

Cyber Secure Coder

Duration 3 days

COURSE DESCRIPTION

The stakes for software security are very high, and yet many development teams deal with software security only after the code has been developed and the software is being prepared for delivery. As with any aspect of software quality, to ensure successful implementation, security and privacy issues should be managed throughout the entire software development lifecycle. This course presents an approach for dealing with security and privacy throughout the entire software development lifecycle. You will learn about vulnerabilities that undermine security, and how to identify and remediate them in your own projects. You will learn general strategies for dealing with security defects and misconfiguration, how to design software to deal with the human element in security, and how to incorporate security into all phases of development.

COURSE OBJECTIVES

In this course, you will employ best practices in software development to develop secure software.

You will:

- Identify the need for security in your software projects.
- Eliminate vulnerabilities within software.
- Use a Security by Design approach to design a secure architecture for your software.
- Implement common protections to protect users and data.
- Apply various testing methods to find and correct security defects in your software.
- Maintain deployed software to ensure ongoing security.

This course includes hands on activities for each topic area. The goal of these activities is to demonstrate concepts utilizing two universal languages Python and Java Script. Developers who use alternate languages will be able apply the principles from the activities to any coding languages. Hands on exercises are designed to keep the typing of code to a bare minimum. CertNexus provides students with all of the code they need to complete activities. The activities do not require a “deep dive” into code to understand the principles being covered.

COURSE OUTLINE

Lesson 1: Identifying the Need for Security in Your Software Projects

- Identify Security Requirements and Expectations
- Identify Factors That Undermine Software Security
- Find Vulnerabilities in Your Software
- Gather Intelligence on Vulnerabilities and Exploits

Lesson 2: Handling Vulnerabilities

- Handle Vulnerabilities Due to Software Defects and Misconfiguration
- Handle Vulnerabilities Due to Human Factors
- Handle Vulnerabilities Due to Process Shortcomings

Lesson 3: Designing for Security

- Apply General Principles for Secure Design
- Design Software to Counter Specific Threats

Lesson 4: Developing Secure Code

- Follow Best Practices for Secure Coding
- Prevent Platform Vulnerabilities
- Prevent Privacy Vulnerabilities

Lesson 5: Implementing Common Protections

- Limit Access Using Login and User Roles
- Protect Data in Transit and At Rest
- Implement Error Handling and Logging
- Protect Sensitive Data and Functions
- Protect Database Access

Lesson 6: Testing Software Security

- Perform Security Testing
- Analyze Code to find Security Problems
- Use Automated Testing Tools to Find Security Problems

Lesson 7: Maintaining Security in Deployed Software

- Monitor and Log Applications to Support Security
- Maintain Security after Deployment

Appendix A: Mapping Course Content to Cyber Secure Coder (Exam CSC-210)**PREREQUISITES**

This course presents secure programming concepts that apply to many different types of software development projects. Although this course uses Python®, HTML, and JavaScript® to demonstrate various programming concepts, you do not need to have experience in these languages to benefit from this course. However, you should have some programming experience, whether it be developing desktop, mobile, web, or cloud applications. Logical Operations provides a variety of courses covering software development that you might use to prepare for this course, such as:

- Python® Programming: Introduction
- Python® Programming: Advanced
- HTML5: Content Authoring with New and Advanced Features
- SQL Querying: Fundamentals (Second Edition)

WHO SHOULD ATTEND

This course is designed for software developers, testers, and architects who design and develop software in various programming languages and platforms, including desktop, web, cloud, and mobile, and who want to improve their ability to deliver software that is of high quality, particularly regarding security and privacy. This course is also designed for students who are seeking the CertNexus Cyber Secure Coder (CSC) Exam CSC-210 certification.

COURSE-SPECIFIC TECHNICAL REQUIREMENTS**Hardware**

For this course, you will need one computer for each student and one for the instructor. Each computer will need the following minimum hardware configurations:

- 1 GHz or faster 32-bit (x86) or 64-bit (x64) processor
- 2 gigabytes (GB) RAM (32-bit or 64-bit)

- 20 GB available hard disk space (32-bit or 64-bit)
- Keyboard and mouse (or other pointing device)
- 1,024 x 768 resolution monitor recommended
- Projection system to display the instructor's computer screen
- Local area network and Internet connection

Software

- Windows® 10/8.1/8/7/Vista (64-bit). This course was successfully keyed on Windows 10. Some activity steps may not key exactly as written if students key on a different version of Windows.
- Python version 3.8.1 (python-3.8.1.amd64.msi, provided with the course data files).
- PyCharm Community Edition version 2019.3.3 (pycharm-community-2019.3.3.exe, provided with the course data files). Python is distributed under the Python Software Foundation License (PSFL). PyCharm Community Edition is distributed under the Apache® License 2.0.
- If necessary, software for viewing the course slides. (Instructor machine only.)