DO NOT TYPE IN THIS BOX

Bulletin #: _____

Academic Year: _____

FLORIDA INTERNATIONAL UNIVERSITY
UNIVERSITY CURRICULUM COMMITTEE
Proposal for a Course Change

PART I. FILL OUT THIS SECTION COMPLETELY

1. School/College Engineering and Computing

   Div./Dept. in Which Taught School of Computing and Information Sciences

2. | CEN | 4 | 010 | | 3 |
   |---|---|---|---|---|
   | Alpha Prefix | 1st Digit | Last 3 Digits | "C"-lec-lab "L"-Lab | Cr. Hrs. |

3. Present Course Title Software Engineering I

PART II. FILL OUT CHANGE INFORMATION ONLY

Change Effective 1 / 6 / 2020

4a. New Course Title _____

b. New Abbreviated course Title *(for computer class schedules, transcripts)* _____

LIMITED TO 25 Characters *(including spaces)*

5a.

New Alpha Prefix	New 1st Digit	New Last 3 Digits	Change "C"-lec-lab "L"-Lab

5b. Change Credit Hours: From _____ To _____

6. New Catalog Description/Major Topics *(not to exceed 200 characters including spaces)*
   College of Medicine and College of Law: Attach description not exceeding 1,000 characters including spaces.

7. New Prerequisite(s): CGS 3095 and COP 3337

8. New Corequisite(s): None

9. Explain Reclassification Request:

   Current "Prereq: CGS3095 and COP3530 and COP4710 & Coreq:COP4710" are not essential. This new prereq and no coreq requirement reduces prereq chain dependency and accelerates 4-year graduation rate

10. Does this proposed change impact the assessment process of a program or certificate? If yes, then send notification to assessment@fiu.edu.

PROPOSAL REQUESTED BY:

Faculty Contact Nagarajan Prabakar (Type name) — Prabakar (Signature) — 3 / 21 / 2019

prabakar@cis.fiu.edu (Email address) — 305-348-2033 (Phone number)

Chairperson (Dept./Div.) S.S. Iyengar (Type name) — (Signature) — 3 / 21 / 2019

Chairperson (Curr. Comm.) Cesar Levy (Type name) — (Signature) — ____ / ____ / 2019

College/School Dean John Volakis (Type name) — (Signature) — ____ / ____ / 2019

Submit one original form. Attach one copy of the course justification and course syllabus: course description, objectives, learning outcomes, major topics and textbooks.

