

**SOUTHERN UNIVERSITY
UNIVERSITY CURRICULUM COMMITTEE**

COURSE SYLLABUS

**Software Engineering
Southern University and A&M College
Baton Rouge, LA 70813**

I. Descriptive Information

A. Course Number: 378

B. Course Title: Software Engineering

C. Catalog Description: The application of engineering principles to the production of correct and reliable software. Topics include the theory and practice of software development, the software crisis, human-computer interaction, computer-aided software engineering (CASE), cost estimation, maintenance, object-oriented software development.

D. Instructor's Statement of Course Description: Designing and writing a correct program is only the first step in a long process in the life-cycle of a software product. This course is intended to present the processes involved in ensuring a quality product. Testing methods are key to quality assurance, as well as design and programming techniques that facilitate software maintenance. We shall investigate verification and validation issues, and apply these principles to actual projects. Familiarity with at least one object-oriented language (preferably C++ or Java) is assumed.

E. Credit Hours: 3.0

F. Prerequisites: CMPS 300

G. Intended Audience: Computer Science Majors.

H. Instructors: Mrs. Jessie M. Bethly-Betz
Office: Room 110A, Allain
Phone: 771-2060
Email: bethlybj@cmps.subr.edu

II. Specification of Course Goals and Objectives

A. Statement of General Goals: The general goals of the course are:

1. Understand the software crisis.
2. Appreciate the importance of software maintenance.
3. Test, validate, and verify a software product, using standard industry metrics.
4. Design and develop correct and robust software products, going beyond freshman and sophomore levels of programming.
5. Apply CASE tools to facilitate timely software production.
6. Understand business requirements pertaining to software development.

B. Statement of Course Objectives: Upon course completion, students will have demonstrated through projects, reading assignments, presentations, and examinations, enhanced ability beyond that expected from freshman and sophomore students. They will be able to:

1. Plan and execute a successful software development project.
2. Design, construct, and test software products for current use and for future maintenance.

3. Contribute meaningfully in team software development, as reflected in peer assessments.
4. Document clearly and concisely all stages of the development of a software product.
5. Demonstrate accountability in requirements fulfillment through project performance.

C. Statement of Course Content: This course will be project-driven. Throughout the semester, students will be assigned small groups (3 to 4 persons each) to develop small software systems. They are required to document every stage of the process. Concurrently, software engineering concepts and issues will be presented and discussed. At the end of the semester, teams shall present their projects in a formal setting.

III. Readings

A. Required Textbook:

Pressman, Roger S. **Software Engineering: A Practitioner's Approach** 6th Edition. McGraw-Hill Higher Education. ISBN: 0072853182 © 2005 Price: \$114.75

B. General Reading Sources:

Schach, Stephen R. *Classical and Object-Oriented Software Engineering: With UML and C++*. Boston, MA: McGraw-Hill. .

Sommerville, Ian. *Software Engineering*. Reading, MA: Addison-Wesley. .

Pfleeger, S. L. *Software Engineering: Theory and Practice*. Upper Saddle River, NJ: Prentice-Hall.

Raskin, Jef. *The Humane Interface: New Directions for Designing Interactive Systems*. Boston: Addison-Wesley.

Harmelen, Mark van, editor. *Object Modeling and User Interface Design: Designing Interactive Systems*. Boston: Addison-Wesley.

Eriksson, H. E., and M. Penker. *UML Toolkit*. New York: Wiley.

Braude, E. J. *Software Engineering: An Object-Oriented Perspective*. New York: Wiley.

C. Reference Journals:

ACM Computing Surveys.
 ACM Software Engineering Notes.
 Communications of the ACM.
 Human-Computer Interaction.
 IEEE Computer.
 IEEE Software.
 IEEE Trans. on Software Engineering.
 SIGPLAN Notices.

IV. Instructional Procedure

Instruction in this course consists of an ongoing team project, formal lectures, discussions, and exams. All team participants are required to give their full effort.

V. Course Requirements

A. Academic Requirements:

Students are required to take the quizzes, examinations, as well as to present the semester project, according to the requirements and deliverables.

B. Administrative Requirements:

Students are expected to attend all class sessions, take all quizzes and examinations, and submit all assignments and make all presentations by the due date. The Computer Science Department of Southern University places heavy emphasis on regular attendance.

VI. Course Schedule

Week	Topic
1.	Fundamentals of software engineering. Software life-cycle.
2.	Requirements and specifications. Formal documentation.
3.	Prototyping. Brief introduction to Z. Risk analysis.
4.	Design. Procedural architectural design and user-interface issues.
5.	Object-oriented architectural design.
6.	Real-time system design. First project progress presentation.
7.	Mid-term Exam.
8.	Implementation and integration.
9.	Verification and validation. Testing metrics.
10.	Second project progress presentation.
11.	Black-box, glass-box testing. Walkthroughs.
12.	Software maintenance. Configuration management. Software quality control.
13.	Corrective, adaptive, and enhancement maintenance processes.
14.	Final project presentations.
15.	Final Exam.

VII. Evaluation of Students

Attendance/participation	20%
Assignments	20%
Presentation	20%
Mid-term Exam	20%
Final Exam	20%

VIII. Grading

The grading scale is:

90-100	A
80-89	B
70-79	C
60-69	D
Below 60	F