Cyber Secure Coder (Exam CSC-110)

Course Length:
3 days

Overview:
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.

Target Student:
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 Logical Operations Cyber Secure Coder (CSC) Exam CSC-110 certification.

Prerequisites:
This course presents secure programming concepts that apply to many different types of software development projects. While 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. A variety of courses covering software development that you might use to prepare for this course, such as:

- *Developing Secure Universal Windows® Platform Apps in C# and XAML*
- *Developing Secure iOS® Apps for Business*
- *Developing Secure Android™ Apps for Business*
- *Python® Programming: Introduction*
- *Python® Programming: Advanced*
- *Programming Google App Engine™ Applications in Python®*
- *HTML5: Content Authoring with New and Advanced Features*
- *SQL Querying: Fundamentals*

Course Content

Lesson 1: Identifying the Need for Security in Your Software Projects
Topic A: Identify Security Requirements and Expectations
Topic B: Identify Factors That Undermine Software Security
Topic C: Find Vulnerabilities in Your Software
Topic D: Gather Intelligence on Vulnerabilities and Exploits

Lesson 2: Handling Vulnerabilities
Topic A: Handle Vulnerabilities Due to Software Defects and Misconfiguration
Topic B: Handle Vulnerabilities Due to Human Factors
Topic C: Handle Vulnerabilities Due to Process Shortcomings

Lesson 3: Designing for Security
Topic A: Apply General Principles for Secure Design
Topic B: Design Software to Counter Specific Threats

Lesson 4: Developing Secure Code
Topic A: Follow Best Practices for Secure Coding
Topic B: Prevent Platform Vulnerabilities
Topic C: Prevent Privacy Vulnerabilities

Lesson 5: Implementing Common Protections
Topic A: Limit Access Using Login and User Roles
Topic B: Protect Data in Transit and At Rest
Topic C: Implement Error Handling and Logging
Topic D: Protect Sensitive Data and Functions
Topic E: Protect Database Access

Lesson 6: Testing Software Security
Topic A: Perform Security Testing
Topic B: Analyze Code to find Security Problems
Topic C: Use Automated Testing Tools to Find Security Problems

Lesson 7: Maintaining Security in Deployed Software
>	**Topic A:** Monitor and Log Applications to Support Security
>	**Topic B:** Maintain Security after Deployment

Appendix A: Mapping Course Content to Cyber Secure Coder (Exam CSC-110)

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 recommended

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 2.7.13 (python-2.7.13.amd64.msi, provided with the course data files).
- PyCharm Community Edition version 2017.1.1 (pycharm-community-2017.1.1.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.)