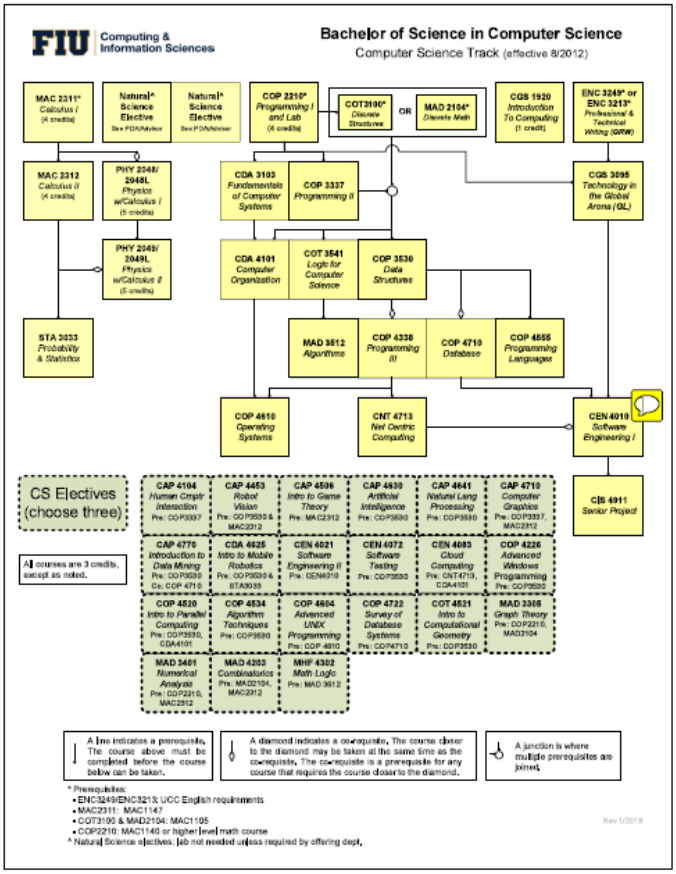
CEN-4010 Software Engineering I

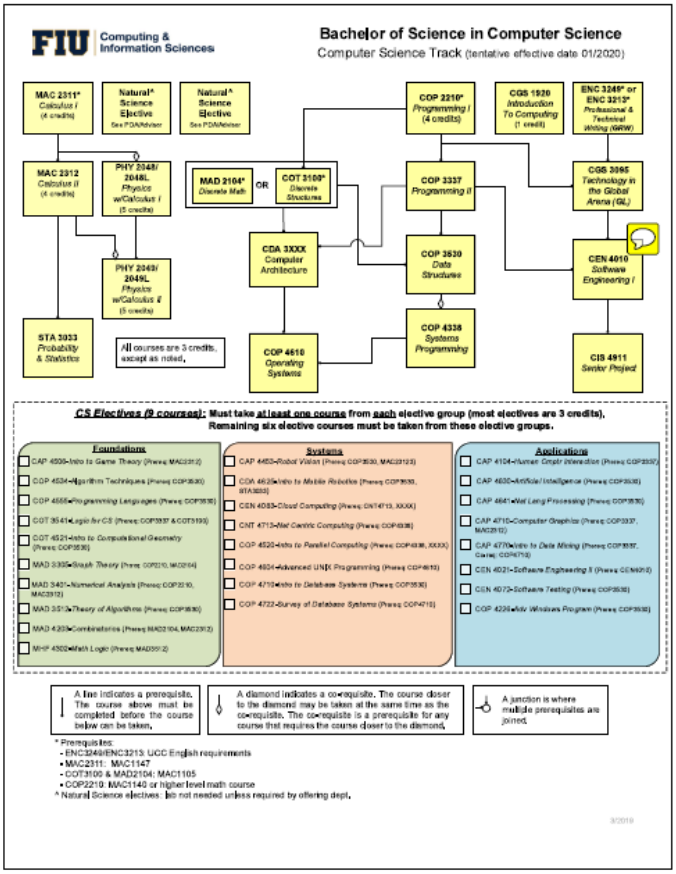
Course Change Justification

Software-Engineering-I course requires that students must have completed two levels of programming courses (COP-3337) and some oral and written communication skills (CGS-3095).

However, the current “Prereq: CGS3095 and COP3530 and COP4710 & Coreq: COP4710" requirement increases the length of the prerequisite chain of the degree program as in the left flowchart (present curriculum model) below. Since this requirement unnecessarily extends student graduation period, this course change proposal specifies only the essential courses as “Prereq: CGS3095 and COP3337 & Coreq: None” as shown in the right flowchart (restructured new curriculum model) below.

This course change will enable students to expedite the graduation time by allowing them to enroll in CEN-4010 during junior year and will increase the 4-year graduation rate.





School of Computing and Information Sciences

Course Title: Software Engineering I **Date:** March 18, 2019

Course Number: CEN 4010

Number of Credits: 3

Subject Area: Software Engineering	**Subject Area Coordinator:** Peter Clarke **email:** clarkep@cis.fiu.edu
Catalog Description: Software Process Model, Software Analysis and Specification, Software Design, Software Testing	
Textbook: Bernd Bruegge, Allen H Dutoit, "Object-Oriented Software Engineering: Using UML, Patterns, and Java", 3rd Edition, Prentice Hall, ISBN 0136061257.	
References:	
Prerequisites Courses: CGS 3095 and COP 3337	
Corequisites Courses: None	

Type: Required for CS Major

Prerequisites Topics:

- Programming
- Data Structures
- Oral and written communication skills

Course Outcomes:

1. Be familiar with the Software Development Life Cycle
2. Master the techniques to gather and specify the requirements of a medium-size software system using UML
3. Master the techniques to design and implement a medium-size software system
4. Be familiar with software testing techniques
5. Be familiar with system walkthroughs
6. Be familiar with software documentation
7. Be familiar with working in a small software development team
8. Demonstrate the ability to communicate the details of the technical solution through verbal and written modes.

