General disclaimer

This document presents details about the training offerings from Synopsys at the time of its creation. Synopsys has used reasonable efforts to ensure that the information provided in this document is accurate and up-to-date, but details and offerings are subject to change.

This document contains confidential information about Synopsys and its businesses. Copies of this document may only be provided, and disclosure of the information contained in it may only be made, with written prior agreement from Synopsys.

Ownership and disposal

The information contained in this document is owned by Synopsys. The recipient shall dispose of the data as confidential waste and/or return the document to Synopsys upon request.

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior.

For more information about the Synopsys Software Integrity Group, visit us online at www.synopsys.com/software.

Synopsys, Inc.
185 Berry Street, Suite 6500
San Francisco, CA 94107 USA

U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com
# Table of contents

- **Introduction** .................................................. 1
- **Our curriculum** ................................................ 1
- **In-person or virtual: Your choice** .......................... 1
- **Training for every level** ...................................... 2

## Introductory .................................................... 3
- **Attack and Defense** ......................................... 4
- **OWASP Top 10** ................................................ 4
- **Principles of Software Security** ........................... 5

## Defending ....................................................... 6
- **Defending Android** .......................................... 7
- **Defending .NET Core** ....................................... 7
- **Defending C# ASP.NET** ..................................... 8
- **Defending C or C++** ......................................... 8
- **Defending Golang** ........................................... 9
- **Defending HTML5** ........................................... 9
- **Defending iOS** ................................................ 10
- **Defending Java EE** .......................................... 10
- **Defending Java SE** .......................................... 11
- **Defending JavaScript** ....................................... 11
- **Defending Python** .......................................... 12
- **Principles of Software Security for COBOL** ............ 12

## Securing ........................................................ 13
- **Securing Mobile Platforms 101** ........................... 14
- **Securing Mobile Platforms 201** ........................... 14
- **Securing AWS** ................................................ 15
- **Securing Azure** .............................................. 15
- **Securing Containers with Docker** ......................... 16
- **Securing APIs using OAuth 2.0** ............................ 16
- **Securing SAML** .............................................. 17

## Attacking ....................................................... 18
- **Attacking Code Using Static Analysis** .................... 19
- **Attacking Networks** ........................................ 19
- **Attacking Web Applications** .............................. 20

## Workshops ...................................................... 21
- **Red Teaming** ................................................. 22
- **Threat Modeling** ............................................. 22
- **Securing Software with DevSecOps** ....................... 23
- **Embedded Systems Security** ............................... 23
Introduction

Synopsys’ instructor-led courses are developed and taught by experts at the forefront of the software security field. Our instructors are certified security professionals who have hands-on experience working directly with clients on their security challenges.

Our curriculum includes training modules for professionals just starting out with software security as well as those who are looking to develop more advanced skills. Synopsys continuously develops its courses to accommodate the rapid changes in software security.

Our curriculum

Synopsys’ curriculum is a series of complementary courses designed to meet your organization’s needs. You can select the courses that best match your level of experience, your role, and the development platforms in your organization. Our courses are grouped into the following software security activities:

2. Defending. Learn defensive programming skills in context, in specific languages targeted to specific development platforms, so you can defend against attacks in your code.
3. Securing. Learn how to identify common vulnerabilities and essential strategies to secure your deployment and tech stack from external threats.
4. Attacking. Use your knowledge to test your applications for security vulnerabilities.

We can work with you to select a curriculum that is right for your organization.

In-person or virtual: Your choice

If you choose traditional instructor-led training, our certified instructors will travel to the location of your choice. Our instructors are trained to engage your audience through group discussion and interactive hands-on labs designed to simulate real-world environments. On-site instructors can make course adjustments to better complement the needs, interests, and experience level of your participants.

If you have a distributed workforce, your participants can avoid travel and time away from the office using our Virtual Instructor-Led Training (VILT). VILT is separated into shorter sessions to optimize participant engagement. VILT can be delivered over consecutive working days or on a weekly basis, depending on your team's preference. Virtual training is a cost-conscious alternative for supporting your employees’ professional development.

Instructor-led courses are held on your schedule in the format that works best for you.
Training for every level

Synopsys' software security curriculum provides valuable knowledge across every role within software development organizations. Our instructor-led training features a broad library of over 30 courses, so you can design a long-term plan to increase the security knowledge and skills of everyone within your SDLC.

Below, you’ll find some sample learning paths for developers, engineers, and architects. Pick and choose the courses your developers need, or design your own learning path—it's up to you.
Introductory
Attack and Defense

Description
This course provides software builders and testers an in-depth look at standard attacks and their corresponding defenses. It empowers students of this course to solve tricky problems securely in their own environment by mapping them to known problems and tried and tested solutions.

