BOWIE STATE UNIVERSITY
Syllabus
Department of Computer Science

COSC 475 /565 (3 Cr) **Software Engineering I**
Fall Semester 2018 Course Information

Instructor:	Sharad Sharma
Office Location:	Computer Science Building Room 317
Phone:	301-860-4502
Email:	ssharma@bowiestate.edu
Class Hours:	Tuesday: 4:55 PM to 7:25 PM
Office Hours:	Tuesday: 2:55 PM to 4:55 PM or by appointment
COURSE WEBSITE:	http://www.cs.bowiestate.edu/sharad/software/

COURSE DESCRIPTION

This course introduces the student to major topics in software engineering such as: requirements specification, analysis and design, testing, project management, and implementation. Additional topics such as software life cycle models, the Unified Modeling Language (UML), agile software development techniques, configuration management, change control, and project documentation will be discussed.

COURSE PREREQUISITE

COSC 214 or COSC 504

REQUIRED TEXTS

Somerville, Ian (2001) Addison-Wesley *Software Engineering 9th Edition, Massachusetts: Addison Wesley,* ISBN-10: 0137035152, ISBN-13: 978-013703515

PROGRAM OUTCOMES (PO):

This course is required for all computer science major students and has significant relationship with the following program outcomes:
1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions. [CS]

STUDENT COURSE LEARNING OBJECTIVES (SCLO)

The essential objectives for this course are to:

1. Illustrate proficiency in developing and producing software process artifacts, most importantly the code and user documentation.
2. Demonstrate and acquire skills that help you work effectively as a member of a software development team.
3. Describe the software life cycle, roles, artifacts, and activities.
4. Identify and understand the concepts of software "best practices" and when they apply.
5. Demonstrate that you are able to adapt a process to your needs and select an appropriate set of best practices that will guide you in completing a software development project.

STUDENT EXPECTED OUTCOMES

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

1. Define and understand the role and responsibilities of a software engineer
    - Instruments: Quiz 1, Assignment1, Exam
    - Covers PO (1), (2), (4), and SCLO (1), (2)
2. Define Project management and its role in systems planning, systems analysis, systems design, systems implementation, and systems support.
    - Instruments: Quiz 1, Assignment1, Exam
    - Covers PO (2), (6) and SCLO (1), (2), (3)
3. Describe and define a feasibility plan, requirements and design documentation.
    - Instruments: Quiz 1, 2, Assignment1
    - Covers PO (1), (2) and SCLO (1), (2)
4. Define and perform data modeling and process modeling, and explain why they are important.
    - Instruments: Quiz 2, Exam
    - Covers PO (b), (c) and SCLO (1), (2), (4)
5. Design, develop, test, and demonstrate a major piece of software.
    - Instruments: Project, Assignment 2
    - Covers PO (3), (5), (6), and SCLO (1), (2), (5)
6. Demonstrate the fundamental principles of software engineering. Be able to identify and describe the software life cycle, roles, artifacts, and activities.
    - Instruments: Quiz 2, Exam
    - Covers PO (1), (2) and SCLO (1), (2), (3)
7. Demonstrate that you are able to adapt a process to your needs and select an appropriate set of best practices that will guide you in completing a software development project.
    - Instruments: Quiz 2, Assignment1, Exam
    - Covers PO (2), (6) and SCLO (1), (2), (4), (5)

TEACHING MODES

All course material will be provided on a course web site including lecture notes, useful links on the web, recommended references, time schedule, and contact information for faculty, guidelines for projects, coding standards, and more. The primary teaching mode will be lecture and discussion.

COURSE REQUIREMENTS AND EXPECTATIONS

Policy on Attendance: Regular attendance in the class is mandatory. Students will be responsible for any loss of information, assignments, and projects due to absence from class.

Departmental Policy on Submission of Late Work: There will be no make-up for any missed classes, projects, assignments, and exams. 1/2 letter grade off for assignment each day late without documented excuse; papers more than one week late will not be accepted.

Academic Integrity: Academic dishonesty includes plagiarism, cheating, and other illegal or unethical behaviors in doing the work of the course. Plagiarism is the act of representing another's ideas, words or information as one's own. If you receive assistance on an assignment from someone else, you must avoid plagiarism by giving proper credit for this assistance. Include in your assignment a comment naming the person who assisted you and stating what the assistance was. Students who are guilty of academic dishonesty are subject to severe penalties ranging from a reduction in points (and possible failure) for the assignment/project, to failing the course, or in extreme cases, dismissal from the University. Do not copy other student's projects, codes, and design. A group of students working together on a project must change their forms and codes to differentiate from others.

