. Produce a formal problem statement document.
 - c. Produce a project plan- a project planning document (template provided) is a document that estimate the effort required to produce your proposed system.
3. Weeks Two and Three 6 hours
 - a. Meeting with the client to clarify/confirm the requirements. Please email the client the first week of classes and schedule a meeting.
 - b. At the meeting, you are expected to present in REPORT form and in PP
 - i. Ultimate goal of the project
 - ii. What the system is suppose to do / not do.
 - iii. Your primary concerns with the requirements.
 - iv. Process model
 - v. Organizational structure and responsibilities.
 - vi. Based-line effort schedule
 - vii. Range of skills and experience
 - viii. Overview of data requirements
 - ix. User view of the system
 - x. Most of the items covered are coming from the Project Plan report.
 - c. Start the requirements engineering process
 - i. Review of the UML models for requirements engineering
 - ii. Develop a requirements specification document.
 - iii. Once we all know what the system is suppose to do, then Prototyping should start there and then.
4. Weeks 4 through 13 30 hours
 - a. Design
 - i. Architectural design
 - ii. Detailed design
 - iii. Develop a design document
 - b. Rest of the implementation and testing of the system.

Course Revision and Number Change - Computer Science Curriculum

- c. Develop a User Manual.
- d. Produce a systems manual which consists of a series of all reports
 - i. Planning document
 - ii. Software Requirements Specification
 - iii. Software Design Specification
 - iv. Validation and Verification plan

5. Week 14

3 hours

- a. Complete systems testing
- b. Package system for deployment and present demo to class (Burn the whole system and manuals onto a CD).
- c. You are also required to produce an individual report on your experiences of working on the system. This has to be submitted on the final day of classes.

Total: 42 hours

6. Finals Week

2 hours

- a. Deliver complete and packaged system, including the user manual to client
- b. Do a final presentation to the client

IV. Evaluation Methods

Group Journals	10 %
Group Reports	50 %
Individual report	10 %
Presentations	15 %
Class Attendance grade	15 %

Attendance policy:

Each student is expected to attend classes EVERYDAY and to arrive on time. Since this is a seminar course and active participation is needed, the attendance policy will be strictly enforced. You may miss up to one class with a written doctor's excuse or a printed funeral notice without losing attendance points. Failure to attend classes will adversely affect your grade. Attendance will be factored into the class participation part of the grade.

Travel, interview trips, and absences: If you must miss class for an interview or any other reason, inform your group and your instructor well in advance. It is your responsibility to be certain that your tasks are covered during your absence. Do not assume that your group will just do the work when you are not here -- this must be planned. In order to learn what has happened in class during your absence, refer to the I drive and ask classmates.

The final grade for the project work is determined by a combination of the student participation grade and the instructor group grade. Each contributes 50%. The student participation grade will be based on peer evaluation with the main weight given to the other group members' evaluation. A peer evaluation form will be distributed in class on the day of the submission of an assigned report. The peer evaluation form highlights a summary of the participation of each group member. Participation consists of attendance to meetings and ability to finish assigned task and any grievances filed against group members with poor participation.

Attendance policy: The attendance policy will conform to the University wide attendance criteria.

V. Textbooks, Readings, and other Readings

Mainly Internet resources will be used for this course including class notes from the COSC319 class. Students are encouraged to acquire a reference book in software engineering. The following are recommended, but not required:

- Ian Sommerville, *Software Engineering*, 7th Edition: Addison Wesley, 2004
- J. Fernando Naveda and Stephen B. Seidman, *IEEE Computer Society Real-World Software Engineering Problems: A Self-Study Guide for Today's Software Professional*, Wiley Interscience, 2006
- Roger Pressman, *Software Engineering Software Engineering: A Practitioner's Approach* 6th edition, McGraw-Hill, 2004.

Course Revision and Number Change - Computer Science Curriculum

Stephen Schach, *Object-Oriented and Classical Software Engineering*, 7th edition, McGraw-Hill, 2007.

VI. Special Resource Requirements

None.

5. Letters or Acknowledgement

N/A