UMBC UGC New Course Request: IS 472: Software Security

Date Submitted: 12/15/15 Proposed Effective Date: Fall 2016

	Name	Email	Phone	Dept
Dept Chair or UPD	Carolyn Seaman	cseaman@umbc.edu	53937	IS
Other Contact	Vandana Janeja	vjaneja@umbc.edu	56238	IS

COURSE INFORMATION:

Course Number(s)	IS 472
Formal Title	Software Security
Transcript Title (≤30c)	Software Security
Recommended Course Preparation	
Prerequisite **NOTE:** Unless otherwise indicated, a prerequisite is assumed to be passed with a “D” or better.	IS 247 with C or better
Credits	3
Repeatable?	☐ Yes ☒ No
Max. Total Credits	3 This should be equal to the number of credits for courses that cannot be repeated for credit. For courses that may be repeated for credit, enter the maximum total number of credits a student can receive from this course. E.g., enter 6 credits for a 3 credit course that may be taken a second time for credit, but not for a third time. Please note that this does NOT refer to how many times a class may be retaken for a higher grade.
Grading Method(s)	☒ Reg (A-F) ☐ Audit ☐ Pass-Fail

PROPOSED CATALOG DESCRIPTION (no longer than 75 words**):**

Many cybersecurity attacks are facilitated by software vulnerabilities, i.e. characteristics of software source code that unintentionally allow unauthorized access to computer memory. Future cybersecurity professionals require an understanding of various techniques that can be applied throughout the software development lifecycle to prevent, detect, and remove such vulnerabilities. Through this course students will: 1) Develop an understanding of common vulnerabilities and emerging attacks; 2) Learn how to apply secure coding standards and techniques to ensure that their source code is as free from vulnerabilities as possible; 3) Be provided hands-on experience in handling software vulnerabilities.

RATIONALE FOR NEW COURSE:

a) Why is there a need for this course at this time?
According to the National Vulnerability Database, the United States government repository of standards based vulnerability management data, several hundred software vulnerabilities are discovered each month. Such vulnerabilities compromise the security of our cyberinfrastructure and allow for malicious cyber attacks. In many cases, software vulnerabilities are a result of poor programming practices. Cyber attacks that exploit software vulnerabilities can have a catastrophic impact on an organization legally and financially. Thus, security organizations are emphasizing the importance of proactive measures to alleviate software vulnerabilities. Actively adhering to well documented and enforceable standards in addition to patching software vulnerabilities in retrospect will help secure our cyberinfrastructure. This course will provide a foundation towards addressing this need.

b) How often is the course likely to be taught?
Every other semester

c) How does this course fit into your department's curriculum?
The department currently does not have an undergraduate course in software security. This course will fulfill that need. Moreover, a new undergraduate certificate in Cybersecurity Informatics is being proposed by the IS department. This course will serve as a core course requirement for that certificate. Also, this course can be used by

IS majors as the required upper-level IS elective (if it is not being used for the certificate) or as the required third programming course.

d) What primary student population will the course serve?
Undergraduate IS majors who need a third programming course or an upper-level IS elective, or undergraduate IS majors who declare the Cybersecurity Informatics certificate.

e) Why is the course offered at the level (ie. 100, 200, 300, or 400 level) chosen?
This is an advanced upper level elective course, with a 200-level prerequisite, therefore has a 400 level designation.

f) Explain the appropriateness of the recommended course preparation(s) and prerequisite(s).
In order to understand software engineering techniques for security, prior understanding of software creation, or programming, is required. IS 247 is the IS department's second Java course and provides an appropriate foundation in programming.

g) Explain the reasoning behind the P/F or regular grading method.
This course will test the students' skills in programming and software engineering for security. There will be graded assignments and exams which will help students accumulate the points towards a final grade. The level of skill proficiency by the end of the semester will determine the level of the grade. Therefore it is a regular, letter graded course.

h) Provide a justification for the repeatability of the course.
This course is not repeatable.

ATTACH COURSE OUTLINE (mandatory):

Please see syllabus Attached.

