# Certified Secure Coding for Software Developers (CSCSD)

## Course Duration: 16 Hours (2 Days)

## Overview

The Certified Secure Coding for Software Developers (CSCSD) course is a comprehensive training program designed to equip software developers with the essential principles and practices for writing secure code. The course provides an in-depth understanding of security threats and how to mitigate them throughout the software development lifecycle. Module 1 kicks off with an introduction to the course, setting the stage for the importance of secure coding. Module 2 delves into core security concepts such as Confidentiality, Integrity, Availability, and Non-repudiation, emphasizing the importance of Data anonymization, User consent, and Disposition. In Module 3, learners explore the Secure Development Lifecycle, comparing methodologies like Waterfall and Agile, and examining frameworks like the Microsoft SDLC, Touchpoints, and CLASP. Module 4 focuses on Security Design Principles, teaching developers to apply concepts like Least privilege, Defense in depth, and Fail-safe to prevent vulnerabilities. Finally, Module 5 addresses Secure Development Principles, stressing the importance of Canonicalization, Output encoding, and secure practices for Authentication & authorization, Auditing & logging, and maintaining Secure communications. Learners who complete the CSCSD course will be equipped to write safer, more secure code, reducing the risk of security breaches and enhancing the overall security posture of their software applications.

## Audience Profile

The Certified Secure Coding for Software Developers (CSCSD) course focuses on security principles and practices for building robust, secure applications.

- Software Developers
- Application Programmers
- Security Analysts
- Software Architects
- Systems Engineers
- IT Security Consultants
- Software Auditors
- Quality Assurance specialists
- Project Managers (with a technical background)

## Course Syllabus

### 1. Introduction

- **a.** Disclaimer
- **b.** Trends & Metrics

- **c.** Lab Environment

## 2. Core Security Concepts

- **a.** Confidentiality, Integrity, and Availability
- **b.** Authentication and Authorization
- **c.** Accounting
- **d.** Non-repudiation
- **e.** Privacy
- **f.** Data Anonymization
- **g.** User Consent
- **h.** Data Disposition
- **i.** Test Data Management

## 3. Secure Development Lifecycle

- **a.** Waterfall vs. Agile
- **b.** Microsoft SDLC
- **c.** Touchpoints
- **d.** CLASP (Comprehensive, Lightweight Application Security Process)
- **e.** Lifecycle Comparison

## 4. Security Design Principles

- **a.** Least Privilege
- **b.** Separation of Duties
- **c.** Defense in Depth
- **d.** Fail-Safe Defaults
- **e.** Economy of Mechanism
- **f.** Complete Mediation
- **g.** Open Design
- **h.** Least Common Mechanism
- **i.** Psychological Acceptability
- **j.** Weakest Link Principle
- **k.** Leveraging Existing Components

## 5. Secure Development Principles

- **a.** Input Validation
- **b.** Canonicalization
- **c.** Output Encoding
- **d.** Error Handling

- **e.** Authentication & Authorization
- **f.** Auditing & Logging
- **g.** Session Management
- **h.** Secure Communications
- **i.** Secure Resource Access
- **j.** Secure Storage
- **k.** Cryptography

## 6. Best Practices

## 7. Conclusion