CGS Agenda Item: 18-35
Effective Fall 2018

Eastern Illinois University
New/Revised Course Proposal Format
(Approved by CAA on 4/3/14 and CGS on 4/15/14, Effective Fall 2014)

Banner/Catalog Information (Coversheet)

1. **_X__New Course or _____Revision of Existing Course**

2. **Course prefix and number:** _TEC 5853 ____

3. **Short title:** _ Sys. Analysis & Design____________________________________

4. **Long title:** _System Analysis & Design ________________________________

5. **Hours per week:** _2__ Class _2__ Lab _3_ Credit

6. **Terms:** ___ Fall ___ Spring __ Summer _X_ On demand

7. **Initial term**: _X__ Fall ___ Spring ___ Summer Year: _2018___

8. **Catalog course description:** _This course provides introduction to software engineering principles, software development life-cycles, and modelling in software engineering, to analyze and design a software system. Current techniques, notations, methods, processes and tools used for system analysis and design using UML modelling are also included.

9. **Course attributes:**

   General education component: ___ N/A ____

   ___ Cultural diversity ___ Honors ___ Writing centered ___ Writing intensive ___Writing active

10. **Instructional delivery**
    Type of Course:

    ___ Lecture ___ Lab _X__ Lecture/lab combined ___ Independent study/research

    ___ Internship ___ Performance ___ Practicum/clinical ___ Other, specify: ________________

    Mode(s) of Delivery:

    _X__ Face to Face _X__ Online ___ Study Abroad

    _X_ Hybrid, specify approximate amount of on-line and face-to-face instruction___~ 51% Face-to-face and 49% online ________________

11. Course(s) to be deleted from the catalog once this course is approved. ___NONE_______________

12. **Equivalent course(s):** _____N/A_

    a. **Are students allowed to take equivalent course(s) for credit?** ___ Yes _X__ No

13. **Prerequisite(s): _TEC 5373 or equivalent or permission of instructor__**

    a. **Can prerequisite be taken concurrently?** ___ Yes _X__ No

b. Minimum grade required for the prerequisite course(s)? _C__

c. Use Banner coding to enforce prerequisite course(s)? _X_ Yes ___ No

d. Who may waive prerequisite(s)?

___ No one _X__ Chair _X__ Instructor _X_ Advisor ___ Other (specify)

14. **Co-requisite(s):** ___NONE____

15. **Enrollment restrictions**

a. Degrees, colleges, majors, levels, classes which <u>may</u> take the course: _

Graduate students in Technology OR
permission of instructor or chair

b. Degrees, colleges, majors, levels, classes which may <u>not</u> take the course: __ ALL OTHERS __________

16. **Repeat status:** _X__ May not be repeated ___ May be repeated once with credit

17. **Enter the limit, if any, on hours which may be applied to a major or minor:** ___

18. **Grading methods:** _X__ Standard ___ CR/NC __ Audit ___ ABC/NC

19. **Special grading provisions:**

___ Grade for course will <u>not</u> count in a student's grade point average.

___ Grade for course will <u>not</u> count in hours toward graduation.

___ Grade for course will be removed from GPA if student already has credit for or is registered in: ____

___ Credit hours for course will be removed from student's hours toward graduation if student already has credit for or is registered in: _______________________________________

20. **Additional costs to students:**
Supplemental Materials or Software____

Course Fee _X__No ___Yes, Explain if yes____________________________________

21. **Community college transfer:**

___ A community college course may be judged equivalent.

_X__ A community college may <u>not</u> be judged equivalent.

Note: Upper division credit (3000+) will <u>not</u> be granted for a community college course, even if the content is judged to be equivalent.

Rationale, Justifications, and Assurances (Part I)

1. ___Course is required for the major(s) of ____

___Course is required for the minor(s) of _____________________

___Course is required for the certificate program(s) of _______________

_X__ Course is used as an elective in Masters of Technology

2. Rationale for proposal :

The purpose of this course is to introduce graduate students to system analysis and design to model complex software related projects. Effective communication within computer technology domain has become necessary with ever changing user requirements and increased use of multi-vendor systems. Also within the realms of software design comes increased pressure for managers to design, architect and develop solutions for not only new projects but to incorporate these in existing projects. This course allows modeling and communication for this purpose.

Graduate students in Technology are required to take a minimum of 19 hours of elective courses numbered 5000 and above. Currently, we offer fewer courses at 5000 and above in Computer domain in the graduate program of Technology

Students in TEC 5853 will be held to higher performance standards in all facets of the course. Additional requirements for students in TEC 5853 will include: (1) Homework will contain higher level analytical questions requiring primary literature review. (2) Research Paper will be required of the graduate class, and none is required in the undergrad class (3) Homework will also be held to a higher standard for their research analysis, literature review, writing style and maturity of thought. (4) Students in TEC 5853 will be assigned to lead in-class discussions.