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Relationship between Course Outcomes and Program Outcomes

BS in CS: Program Outcomes	Course Outcomes
a) Demonstrate proficiency in the foundation areas of Computer Science including mathematics, discrete structures, logic and the theory of algorithms	
b) Demonstrate proficiency in various areas of Computer Science including data structures and algorithms, concepts of programming languages and computer systems.	3, 4
c) Demonstrate proficiency in problem solving and application of software engineering techniques	1, 2, 3, 4, 5, 6, 7
d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.	
e) Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.	
f) Demonstrate the ability to work cooperatively in teams.	2, 5, 7
g) Demonstrate effective communication skills.	2, 5, 6, 7

Assessment Plan for the Course & how Data in the Course are used to assess Program Outcomes

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan: http://www.cis.fiu.edu/programs/undergrad/cs/assessment/

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Outline

Topic	Number of Lecture Hours	Outcome
• Introduction to Software Engineering o Concepts o Life Cycle Model o Products o Reviews o Development Team	6	1, 5, 6, 7
• Software Modeling o Concepts o Modeling with UML	6	1, 2, 6
• Requirement Gathering and Analysis o Concepts and Activities o Functional Requirement ▪ Scenarios and Use Cases o Non-functional requirements o Requirement Validation	12	1, 2, 5, 6, 7
• Software Design o System Design ▪ Design Goals ▪ Cohesion and Coupling ▪ Persistent Data ▪ Access Control o Object Design ▪ Object Interface ▪ Invariants ▪ Pre and post conditions	12	1, 3, 5, 6, 7
• Testing o Testing Concepts o Test Planning o Unit Testing o Integration Testing o Usability Testing	6	4, 5

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Course Outcomes Emphasized in Laboratory Projects / Assignments

Outcome	Number of Weeks
1. Software Requirement and Analysis Model Outcomes: 1,2,6,7	4
2. Software Design Document Outcomes: 1,3,6,7	4
3. Final Software Project Demonstration Outcomes: 1,2,3,4,5,6,7	4

Oral and Written Communication:

Written Reports		Oral Presentations	
Number Required	Approx. Number of pages for each	Number Required	Approx. Time for each
3 (Software Requirement, Design Document and Final System Document)	30	2	15 minutes per group (5 minutes per student)

Social and Ethical Implications of Computing Topics:

No significant coverage

Topic	Class time	Student Performance Measures

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Approximate number of credit hours devoted to fundamental CS topics

Topic	Core Hours	Advanced Hours
Algorithms:		
Software Design:		2.0
Computer Organization and Architecture:		
Data Structures:		
Concepts of Programming Languages:		

Theoretical Contents

Topic	Class time
Invariants, pre and post conditions	1.0

Problem Analysis Experiences

Software requirement and analysis model

Solution Design Experiences

1. System Design using Architectural Patterns
2. Detailed Object design using Design Patterns

School of Computing and Information Sciences
CEN 4010
Software Engineering I

The Coverage of Knowledge Units within Computer Science Body of Knowledge[1]

Knowledge Unit	Topic	Lecture Hours
SE 1	Fundamental Design concepts and principles, Software Architecture, Object-Oriented Design	12
SE 4	Software Life-Cycle and Process Models	6
SE 5	Requirement Elicitation, Requirements Analysis Modeling Techniques, Functional and Nonfunctional requirements, Basic Concepts of Formal specification techniques	12
SE 6	Validation Planning, Testing Fundamentals, Black-box and White-box testing, Unit, integration, validation and system testing, Object-Oriented Testing, Inspections	6
SE 8	Team Management, Software measurement and scheduling techniques, Project management tools	2
SE 10	Pre and post assertions	1

[1]See http://www.computer.org/education/cc2001/final/chapter05.htm for a description of Computer Science Knowledge units