IS 472 Software Security DRAFT

Important Information:

Meets:	Mondays and Wednesdays, 10-11:30am, in ITE 458
Professor:	Dr. Carolyn Seaman
Office:	ITE 404B
Phone:	410-455-3937
Email:	cseaman@umbc.edu
Office hours:	Mondays 12-2pm and Thursdays 10am-12noon, and by appointment
Required text:	Secure Software Development by Jason Grembi, Cengage Learning (other required readings will be provided by the instructor)

Course Description

Many cybersecurity attacks are facilitated by software vulnerabilities, i.e. characteristics of software source code that unintentionally allow unauthorized access to computer memory. Future cybersecurity professionals require an understanding of various techniques that can be applied throughout the software development lifecycle to prevent, detect, and remove such vulnerabilities. Through this course students will: 1) Develop an understanding of common vulnerabilities and emerging attacks; 2) Learn how to apply secure coding standards and techniques to ensure that their source code is as free from vulnerabilities as possible; 3) Be provided hands-on experience in identifying and preventing software vulnerabilities.

Learning Objectives:

After successfully completing this course, students should:

- Understand how each phase of the Software Development Life Cycle (SDLC) contributes to developing secure software
- Understand how software vulnerabilities are exploited in cyber attacks
- Be able to apply secure coding principles
- Be able to apply techniques in design, coding, quality assurance, and maintenance, for producing secure software
- Understand the published standards in place for secure software development

Blackboard site

A Blackboard site will be maintained for the course throughout the semester. The site will contain all **announcements** pertinent to the course, as well as all class materials, handouts, and assignments. You will also use the Blackboard site to submit most assignments. **Each student is responsible for checking the site regularly, and for being aware of any information posted there.** In particular, it is advised that you check the Blackboard site on the day before each class in order to download any handouts you will need during class, and any information about preparing for class.

Office Hours

Every student is strongly encouraged to make use of office hours. I am willing to go over anything you are having problems with, or to discuss any issues having to do with the course or the program. My official office hours are listed above, but I am also available by appointment, which means that you should call or email me before stopping by my office to make sure that I will be in if it's outside of the stated office hours. Please feel free to discuss things with me via email and phone as well. I check both numerous times each day and will respond promptly. I cannot guarantee that I will check my messages on the weekend, but I often do.

Grading

The University's Undergraduate Catalogue states that, "A, indicates superior achievement; B, good performance; C, adequate performance; D, minimal performance; F, failure". There is specifically no mention of any numerical scores associated with these letter grades. Consequently, there are no pre-defined numerical boundaries that determine final letter grades. These boundaries can only be defined at the end of the semester after all scores have been earned. At that point, numerical boundaries for final letter grades can be defined (usually using a "curve"). This means that it is not appropriate to assume that a given numerical score corresponds to a particular letter grade. It is also important to understand that final letter grades reflect academic achievement and not effort.

While I am more than happy to correct mistakes in the computation of grades and grade recording errors, in all other situations final letter grades are not negotiable.

Your final course grade will be based on scores received on three exams, reading quizzes, and in-class exercises, as follows:

- Exams (3) – 10% each
  There will be three exams occurring throughout the semester (see the Schedule, below). All will be take-home, open-book, essay question exams. See Policies, below, for my rules about missing exams.

- Quizzes – 10%
  There will be a minimum of 5 quizzes during the course of the semester (probably more). The quizzes will be in-class, closed-book, and unannounced. Each quiz will be given at the beginning of the class session, and the topic of the quiz will be limited to what was covered in the assigned reading for that day. The objective of the quizzes is to motivate students to attend class, be prepared for class, and keep up with the assigned reading. I will drop each student's lowest quiz grade in calculating the final grade for the semester. See Policies, below, for my rules about missing quizzes.

- In-class exercises – 30%

There will be several in-class exercises over the course of the semester (see the Schedule, below). Each exercise will involve the application of some software engineering or coding technique for creating secure software. The exercises will be cumulative, in that each will contribute to a single software artifact that will be complete at the end of the semester. Exercises will also involve reviewing other students' work. I will drop each student's lowest exercise grade in calculating the final grade for the semester.

- Final group project – 30%
  The final project will build on components developed over the course of the semester in the in-class exercises. More information about the final project will be supplied in a separate document.

Policies

1. ***Missing exams***
   In general, if you miss the exam, you will receive a grade of 0 for the exam. If you know that you will have to miss the exam in advance, come talk to me about it. **If** I am given sufficient notice, and I agree that your absence cannot be avoided, then I can arrange a makeup exam. If you miss an exam due to an unforeseen emergency, then we can arrange a makeup exam **if** I agree that your absence was due to a bona fide emergency and you can document that emergency to my satisfaction. In all cases, you should be warned that makeup exams are generally more difficult and more prone to errors and misunderstandings than the original exam, simply because I do not have the time to devote to writing a makeup exam as carefully as I do other exams.