EVALUATION: Following is the Evaluation system for the Final Grade. Each assignment will be graded. Students are responsible for completing them as scheduled.

1. Two Assignments	20%
2. Quizzes (2)	10%
3. Mid-Term Exam	20%
4. Final Exam	20%
5. Final Project	30%

Final Project, Mid-term and Final exams are mandatory.
Assignments: One assignment given prior to the mid-term exam and one assignment given after the mid-term exam.
Seminar (for undergraduate students) will involve attending the department seminar and writing a two page essay on it. It will be 2% of the grade.

Final Project: The purpose of the course project is to provide the students with the knowledge of software engineering methodology and the skills to apply it. The project

consists of **two iterations**, both focused around the *same* software product. The first iteration is *exploratory* and represents the first attempt at developing the proposed software product. The second *iteration* is development and also includes revision of the project goals. The deliverables for the **first and second iteration** are reports and demos.

GRADING: Academic dishonesty will result in grade F. The following grade scale will be used:

90 % - 100% = A
80 % - 89% = B
70 % - 79% = C
60 % - 69% = D
0 - 59% = F

Final grades will be computed based upon credits earned for all the five components mentioned above.

COURSE/TOPICAL OUTLINE

Week 1. Introduction: Ethics, ACM/IEEE code of ethics, professional software development.

Week 2. Software processes: Process models, process activities, software design activities, prototyping, RUP

Week 3. Agile software development: Plan-driven and agile development, extreme programming, refactoring, scrum, scaling up and scaling out

Week 4. Requirements engineering: requirement types, functional and non-functional requirements, requirements engineering processes and specification, elicitation and analysis, requirement document and management

Week 5 & 6. System modeling: UML diagram types, aggregation, inheritance, aggregation, composition, multiplicity, composition, generalization, association, context models, interaction models, structural models, behavioral models, model-driven engineering

Week 7. Architectural design: design decisions, abstraction, views, architectural patterns, system quality attribute, and application

Week 8 & 9. Design and Implementation: Object-oriented design using the UML, design patterns, implementation issues, open source development and licensing, object-oriented design process, configuration management

Week 10 & 11. Software testing & Unit Testing: Test types, development testing, test-driven development, release testing, user testing

Week 12 & 13. Software Evolution, Dependability and Security: Evolution processes, program evolution dynamics, software maintenance, reengineering, legacy system management, security, risk management

Week 15. Object Oriented Design: objects, cohesion, coupling, data encapsulation, abstraction, information hiding, polymorphism, dynamic binding.

Reminder: English Proficiency Examination
After successfully completing ENGL 101 and 102, Composition and Literature I and II, students must take and successfully pass the Bowie State University English Proficiency Examination. Transfer students who completed their English composition requirements at another university should take the English Proficiency Examination during their first semester of enrollment at Bowie State University.

ADA Statement:
Students with disabilities who wish to receive ADA accommodations should report to the Office of Special Populations, Center for Learning and Technology (CLT) building, Suite 302 (301-860-3292).

Students who have a disability and want accommodations should report immediately to **Disability Support Services** (DSS), located in Room 1328 in the Business and Graduate Studies Building or call Mr. Michael S. Hughes, DSS Coordinator, at 301-860-4067.

SELECTED BIBLIOGRAPHICAL REFERENCES

1. Braude, Eric J. (2001). *Software Engineering An Object-Oriented Perspective*, John Wiley & Sons, Inc., ISBN 0-471-32208-3.
2. Barnes D. J., and Kölling, M. (2003) *Objects First With Java*, Prentice Hall,ISBN 0-13-044929-6.
3. Bruegge, B., and Dutoit, A. (2000). *Object-Oriented Software Engineering Conquering Complex and Changing Systems,* Prentice-Hall, ISBN 0-13-489725-0.
4. Bruegge, B., and Dutoit, A. H. (2004). *Object-Oriented Software Engineering: Using UML, Patterns and Java,* Second Edition, Prentice Hall, ISBN 0-13-0471100
5. David, K. (1998). *The Art of Computer Programming*, V. 1-3, 2nd ed. Boxed Set, Addison-Wesley, ISBN: 0201485419.
6. Ghezzi, J., and Mandrioli, P. (1991). *Fundamentals of Software Engineering* by Hall.