3. Justifications for (answer N/A if not applicable)

Similarity to other courses: N/A

Prerequisites*: Material in this course is built on the knowledge of TEC 5373 or equivalent, and uses their concepts and knowledge as a foundation for this course TEC 5853.*

Co-requisites: N/A

Enrollment restrictions: *Graduate students in Technology OR permission of instructor or chair*

Writing active, intensive, centered: N/A

4. General education assurances (answer N/A if not applicable)

General education component: N/A

Curriculum: N/A

Instruction: N/A

Assessment: N/A

5. **Online/Hybrid delivery justification & assurances (answer N/A if not applicable)**

Online or hybrid delivery justification:

The content and structure for this course relies upon independent research, in-depth group discussion, and video based lecture. As compared to many lab courses already offered in technology area, this course requires online delivery of lecture and discussion and face-to-face lab activities for applied projects. For content delivered online, the course employs online video presentations, structured web discussions focused on reading assignments, and linked to articles submitted to the instructor. Students are required to draw on research and review of articles to discuss and develop fundamental procedural knowledge of application. Discussions invite students to explore in more detail the required knowledge and procedures to analyze and design systems. Discussions and examinations will be administered and submitted via the online course management tool. With the current technology, many software design companies have made their software tools more readily accessible for students. The Internet connection speed for many users has increased thereby allowing for higher quality rich media instruction to be delivered. Finally, the course management tools that the university now uses allows there to be a richer interaction between students and faculty. To accommodate this situation, many of the given activities may be completed in a hybrid format.

Instruction:

This course employs instructor led online presentations, student reading assignments, student applied design assignments, peer critique and troubleshooting, student presentations, and examinations. After reviewing the instructor led presentations and completing the student reading assignments, students will be required to draw on what they have read and then to apply it to a context of analyzing and designing software systems. While working on these projects, students may engage in the activity of troubleshooting or critique while posting their work in an online discussion board for both classmates and the instructor to provide feedback and guidance. Presentations will provide learners a forum to share the results of their work and receive further feedback. Reading assignments, applied projects, and examinations will be administered, collected, and/or submitted via the online course management tool. Presentations may also be delivered in the course tool or face-to-face. All faculty who will deliver this course online are/will be OCDi (or appropriate equivalent) trained.

Integrity:

Work submitted online, such as discussions and examinations, will be substantiated via learners providing citation in APA format and submitting related articles to quantify work. Further, the length, frequency, quality, and integrity of discussion posts can be monitored via the online course management tool. Examinations will require the same of learners and additionally will use software tools, such as "TURNITIN", to check work for the integrity and authenticity of submitted assignments. The examinations will be time restricted and of sufficient length to prohibit consultation of unauthorized sources. Work submitted face-to-face in applied lab projects will be checked for authenticity via the individualized nature of project completion. Requirements for projects will *require learners to engage in activities that require creation of original content for either themselves or local entity.*

Interaction:

For online content, the course employs email, web-based discussions, exploration of off-site Internet resources, web-based presentations, web chat rooms and lab based applied project work. The instructor will communicate with students through the online discussion board and web-based discussions. Email may also be a tool used for the instructor to communicate with an individual student or to post course announcements. The learners for this course may also communicate with one another for these tools. During digital office hours, the instructor will remain available for discussion during certain times and communicate using a chat room tool in the learning management system. For face-to-face interaction, the instructor may communicate synchronously with the learners during open lab activities and during office hours. The learners are also free to communicate with other learners during lab activities.

Model Syllabus (Part II)

Please include the following information:

1. Course number and title
TEC 5853 – System Analysis & Design

2. Catalog description
This course provides introduction to software engineering principles, software development life-cycles, and modelling in software engineering. Current techniques, notations, methods, processes and tools used for system analysis and design using UML modelling are also included.

