SECURE CODING

DEPARTMENT OF COMPUTER SCIENCE

INSTRUCTOR INFORMATION

- **Instructor:** Sheikh Rabiul Islam, Ph.D.
- **Office:** Business and Science Building, Room 320, 227 Penn Street, Camden, NJ 08102
- **Email:** *sheikh.islam@rutgers.edu*
- **Office Hours:**
    - Monday and Wednesday 1:00-2:00 p.m.
    - Please feel free to send me an email to schedule an appointment at another time.
    - Please don't hesitate to visit at any time to check if I am available to assist you.

COURSE INFORMATION

- **Course:** Secure Coding (50:198:355)
  Semester: Spring 2024; **Credits:** 3.00.
- **Location:** BSB - 416; **Day:** Monday and Wednesday; **Time:** 9:35 am - 10:55 am

COURSE DESCRIPTION

The Secure Coding course focuses on the pivotal relationship between software design and security, deviating from traditional security-centric approaches. It emphasizes the critical role of design decisions in mitigating security vulnerabilities. Commencing with foundational principles and real-world case studies that spotlight design flaws, the course progresses through essential concepts such as domain-driven design, entity integrity, and secure coding constructs. It culminates in an exploration of challenges posed by legacy systems and microservice architectures, highlighting the importance of periodic security assessments in software development. Tailored to suit a diverse audience, the course surpasses programming language-specific barriers, providing unified insights into fortifying software systems against vulnerabilities through a design-oriented perspective.

COURSE OBJECTIVES/STUDENT LEARNING OUTCOMES

The learning objectives are:

- Comprehend the fundamental correlation between software design choices and their impact on security vulnerabilities, recognizing how well-crafted design decisions can preempt security flaws within software systems.
- Gain proficiency in essential concepts of Domain-Driven Design (DDD), employing its principles as foundational elements to construct secure software entities and systems.
- Identify and address security pitfalls prevalent in legacy codebases and modern architectures, including microservices, leveraging design-centric methodologies to rectify vulnerabilities.

- Develop expertise in integrating secure coding constructs such as immutability, data validation, and error handling into software design, ensuring robustness against potential security breaches.
- Cultivate a critical mindset to evaluate existing systems periodically, applying secure design principles to fortify software integrity and proactively mitigate emerging security threats, transcending conventional security measures.

MAJOR TEACHING METHODS

Lectures, Demonstrations, Labs/Assignments, and Reading. Expect to spend at least 9 hours/week on this course, including class meeting times.

TOPICS AND SCHEDULE

Tentative course schedule:

Schedule	Topics	Subtopics	Note
Week 1 *January*	Why design matters for security	Security is a concern, not a feature; design; The traditional approach to software security and its shortcomings; Driving security through design; Dealing with strings and XML	Ch 1
Week 2	Case studies	An online bookstore with business integrity issues; Shallow modeling, Deep modeling	Ch 2
Week 3	Core concepts of domain-driven design	Models as tools for deeper insight; Building blocks for your model; Bounded contexts; Interactions between contexts	Ch 3
Week 4	Code constructs promoting security	Immutability; Validation (Checking the origin, size, lexical content, syntax, and semantics of data)	Ch 4 Lab-1
Week 5 *February*	Domain primitives	Domain primitives and invariants; Read-once objects; The risk with overcluttered entity methods; Taint analysis	Ch 5 Quiz 1
Week 6	Ensuring integrity of state	Managing state using entities; Consistent on creation; Integrity of entities;	Ch 6 Assignment 1
Week 7	Reducing complexity of state	Reducing complexity of state; Entity state objects; Entity snapshots; Entity relay; Splitting the state graph into phases;	Ch 7
Week 8 *March*	Leveraging your delivery pipeline for security	Using a delivery pipeline; Securing your design using unit tests; Verifying feature toggles; Dealing with combinatory complexity; Automated security tests; Testing for availability; Validating configuration;	Ch 8, Test-1, Lab-2
Week 9	Handling failures securely	Using exceptions to deal with failure; Handling failures without exceptions; Designing for availability (Resilience, Responsiveness, Circuit breakers and timeouts, Bulkheads); Handling bad data	Ch 9
Week 10	Benefits of cloud thinking	The twelve-factor app and cloud-native concepts; Storing configuration in the environment; Separate processes; Avoid logging to file; Admin processes; Service discovery and load	Ch 10 Assignment-2

