

# COURSE SYLLABUS

## Securitate Software

### 1. Program identification details

1.1 Higher education institution	UNIVERSITATEA „OVIDIUS”, CONSTANȚA
1.2 Faculty	Facultatea de Matematică și Informatică
1.3 Department	Matematică și Informatică
1.4 Field of studies	Informatică
1.5 Cycle of studies (degree)	Master
1.6 Degree program/qualification	Securitate Cibernetică și Învățare Automată
1.7 Academic year	2025-2026

### 2. Course identification details

2.1 Course title	SECURITATE SOFTWARE				
2.2 Course code	FMI.CSML.II.2.05				
2.3 Instructor	Conf. Univ. Dr. Mihailescu Marius Iulian				
2.4 Teaching assistant	Conf. Univ. Dr. Mihailescu Marius Iulian				
2.5 Year	2	2.6 Semester	2	2.7. Evaluation type	E
2.8 Course type */**					DCA/DI

\* DF – fundamental course, DD – field course, DS – specialty course, DC – complementary course, DAP – advanced study course, DSI – synthesis course, DCA – advanced knowledge course.

\*\* DI – mandatory course; DO – optional course.

### 3. Estimated workload (hours per semester)

3.1 Number of teaching hours/week	3	of which: 3.2 course	2	3.3 applications***	1
3.4 Total of teaching hours within the program/semester	36	of which: 3.5 lecture	24	3.6 seminar	12
<b>3.7 Student workload for individual study</b>					139
<b>Distribution of workload</b>					[hours]
Studiu individual al manualelor, ghidurilor/cartelor de lectură, bibliografiei și notițelor					53
Cercetare suplimentară (bibliotecă, resurse electronice, muncă de teren)					26
Teme (pregătirea prezentărilor de seminar, portofoliilor, eseurilor critice, lucrărilor de cercetare etc.)					28
Consultări individuale (opțional)					28
Evaluări / examene					4
Alte activități					
<b>3.8 Total hours per semester</b>	36+139=175				
<b>3.9 Number of credits</b>	7				

\*\*\* S - seminar; L - laboratory; P - project

### 4. Prerequisites (where applicable)

4.1 Curriculum-related	Undergraduate studies; Software engineering;
4.2 Skills-related	

### 5. Necessary requirements for optimum teaching and learning (where applicable)

5.1. For running the course	Sală de clasă disponibilă
5.2. For running the seminar / laboratory /project	Sală de clasă disponibilă / sală de laborator disponibilă

\*The type is to be chosen according to the discipline

## 6. Course objectives

6.1 The general objective of the course	Dezvoltarea de sisteme software sigure și robuste
6.2 Specific objectives	Elaborarea modelelor necesare pentru procesul de dezvoltare a unui sistem software securizat și robust Implementarea sistemelor securizate

## 7. Learning outcomes

Knowledge	<p><i>Principii fundamentale ale securității software:</i> Concepte de bază, inclusiv confidențialitatea, integritatea, disponibilitatea, autentificarea și autorizarea; Modele și politici de securitate (de exemplu, Bell-LaPadula, Biba, controlul fluxului de informații); Metodologii de evaluare a riscurilor și modelare a amenințărilor (de exemplu, STRIDE, arbori de atac, DREAD). <i>Vulnerabilități și exploatare software:</i> Clase comune de vulnerabilități software (de exemplu, buffer overflow-uri, injection flaws, race condition, deserializare, escaladare de privilegii); Cauzele principale ale vulnerabilităților în diverse paradigme de programare (procedurale, orientate pe obiecte, funcționale, bazate pe web); Tehnici de exploatare și mecanismele lor subiacente, inclusiv coruperea memoriei, deturnarea fluxului de control și defectele logice. Proiectare și dezvoltare software securizată: Principii de securitate prin proiectare (privilegii minime, apărare în profunzime, valori implicite de siguranță, valori implicite securizate); Tehnici de scriere a codului securizat în diferite medii (limbaje de nivel scăzut, runtime-uri gestionate, framework-uri web); Ciclul de viață al dezvoltării securizate (SDL), inclusiv analiza cerințelor, ghiduri de codare securizată, revizuirea codului și testarea securității. <i>Mecanisme și arhitecturi avansate de securitate:</i> Fundamente criptografice relevante pentru securitatea software-ului (de exemplu, gestionarea securizată a cheilor, TLS/SSL, protocoale de autentificare); Modele de control al accesului, gestionarea identității și sisteme de autentificare federative; Izolare și sandboxing, protecția memoriei, containerizare și securitatea microserviciilor. <i>Testarea și verificarea securității:</i> Principii de analiză statică și dinamică pentru detectarea vulnerabilităților; Testare fuzz, testare de penetrare, verificare formală și monitorizare în timpul rulării; Utilizarea instrumentelor automate pentru analiza codului, scanarea vulnerabilităților și securitatea integrării continue. Tendințe actuale și amenințări emergente: Amenințări persistente avansate (APT), atacuri asupra lanțului de aprovizionare și manipulare software; Provocări de securitate în tehnologiile emergente (de exemplu, IoT, aplicații cloud-native, sisteme AI/ML, blockchain); Aspecte juridice, etice și de reglementare ale securității software-ului (de exemplu, GDPR, NIS2, ISO/IEC 27034).</p>
-----------	---

<b>Skills</b>	<p>Până la sfârșitul cursului, studenții masteranzi vor fi capabili să aplice abilități avansate de analiză, proiectare și implementare pentru a aborda provocări complexe de securitate software în contexte reale. Vor fi capabili să identifice și să evalueze sistematic vulnerabilitățile din sistemele software prin metode precum modelarea amenințărilor, analiza codului și testarea penetrării. Folosind atât tehnici manuale, cât și instrumente automate, studenții vor efectua evaluări amănunțite ale securității, vor interpreta constatările și vor propune strategii eficiente de atenuare. Studenții masteranzi vor dezvolta capacitatea de a proiecta și implementa soluții software securizate prin integrarea considerațiilor de securitate pe parcursul întregului ciclu de viață al dezvoltării software. Aceasta include aplicarea unor practici de codare sigure, încorporarea testelor de securitate în conductele de integrare continuă și integrarea controalelor de securitate în arhitecturile de sistem. Mai mult, studenții vor dobândi abilitățile de a analiza și de a răspunde la amenințările de securitate în evoluție prin aplicarea de mecanisme criptografice, proiectarea de scheme de autentificare și control al accesului securizate și implementarea de măsuri de protecție în timpul rulării. Vor fi echipați pentru a lucra cu sisteme complexe, distribuite - inclusiv medii cloud-native și containerizate - și pentru a aplica mecanisme de securitate adecvate, adaptate acestor contexte.</p>
<b>Responsibility and autonomy</b>	<p>La finalizarea cursului, studenții vor putea exercita o responsabilitate și o autonomie substanțială în gestionarea sarcinilor și proiectelor complexe de securitate software. Vor demonstra capacitatea de a planifica, executa și evalua independent activitățile de securitate pe tot parcursul ciclului de viață al dezvoltării software, de la proiectare și implementare până la testare și întreținere. Aceasta include luarea inițiativei în identificarea priorităților de securitate, selectarea metodologiilor adecvate și aplicarea lor eficientă atât în contexte individuale, cât și în contexte de colaborare.</p> <p>Studenții vor fi pregătiți să ia decizii informate, etice și solide din punct de vedere tehnic atunci când se confruntă cu provocări de securitate din lumea reală, inclusiv situații care implică incertitudine, informații incomplete sau amenințări emergente. Vor fi capabili să își asume roluri de conducere în echipe multidisciplinare, îndrumând colegii și părțile interesate în implementarea practicilor software securizate și promovând o cultură a securității în cadrul organizațiilor.</p> <p>Mai mult, studenții vor demonstra capacitatea de a reflecta critic asupra practicii lor profesionale, de a-și actualiza continuu cunoștințele ca răspuns la noile evoluții din peisajul securității și de a-și asuma responsabilitatea pentru implicațiile societale și juridice ale deciziilor lor tehnice. Vor acționa cu integritate și responsabilitate, asigurându-se că activitatea lor în domeniul securității se aliniază cu cadrele legale, politicile organizaționale și standardele etice mai largi.</p>

## 8. Contents

8.1 Lecture	Teaching methods	No. of hours
Security in software systems	Prelegere cu sinteza și esențializarea informațiilor	2 hours
Security Requirements Analysis	Metode de predare prin învățare interactivă	2 hours
Threat modeling		2 hours
Security design	Dialog	2 hours
Security design patterns		2 hours
Security protocols	Problematică	4 hours
Implementation of secure systems	Conversație	8 hours

User training on system security	Metode care contribuie la dezvoltarea gândirii critice	2 hours
	Programe	
	Învățare independentă și cooperativă	
<b>Bibliography:</b> [1]. Kohnfelder, Loren. Designing Secure Software: A Guide for Developers. No Starch Press, 2022. [2]. Saini, Kavita, and Pethuru Raj, editors. Advancing Smarter and More Secure Industrial Applications Using AI, IOT, and Blockchain Technology. Engineering Science Reference, 2021. [3]. A.K. Talukder, M. Chaitanya, <i>Architecting secure software systems</i> , CRC Press, 2009 [4]. A. Shostack, <i>Threat Modeling. Designing for security</i> , John Wiley & sons, 2014 [5]. D. Huang, A. Chowdhary, S. Pisharody, <i>Software-defined networking and security. From theory to practice</i> , CRC Press, 2019 [6]. C. W. Axelrod, <i>Engineering safe and secure software systems</i> , Artech House, 2012		

8.2 Applications* (seminar/lab/project) <i>*The type is to be chosen according to the discipline</i>	Teaching methods	Number of hours
Writing of requirement documents	Dialog Problematizare Conversație Metode care contribuie la dezvoltarea gândirii critice Programe Învățare independentă și cooperativă	1 hour
Cyber threat modeling techniques		1 hour
Applying security design models to build software architecture		2 hours
Applications for ensuring the security of software systems		4 hours
Implementation of an IDS / IPS system		4 hours
<b>Bibliography:</b> [1]. Kohnfelder, Loren. Designing Secure Software: A Guide for Developers. No Starch Press, 2022. [2]. Saini, Kavita, and Pethuru Raj, editors. Advancing Smarter and More Secure Industrial Applications Using AI, IOT, and Blockchain Technology. Engineering Science Reference, 2021. [3]. A.K. Talukder, M. Chaitanya, <i>Architecting secure software systems</i> , CRC Press, 2009 [4]. A. Shostack, <i>Threat Modeling. Designing for security</i> , John Wiley & sons, 2014 [5]. D. Huang, A. Chowdhary, S. Pisharody, <i>Software-defined networking and security. From theory to practice</i> , CRC Press, 2019 [6]. C. W. Axelrod, <i>Engineering safe and secure software systems</i> , Artech House, 2012		

## 9. Correlation between the content of the course and the needs/expectations of the epistemic community, professional associations and/or significant employers relevant for the program

Conținutul cursului este în strânsă concordanță cu tendințele actuale de cercetare, standardele industriei și așteptările profesionale în domeniul securității software.

## 10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	Active participation		
10.5 Applications* (Laboratory) <i>*The type is to be chosen according to the discipline</i>	Active participation		
	Project	Oral	60%

	Exam	Oral	40%
10.6 Minimum standard of achievement for the acquisition of the ECTS credits			
Redactarea documentului de cerințe pentru un sistem software securizat			

Date of completion

12.09.2025

Course Instructor,

Conf. Univ. Dr. Mihailescu Marius

Teaching Assistant,

Conf. Univ. Dr. Mihailescu Marius

Date of approval in the Department

19.09.2025

Head of Department,

Assoc. Prof. Pelican Elena, Ph.D

Dean,

Assoc. Prof. Nicola Aurelian, Ph.D