

School of Computer Science

Course Title: Software Engineering I

Date: December 1, 2003

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: Bruegge and Dutoit, Object-Oriented Software Engineering: Using UML, Patterns, and Java, Prentice Hall, 2004. ISBN: 013471100	
References:	
Prerequisites Courses: COP 3530	
Corequisites Courses: None	

Type: Required

Prerequisites Topics:

- Programming
- Data Structures

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

School of Computer Science
CEN 4010
Software Engineering I

Outline

Topic	Number of Lecture Hours	Outcome
<ul style="list-style-type: none"> • Introduction to Software Engineering <ul style="list-style-type: none"> ○ Concepts ○ Life Cycle Model ○ Products ○ Reviews ○ Development Team 	6	1, 5, 6, 7
<ul style="list-style-type: none"> • Software Modeling <ul style="list-style-type: none"> ○ Concepts ○ Modeling with UML 	6	1, 2, 6
<ul style="list-style-type: none"> • Requirement Gathering and Analysis <ul style="list-style-type: none"> ○ Concepts and Activities ○ Functional Requirement <ul style="list-style-type: none"> ▪ Scenarios and Use Cases ○ Non-functional requirements ○ Requirement Validation 	12	1, 2, 5, 6, 7
<ul style="list-style-type: none"> • Software Design <ul style="list-style-type: none"> ○ System Design <ul style="list-style-type: none"> ▪ Design Goals ▪ Cohesion and Coupling ▪ Persistent Data ▪ Access Control ○ Object Design <ul style="list-style-type: none"> ▪ Object Interface ▪ Invariants ▪ Pre and post conditions 	12	1, 3, 5, 6, 7
<ul style="list-style-type: none"> • Testing <ul style="list-style-type: none"> ○ Testing Concepts ○ Test Planning ○ Unit Testing ○ Integration Testing ○ Usability Testing 	6	4, 5

School of Computer Science
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:

Number of written reports: 2 (Software Requirement and Design Documents)

Approximate number of pages for each report: 30

Number of required oral presentations: 2

Approximate time for each presentation: 15 minutes per group (5 minutes per student)

Social and Ethical Implications of Computing Topics:

Topic	Class time	Student Performance Measures
Consequences of software failure	Throughout the semester	Not measured

**School of Computer Science
CEN 4010
Software Engineering I**

Approximate number of credit hours devoted to fundamental CS topics

Topic	Core Hours	Advanced Hours
Algorithms:		
Software Design:		0.1
Computer Organization and Architecture:		
Data Structures:		
Concepts of Programming Languages:		

Theoretical Contents

Topic	Class time
Invariants, pre and post conditions	1.5

Problem Analysis Experiences

Software requirement and analysis model

Solution Design Experiences

Software design document

School of Computer Science
CEN 4010
Software Engineering I

The Coverage of Knowledge Units within Computer Science Body of Knowledge¹

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	1.5
SE 8	pre and post assertions	1.5

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