3. Learning objectives.
Upon completion of this course, students will be able to:

a) Design and evaluate a solution to a problem using the various phases of software engineering, including requirement elicitation and analysis, system analysis and design (Grad 1,2, 3)
b) Design and develop a software traversing through these phases; (Grad 2,3)
c) Create a model-based software using UML (Grad 1,2,3)
d) Generate and support requirements from the customer, thereby composing software requirements under the form of a use case model, using established principles; (Grad 1-3)
e) Develop analysis models made of consistent diagrams (class, sequence, state machine diagrams), following well established heuristics; (Grad 2,3)
f) Design, develop and demonstrate a project using the knowledge gained (Grad 1-4)

Graduate Learning Goals

Objective	Depth of content knowledge	Critical thinking and problem solving	Oral and/or written communication	Advance scholarship through research and creative activity
(a)	X	X	X	
(b)		X	X	
(c)	X	X	X	
(d)	X	X	X	
(e)		X		
(f)	X	X	X	X

4. Course materials.
This will include lecture notes, online resources (such as online tutorials, research papers), etc as well as the following book:

> Bernd Bruegge and Allen Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, Third Edition, Prentice-Hall 2009, ISBN 0-13-606125-7.

5. Weekly outline of content.

Face-to-Face / Online Modality:

Week	**Topics**	**Activities**
Week 1	Introduction to Software Engineering: the nature of software, history and scope of software engineering, relationships with other fields, fundamental principles, software life cycle	
Week 2-3	Various Software Life cycle methodologies	Lab 1
Week 4	Introduction to UML	
Week 5	Use cases modeling	Lab 2
Week 6	Class diagram modeling	Lab 3
Week 7	Object diagram modeling	Lab 4
Week 8	Sequence diagram modeling	Midterm
Week 9	Activity diagram modeling	Lab 5
Week 10	State-base behavior modeling	Lab 6
Week 11 – 12	Object Oriented Analysis using the UML. Producing an analysis model that the developers can unambiguously interpret. Formalizing the requirements (requirement elicitation) into specifications.	Lab 7
Week 13	System Design using the UML - design patterns	Final Project
Week 14	Design patterns (continued)	Lab 8
Week 15	Non-Functional Requirements	Lab 9
Week 16	Final Exam	

Hybrid Modality:

Week	**Topics**	**Activities**
Week 1	Introduction to Software Engineering: the nature of software, history and scope of software engineering, relationships with other fields, fundamental principles, software life cycle	
Week 2-3	Various Software Life cycle methodologies	Lab 1
Week 4	Introduction to UML	
Week 5 Face to Face Meetings 8 am to 5 pm	Question/Answers/Review Sessions of Weeks 1 to 4 Use cases modeling	Lab 2
Week 6	Class diagram modeling	Lab 3
Week 7	Object diagram modeling	Lab 4
Week 8	Sequence diagram modeling	Midterm
Week 9 Face to Face Meetings 8 am to 5 pm	Question/Answers/Review Sessions of Weeks 6 to 8 Activity diagram modeling	Lab 5
Week 10	State-base behavior modeling	Lab 6
Week 11 -12	Object Oriented Analysis using the UML. Producing an analysis model that the developers can unambiguously interpret.	Lab 7

	Formalizing the requirements (requirement elicitation) into specifications.	
Week 13	System Design using the UML - design patterns	Final Project
Week 14 Face to Face Meetings 8 am to 5 pm	Question/Answers/Review Sessions of Weeks 10 to 13 Design patterns (continued)	Lab 8
Week 15-16	Non-Functional Requirements	Lab 9 / Final Exam

6. Assignments and evaluation, including weights for final course grade.

	Graduate
Lab Assignments	40 %
Class Participation	5 %
Exams	35 %
Class Projects	10 %
Research Paper	10%
Total	100%

7. Grading scale.
A = 90% or above, B = 80 – 89%, C = 70 – 79%, D = 60 – 69%, F = Below 60%

8. Correlation of learning objectives to assignments and evaluation.

Objective	Assignments (40)	Class Participation (5%)	Projects (10%)	Midterm (15%)	Final (20%)	Research Paper (10%)
a) Design and evaluate a solution to a problem using the various phases of software engineering, including requirement elicitation and analysis, system analysis and design (Grad 1,2, 3)	X	X	X	X		X
b) Design and develop a software traversing through these phases; (Grad 2,3)	X	X	X	X		X
c) Create a model-based software using UML (Grad 1,2,3)	X	X	X	X	X	X
d) Generate and support requirements from the customer, thereby composing software requirements under the form of a use case model, using established principles;	X		X		X	X

(Grad 1-3)						
e) Develop an analysis models made of consistent diagrams (class, sequence, state machine diagrams), following well established heuristics; (Grad 2,3)	X		X		X	X
f) Design, develop and demonstrate a project using the knowledge gained (Grad 1-4)			X			X

Date approved by the department or school: Feb 1, 2018
Date approved by the college curriculum committee: Feb 20, 2018
Date approved by the Honors Council *(if this is an honors course):*
Date approved by CAA: CGS: