

25HCST401	SECURE CODING FUNDAMENTALS	Category	L	T	P	R	Credit	Year of Introduction
		VAC	4	0	0	0	4	2025

Preamble: Secure Coding Fundamentals is designed to equip students with the knowledge and practical skills required to develop secure software applications in today's threat-prone digital environment. This course introduces students to secure coding principles, common software vulnerabilities such as injection flaws and authentication weaknesses, and industry-recognized standards including OWASP guidelines..

Prerequisite: Nil

Course Outcomes: After the completion of the course, the student will be able to:

CO	Course Outcomes	Knowledge Level
CO1	Explain secure coding principles, common software vulnerabilities and the importance of secure software development lifecycle	K2 Understand
CO2	Explain common web application security vulnerabilities and various prevention mechanisms.	K2 Understand
CO3	Apply secure coding practices and validation techniques to develop programs that prevent common security attacks.	K3 Apply
CO4	Demonstrate the use of secure coding standards, code review techniques, and basic static/dynamic analysis tools to detect vulnerabilities in applications.	K3 Apply

Mapping of course outcomes with program outcomes and program specific outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	-	-	-	2	-	-	-	1	3	1
CO2	3	3	2	-	-	-	2	-	-	-	1	3	2
CO3	3	3	3	1	-	-	2	-	-	-	1	3	3
CO4	2	3	3	2	-	-	1	-	-	-	2	3	3
Avg	2.75	2.75	2.25	1.5	-	-	1.75	-	-	-	1.25	3	2.25

Mapping of course outcomes with Knowledge and Attitude Profile (WK)

	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	*	*	*	*	*			*	*
CO2	*	*	*	*	*			*	*
CO3	*	*	*	*	*			*	*
CO4	*	*	*	*	*			*	*

Wks Mapped with the course

WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
*	*	*	*	*			*	*

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	40	60	2.5 hrs

Continuous Internal Evaluation Pattern:

Attendance	: 5 marks
Continuous Assessment Test	: 20 marks
Assignment	: 15 marks

SYLLABUS**Module 1 (14 hrs)**

Secure coding fundamentals and objectives, Importance of application security in modern systems, Secure Software Development Life Cycle, Basic security principles: Confidentiality, Integrity, Availability (CIA), Risk, threat, vulnerability in software applications, Overview of common software vulnerabilities (buffer overflow, injection flaws, improper validation)

Module 2 (14 hrs)

Injection flaws, Broken authentication, Sensitive data exposure, XML External Entities (XXE), Broken access control, Security misconfiguration, Cross-Site Scripting (XSS), Insecure deserialization, Using components with known vulnerabilities, Insufficient logging & monitoring.

Example attack scenarios of each of the vulnerabilities listed; how to prevent them

Module 3 (14 hrs)

Secure input handling and validation techniques, Output encoding and data sanitization, Secure error handling, Authentication mechanisms and secure login implementation, Authorization concepts and Role-Based Access Control, Password management best practices

Module 4 (14 hrs)

Secure coding standards and guidelines, Manual code review techniques, Secure logging practices, Secure error handling and monitoring, Remediation strategies and secure refactorin

Text Books:

1. Kohnfelder, L. (2021). *Designing secure software: A guide for developers*. No Starch Press.
2. Khan, S. A., Kumar, R., & Khan, R. A. (2023). *Software security: Concepts and practices*. CRC Press.

Reference Books:

1. Harwood, M., & Price, R. (2022). *Internet and web application security* (3rd ed.). Jones & Bartlett Learning.
2. Canlas, R. (2021). *ASP.NET Core 5 secure coding cookbook*. Packt Publishing.

END SEMESTER EXAMINATION – MODEL QUESTION PAPER

Course Code	25HCST401
Course Name	SECURE CODING FUNDAMENTALS
Max. Marks: 60	Duration: 2.5 Hours

Part A (Answer all questions. Each question carries 3 marks)				
Qn No.	Question	Marks	CO	BL
1	Explain the objectives of secure coding and its role in software security.	3	1	L2
2	Describe the significance of the CIA triad in secure software development.	3	1	L2
3	Differentiate between a threat, vulnerability, and risk in software applications. .	3	2	L2
4	Explain how improper input validation can lead to software vulnerabilities.	3	2	L2
5	A web application allows users to submit comments that are displayed to other users without proper input validation. Apply your knowledge of Cross-Site Scripting (XSS) to identify the vulnerability and explain how an attacker could exploit it. Suggest appropriate countermeasures to prevent	3	3	L3

	the attack			
6	Summarize the causes and prevention of broken authentication vulnerabilities.	3	3	L2
7	Explain the purpose of Role-Based Access Control (RBAC) in application security.	3	4	L2
8	A software development team is preparing to deploy a web application. Apply code review techniques to examine the source code, identify potential security vulnerabilities, and explain how the review process helps improve software security.	3	4	L3
Part B <i>(Answer any one full question from each Module. Each carry 9 marks)</i>				
Module I				
Qn No.	Question	Marks	CO	BL
9	Explain the Secure Software Development Life Cycle (SSDLC) and discuss how security principles such as confidentiality, integrity, and availability are incorporated throughout the development process.	9	1	L2
OR				
10	Describe common software vulnerabilities such as buffer overflows, injection flaws, and improper validation. Explain their impact on software security.	9	1	L2
Module II				
Qn No.	Question	Marks	CO	BL
11.a	Explain XML External Entity (XXE) attacks and discuss their potential impact on applications.	5	2	L2
11.b	Describe Cross-Site Scripting (XSS) attacks and explain the methods used to prevent them.	4	2	L2
OR				
12.a	A web application displays user-supplied input without proper validation. Explain the resulting vulnerability and its impact.	5	2	L2
12.b	Describe the role of logging and monitoring in detecting security incidents.	4	2	L2
Module III				
Qn No.	Question	Marks	CO	BL
13	A university portal allows students to upload files and enter personal information. Apply secure input validation, output encoding, and data sanitization techniques to protect the portal from common attacks.	9	3	L3

OR				
14	An online banking application requires secure user authentication and authorization. Demonstrate how authentication mechanisms, password management practices, and RBAC can be implemented to improve security.	9	3	L3
Module IV				
Qn No.	Question	Marks	CO	BL
15	A software company conducts a security assessment before releasing an application. Demonstrate how manual code reviews and secure coding standards can help identify and remediate vulnerabilities.	9	4	L3
OR				
16	During a security audit, several coding flaws and insecure logging practices are detected. Apply suitable remediation and secure refactoring strategies to improve application security.	9	4	L3