This course introduces common attacks that most applications deal with. These attacks are cast into different contexts such as web, embedded, thick client, or mobile, and their standard solutions are discussed in the classroom. Students are then guided to apply this knowledge to identify attacks and design defenses for a model application throughout the labs.

OWASP Top 10

Description
This course focuses on the most important security defects found in web applications, covering all issues in the latest OWASP Top 10 list. Each topic describes a vulnerability and provides practical guidance for remediation. This course also provides demonstrations and practical hands-on exercises where students learn what impact these security issues can have on web applications.
Principles of Software Security

Description
Principles of Software Security is a 4-hour fundamentals course delivered virtually to developers, quality assurance, testing, and information systems’ security teams. This course first delves into identifying current software security problems, then addresses the issues by explaining how to infuse software security into the development process early on. It elucidates the Synopsys concept of “Build Security In” as opposed to relying solely on traditional security and testing practices.

This virtual classroom course is available in two flavors:
- The Vanilla flavor explains common problems in software security and describes an approach to infusing software security into the development process through risk management, software security touchpoints, and historical knowledge of software security vulnerabilities.
- The Requirements flavor focuses on introducing important cost-saving software security requirements early in the software development life cycle.

Intended Audience
• Developers
• Architects
• QA Engineers
• Security Practitioners

Delivery Format
• Virtual Classroom

Class Duration
• 4 hours
Defending
Defending Android

Description
This course begins with foundational knowledge about the Android platform, its architecture, and the security model, then builds on it to discuss Android-specific risks.

The core of the course delves into defensive programming techniques for the Android platform and other common application security risks. Defensive programming techniques are explored within the framework of the OWASP Mobile Top 10 Security Project. The risk itself, code which implements the risk, and time-tested techniques for addressing these risks are examined here.

Defending .NET Core

Description
This course is focused on modern .NET Core secure development with an emphasis on microservices, service-oriented architecture and cloud-first applications. Students will learn modern attacker techniques and how to defensively write code to prevent these vulnerabilities in applications. This course also discusses different activities that can be performed during the secure development life cycle (SDLC) that help detect and prevent the vulnerabilities.
Defending C# ASP.NET

Description
This course explores in detail the root cause and remediation of vulnerabilities that apply to ASP.NET C# web applications. It also discusses different activities that can be performed during the secure development life cycle (SDLC) that help detect and prevent the vulnerabilities such as penetration testing and code review.

Defending C or C++

Description
This course provides developers with a strong foundation in software security as it relates to the implementation of applications developed in C or C++. It includes detailed examples and focuses on the correct way to think through security problems by combining structured theory, demonstrations, technical deep-dives, and illustrated explanations. This course connects the habit of building security in through proven programming practices and explains common security-related problems in detail so that students can avoid them in their own work. The course can be delivered either as a C flavor or C++ flavor, but not both.
Defending HTML5

Description
As HTML5 gains relevance in enterprises for its rich features and enhanced online experience, developers need to consider the vulnerabilities that their new code may introduce into their organizations’ web infrastructure. The Defending HTML5 course addresses this concern by identifying risks in the new HTML5 landscape and explaining how to mitigate them with defensive programming strategies and best practices.

Defending Golang

Description
Defending Golang is a full-day course that focuses on developing secure systems software and web applications in Golang. It helps attendees understand generic risks in highly performant, concurrent systems software, as well as specific risks for Golang developers and the specific remedies available to them. This course also explains risks present in developing web applications and web services in Golang. Using real vulnerabilities from production open source projects, this course demonstrates the real-world complexity of security risks and remediation.
Defending iOS

Description
The course begins with an overview of the iOS platform, the securities that are built in, and how they have evolved over the many iterations of the iOS operating system. Next, the course walks through common iOS application security concerns and discusses how best to mitigate or remediate such issues. The course looks at the risk, the code that implements the risk, as well as code examples for the issues and remediation steps.

NOTE: Customers should select the programming language relevant to them—either Objective-C or Swift—when ordering this course.

Defending Java EE

Description
This course focuses on using defensive programming techniques in Java EE (JEE) applications against common web vulnerabilities. It discusses an approach to identify security risks and vulnerabilities, apply defensive programming techniques, and securely configure web applications. This course also provides demonstrations and practical hands-on exercises where students learn how to identify security vulnerabilities in the code and fix them using best practices discussed in the course.
Defending Java SE

Description
The Java Standard Edition (JSE) platform comes with a comprehensive feature set that allows developers to create and deploy Java applications on desktops, servers, and embedded devices. This course examines security aspects of the desktop and client-server software architectures, and aims to eliminate improper trust assumptions that developers commonly make. The core of this course teaches secure coding techniques designed to mitigate security vulnerabilities that affect software in general, as well as issues that are specific to the Java platform.

Intended Audience
• Developers
• DevSecOps
• Architects

Delivery Format
• Traditional Classroom
• Virtual Classroom

Class Duration
• 8 hours

Defending JavaScript

Description
Defending JavaScript is a full-day course that addresses the questions of secure development in front-end and back-end JavaScript. It helps students understand generic web application risks as well as specific risks involved in manipulating JavaScript in the DOM (Document Object Model) on the client side, bypassing browser controls such as same origin policy and sandboxing, sending Ajax requests, and using client-side frameworks and libraries.

This course also explains risks present in server-side code written in JavaScript, and covers security best practices, different types of injections, framework misconfigurations, cross-site request forgery, insecure deserialization, output encoding, and input validation.

NOTE: Customers should select the client-side framework relevant to their development environment—either Angular, AngularJS, or React—when ordering this course. Node.js and Express.js server-side frameworks are available with any client-side framework selection.
## Defending Python

### Description
Python is an extensively used general-purpose, high-level programming language designed to support rapid development, prototyping, and fast iterative development. This course covers common security vulnerabilities in both Python as well as in common Python frameworks.

Django and Flask are two popular Python web frameworks that leverage a model, view, template architecture. They both offer developers pre-built functionality within the core components or as extendable third-party plugins. However, because the framework APIs, configurations, and security features differ, the framework-specific portions of this course are offered in either the Django or Flask flavors depending on the target audience.

*Note: Customers should select either the Django or Flask flavor while ordering this course.*

### Intended Audience
- Developers
- DevSecOps
- Architects

### Delivery Format
- Traditional Classroom
- Virtual Classroom

### Class Duration
8 hours

## Principles of Software Security for COBOL

### Description
Software in any development and execution environment is subject to intrusion. This is certainly true of the Common Business Oriented Language (COBOL) mainframe environment. The course has two main sections. The first section, "Foundations of COBOL Security," looks at security risks to your company and how your code might contribute to that risk. The second section, "COBOL Defensive Programming Techniques," is the core of this course. It looks at COBOL programming best practices and then talks about how to avoid or mitigate the vulnerabilities described in the taxonomy of vulnerabilities. This includes examples of bad code snippets and mitigation for each vulnerability with secure coding examples to avoid the vulnerability.
Securing
Securing Mobile Platforms 101

Description
The course contains three modules:
• Mobile Trends: Mobile First AppSec
• Overview of Mobile Platforms
• Secure Development

This course is aimed at users who are new to mobile, new to mobile security, or have had brief exposure to such topics. It covers areas such as common mobile vulnerabilities and mitigating controls, different app architecture types, and more.

Securing Mobile Platforms 201

Description
This course contains three modules:
• Mobile Payments
• Protecting Client-Side Code
• Protecting Business Critical Applications

This course is aimed at users with some understanding of mobile platforms who want to gain an understanding of advanced security concepts. It places particular emphasis on applications that are highly sensitive or are considered business critical. This course also delves into tampering of mobile apps, jailbreaking, rooting, the use of debuggers, and more.
Securing AWS

Description
Cloud computing has grabbed the world's attention not only for its pervasive, on-demand, convenient usage, but for its ability to be vulnerable to data breaches and novel forms of attack. Since most software uses the cloud in various shared capacities (development, hosting, or integration with third-party code), threats from hackers are inevitable. This hands-on workshop equips students to understand this new landscape of converged infrastructure and shared services, its existing and emerging threats, and provides them with secure mitigation methods.

The Securing AWS course is an introductory course, covering Amazon core services, such as IAM, KMS, EC2, S3 and VPCs, with a focus on security. This course enables students to identify areas for cross-pollination between development and operations that enhance application, infrastructure, and network security.

Securing Azure

Description
In this course, students learn how to secure Azure Infrastructure as a Service (IaaS). This course initially presents a brief overview of the different Azure infrastructure offerings, and then dives into how to secure them. In hands-on labs, students learn how to spot an insecure Azure configuration and subsequently fix it. The course covers the following topics:

- Identity and Access Management
- Storage Security
- Network Security
- Compute Security

This course is aimed at users of all levels of familiarity with Azure, and does not require any prior Azure knowledge.

NOTE: This course is not intended to teach cloud or Azure concepts and offerings. It does, however, provide a brief introduction to several important infrastructure components in Azure, and focuses on how to secure them.
Securing Containers with Docker

Description
Containers have changed the way applications are being deployed these days. “Containerization” has gained traction over the years because it easily enables an application team to build, package and distribute a microservice or an application across different environments. Docker has emerged as the leading container technology that is used by organizations large and small for packaging and deploying these services or applications.

However, as always, security is a challenge that organizations face when it comes to deploying containers securely. Container security refers to the protection of the integrity of the containers which include both the application as well as the infrastructure they make use of. This course features hands-on labs, best practices, and instruction that will enable students to harden the container runtime and the container host.

Securing APIs using OAuth 2.0

Description
Synopsys’ Securing APIs using OAuth 2.0 teaches students how to control access to APIs using OAuth tokens. The course explains each OAuth grant type, the intended use case, and common attacks. Architectural design flaws as well as common implementation bugs related to OAuth authorization servers, resource servers, clients, and tokens are all discussed in this course. The course also highlights the dangers of using plain OAuth for authentication and explains the need for technologies like OpenID/Connect. The content covered in the course is reinforced with instructor-led demos and hands-on group lab exercises.
Securing SAML

Description
The Security Assertion Markup Language (SAML) standard is a vital component in enterprise single sign-on and federated identity systems. SAML allows multiple applications, APIs, or cloud providers to use a central identity provider to handle user logins, thus giving an organization centralized control over user onboarding, access control, and logging. However, given its key role in controlling access to an organization's resources, incorrect use of SAML can have serious security consequences. This course introduces students to common security issues in SAML implementations and how to avoid them.
Attacking
Attacking Code Using Static Analysis

Description
This course focuses on the static analysis process and tools that can be used to test and attack a web application. It explains the static analysis techniques, compares manual and automated code reviews, and discusses the implementation of static analysis in your SDLC.

The course also provides demonstrations and practical hands-on exercises where students learn how to identify common vulnerabilities using code review and common static analysis tools.

Attacking Networks

Description
Attacking Networks is a full day, lecture and lab course aimed at understanding the security posture of a network and how best to uncover its vulnerabilities.

The first part of this course introduces students to network security testing, then discusses a structured approach for performing tests using tools. The next part of the course is dedicated to software exploits, advanced testing skills, and post-exploitation activities. The final part of the course explains how to document and communicate findings from an assessment. Labs are performed throughout the course to tie concepts to the real world.
Attacking Web Applications

Description
The Attacking Web Applications course explains how to test for security issues in web applications. It defines what web security testing is and how it differs from other forms of testing, describes what the testing process looks like, and gives specific guidance on how to test for some of the most important risks in web applications.
Workshops
Red Teaming

Description
Red Teaming is a goal-based assessment approach which allows organizations to gain insight into how their security posture is when faced with a real threat. This hands-on, two-day course introduces students to the concepts of red teaming and how it is different from traditional vulnerability testing. The course also includes guidance for the organization on creating and maintaining its own internal red teams. Students in this course are introduced to physical, social, and electronic testing methods which can be utilized during red team engagements.

Intended Audience
- Developers
- Architects
- Security Practitioners

Delivery Format
- Traditional Classroom

Class Duration
- 8 or 16 hours

Threat Modeling

Description
Discovering weaknesses in the design of a system is the specific goal of threat modeling. Synopsys’ threat modeling approach can reveal security issues not fully addressed by the traditional methods of penetration testing and secure code review. Organizations benefit from this software design analysis because you can perform it without code to discover potential vulnerabilities early in the development cycle.

New for 2020 is a flavor which adds a microservices emphasis to the course. Threat Modeling with Microservices brings to light potential weaknesses in the design of Microservices-oriented architectures. During this course, we will highlight common problem patterns and solutions in both the services and supporting infrastructure.

This lab-reinforced course provides a deep-dive into system threat models with topics such as decomposing and modeling the system, inventorying services, infrastructure, and data stores, modeling attack possibilities, evaluating and interpreting the threat model and creating a traceability matrix.
Securing Software with DevSecOps

Description
Securing Software with DevSecOps is an introductory course aimed at understanding DevSecOps’ key concepts, roles, benefits, challenges and deployment. Differences between agile, continuous integration/continuous delivery (CI/CD), and DevSecOps are explored in this course. DevSecOps sample pipeline demos and case studies enrich the course to make it a complete learning experience for students.

Intended Audience
• Developers
• DevSecOps
• Architects
• QA Engineers
• Security Practitioners

Delivery Format
• Traditional Classroom
• Virtual Classroom

Class Duration
• Virtual: 4 Hours
• Traditional: Varies

Embedded Systems Security

Description
Embedded Systems Security: Designing for Security provides an introduction into security engineering for professionals that develop embedded, IoT (Internet of Things) or other integrated systems. The course content is geared towards those students who have a firm understanding of the principles of designing, engineering, or developing non-IT systems and seek to understand the influence of security as a stakeholder in design.

Students will be provided with a base understanding of cyber security as it relates to various systems and the processes that should be present within their engineering life cycles. The course takes the approach of understanding risks and vulnerabilities typically present in these systems, and outlining processes and techniques to assist in developing software and embedded systems to minimize cyber security risk.

Intended Audience
• Developers
• DevSecOps
• Architects
• QA Engineers
• Security Practitioners
• Engineers
• Engineering Managers

Delivery Format
• Traditional Classroom

Class Duration
• Varies