		balancing; The three R’s (Rotate, Repave, Repair) of enterprise security;	
Week 11	Case Studies	An insurance policy for free;	Ch 11 Quiz 2
Week 12	Guidance in legacy code	Determining where to apply domain primitives in legacy code; Ambiguous parameter lists; Logging unchecked strings; Defensive code constructs; DRY misapplied—not focusing on ideas, but on text; Insufficient validation in domain types; Only testing the good enough; Partial domain primitives;	Ch 12 Lab-3
Week 13 *April*	Guidance on micros Guidance on microservices	Intro to microservice; Each service is a bounded context; Sensitive data across services; Logging in microservices;	Ch 13 Assignment-3
Week 14	A final word: Don’t forget about security!	Conduct code security reviews; Keep track of your stack; Run security penetration tests; Study the field of security; Develop a security Incident mechanism;	Ch 14 Test-2, Term Project/Report;

REQUIRED MATERIALS

1. **Book**

   Secure by Design – authored by Dan Bergh Johnsson, Daniel Deogun, and Daniel Sawanos

   Online Version: https://livebook.manning.com/book/secure-by-design/

2. **Canvas:** https://canvas.rutgers.edu/
   Used for accessing course syllabus, materials, and grades, and turning in assignments.
3. **Software**: Oracle Virtual Box, Burp Suit, OWASP Tool.

GRADES

- All grades will be posted in Canvas.
- If you have an issue with a grade you receive on ANY assignment or exam, you must email the instructor within THREE days of the grade being released to the class.
- Grade Scale:
  89.5–100 = A (Outstanding)
  84.5–89.49 = B+, 79.5–84.49 = B (Good),
  74.5–79.49 = C+, 69.5–74.49 = C (Satisfactory),
  59.5–69.49 = D (Poor),
  0–59.49 = F (Failing).
  A grade of C or better is usually required for Major or Minor courses, while General Requirement courses must only be passed with a D or better. The grade of D is not valid for graduate-level courses. Students may only receive a C or better, F or IN for graduate courses.

Grades in this course are **earned** using the following distribution:

Item	Percentage
Assignments 6 labs/assignments (30%)	30%
Exams 2 exams (40%) 2 quizzes (10%)	50%
Term Report/Project	15%
Participation Responsiveness (5%)	5%

LABS/ASSIGNMENTS (30% OF GRADE)

There will be six labs/assignments for gaining practical experience on the covered concepts.

EXAMS (50% OF GRADE)

- There will be two exams containing 40% of the total grade combined.
- The last exam is not comprehensive.
- There will be two quizzes containing 10% of the grade.

TERM REPORT (15% OF GRADE)

- The term report will be on exploring or solving a coding/programming vulnerability. You can survey the state-of-the-art techniques, and tools on the selected topic or work on a real-world coding/programming security-related problem.
- Projects involving implementation and coding can be done in groups consisting of two members.
- Within the seventh week of the semester, you need to get approval of the proposed project/topic in written format.
- A student will need to submit a term report (at least 4 single-spaced pages) on the project or selected topic. The page requirement for projects/reports containing implementation or programming to solve an issue is half.

PARTICIPATION (5% OF GRADE)

- This is based on your responsiveness in the lecture sessions and discussions.

POLICY

- All students should follow the Academic Integrity Policy as mentioned at: https://deanofstudents.camden.rutgers.edu/sites/deanofstudents/files/Academic%20Integrity%20Policy.pdf

- Whenever you submit any work, **you must acknowledge the source** if any part of the submitted **content is not originally yours**.

SERVICES AND RESOURCES

- A comprehensive list of student services and resources are listed here: https://studentaffairs.camden.rutgers.edu/student-resource-list

- Here are some crucial student services and resources:

- The Center for Learning and Student Success (CLASS) provides academic support and enrichment services for students, at no additional cost, including one-on-one tutoring, small-group tutoring and workshops, online tutoring, writing assistance, student success coaching, learning assessment, and metacognition training. Learn more about this service here: https://class.camden.rutgers.edu/

- Office of Disability Services (ODS)—Students with Disabilities: If you need academic support for your courses, accommodation can be provided as indicated in the accommodation letter. If you have not registered with ODS and you have or think you have a disability (learning, sensory, physical, chronic health, mental health, or attentional), please visit the ODS website: https://success.camden.rutgers.edu/disability-services

- Dean of Students Office—CARES Team: For some students, personal, emotional, psychological, academic, or other challenges may hinder their ability to succeed both in and outside of the classroom. The Dean of Students Office serves as your initial contact if you need assistance with these challenges. You can learn more about the free services by visiting the Dean of Students website http://deanofstudents.camden.rutgers.edu/