2. ***Missing quizzes***
   If you miss a quiz, you can make it up with a 50% penalty. That is, you will only get credit for half of whatever score you get on the quiz. The quiz must be made up as soon as possible after the class on which it was originally given, ideally the same day or at least before the next class. If too much time passes after the quiz was given in class before you request a make-up, I will not allow you to take the make-up and you will get a 0 for the quiz. If you arrive late to class, and a quiz is already in progress when you arrive, you can begin the quiz when you arrive, but must turn it in at the same time as the rest of the class.

3. ***Missing in-class exercises***
   Because of the nature of the in-class exercises, they cannot be made up if you miss class that day. Each exercise is worth 3 points, and you can make up 1 point of that if you turn in a revised version of the assignment (that can be done individually) before the next class session.

4. ***Coming late to class***
   There is no specific penalty for coming late to class, except the potential to miss quizzes. However, nothing said or done in the first part of the class will be repeated for latecomers. If a student's late arrival to class is disruptive in any way, that student will be asked to leave the classroom.

5. ***Academic Dishonesty***
   By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabricating, plagiarism, and helping others to commit these acts are all forms of academic dishonesty and they are wrong. Academic misconduct could result in disciplinary action that may range from a grade of 0 on the relevant assignment or failure of the entire course, to suspension or dismissal from the program.

   In particular, for this course:
   - No cheating will be tolerated on the exams or on quizzes. Cheating includes gaining specific information about the quiz before taking it (e.g. in the case of a make-up), as well as consulting unauthorized materials during the quiz or exam.
   - Plagiarism applies to the in-class exercises. All work submitted for these assignments must be created, during the class period in which the exercise takes place, by the students submitting the work.
   - Academic dishonesty also includes interfering with another student's work or aiding another student to commit academic dishonesty.

Tentative schedule

Below is a tentative schedule of lecture topics, exercises, exams, due dates, and other activities. I will avoid changes if at all possible, but if I have to make a change I will let you know well in advance. The latest updated schedule will always be on Blackboard, so you can check there to be sure.

Date	Topic	Activity	Reading
Monday, January 25, 2016	Course introduction	Syllabus quiz	
Wednesday, January 27, 2016	Software engineering basics	Lecture/discussion	Chapters 1-2
Monday, February 01, 2016	Security basics	Lecture/discussion	Chapters 3-4
Wednesday, February 03, 2016	Tool environment	Lab exercise	
Monday, February 08, 2016	Requirements gathering	Lecture/discussion	Chapter 5
Wednesday, February 10, 2016	Requirements gathering	Lab exercise	
Monday, February 15, 2016	Design	Lecture/discussion	Chapters 6-7
Wednesday, February 17, 2016	Design	Lab exercise	
Monday, February 22, 2016	Java review	Lab exercise	
Wednesday, February 24, 2016	Touchpoints introduction	Lecture/discussion	Chapters 8-9, plus McGraw reading
Monday, February 29, 2016	Exam 1		
Wednesday, March 02, 2016	Touchpoints	Lab exercise	
Monday, March 07, 2016	Touchpoints	Lab exercise	
Wednesday, March 09, 2016	Touchpoints	Lab exercise	
Monday, March 14, 2016	Spring break		
Wednesday, March 16, 2016	Spring break		
Monday, March 21, 2016	Revision/Configuration Management, Build & Integration	Lecture/discussion	
Wednesday, March 23, 2016	Revision/Configuration Management, Build & Integration	Lab exercise	
Monday, March 28, 2016	Exam 2		
Wednesday, March 30, 2016	Testing	Lecture/discussion	Chapter 10
Monday, April 04, 2016	Testing	Lab exercise	
Wednesday, April 06, 2016	Reviews and Logging	Lecture/discussion	
Monday, April 11, 2016	Reviews	Lab exercise	
Wednesday, April 13, 2016	Logging	Lab exercise	
Monday, April 18, 2016	Maintenance	Lecture/discussion	Chapter 11
Wednesday, April 20, 2016	Maintenance	Lab exercise	
Monday, April 25, 2016	Standards	Lecture/discussion	Shoemaker and Sigler reading
Wednesday, April 27, 2016	Exam 3		
Monday, May 02, 2016	Software safety	Lecture/discussion	
Wednesday, May 04, 2016	Software safety	Guest lecture	
Monday, May 09, 2016	Putting it all together	Lab exercise	