The Software Architecture Professional Certificate program will equip you with state-of-the-art architecture practices and concepts.

You will gain experience in effective architecture documentation, design, and analysis techniques.

Software Architect Training
Software architecture determines how you communicate design decisions, structure work breakdowns, and create software product lines. The SEI offers the Software Architecture Professional Certificate to equip software professionals with state-of-the-art practices for designing, documenting, evaluating, and implementing software architectures.

Since 1984 the Software Engineering Institute (SEI) has been identifying, developing, and advocating practices for designing high-quality software and protecting networked systems.

The program is a collection of five courses that equips software professionals with best practices, so they can efficiently design software-reliant systems that meet their intended business and quality goals. The courses are supported by four widely acclaimed books in the SEI Addison-Wesley Series.

Successful completion of the Software Architecture: Principles and Practices exam is also required to qualify for the certificate.

Requirements
A software professional earns the SEI Software Architecture Professional Certificate by successfully completing the

- Software Architecture: Principles and Practices course (classroom or eLearning)
- Software Architecture Design and Analysis course (classroom)
- Documenting Software Architectures course (classroom or eLearning)
- Software Product Lines course (classroom)
- Software Architecture: Principles and Practices Exam

This four-course sequence provides both breadth and depth in presenting software architecture concepts and practices. Beginning with an introduction to software architecture fundamentals, participants gain experience in effective architecture documentation, design, and analysis techniques, and then learn how these techniques can be put to advantage in adopting a product line approach to software.

We recommend that certificate candidates begin with the Software Architecture: Principles and Practices course, which is available as instructor-led classroom training and as eLearning. The courses that follow build on the concepts presented and may be taken in any sequence.

There is no fee for the certificate. There are fees associated with the requisite courses and the examination. Additionally, there is no expiration for the certificate.
The Software Architecture: Principles and Practices Course
This two-day course will provide insight into ways to use software architecture successfully in your software-reliant system development. Case studies illuminate the key technical and business issues regarding software architectures. This course is based on the book Software Architecture in Practice, 2nd Edition, and is available as a classroom or an eLearning course.

Software Product Line Course
This course introduces the basic concepts of software product lines and provides an overview of the essential technical and management practices needed to succeed with software product lines. Topics include the costs and benefits of using a product line approach, activities and practice areas, a phased adoption approach, and case studies of organizations that have achieved success with software product lines. This course is based on the book Software Product Lines: Practices and Patterns.

Software Architecture Principles and Practices Exam
The Software Architecture: Principles and Practices Examination provides an objective validation of the candidate’s knowledge and understanding of software architecture. Candidates for the Software Architecture Professional Certificate must achieve a passing score of 75% in order to qualify for the certificate.

This requirement accords with industry standards for professional certificates, which require that candidates pass an examination based on a set of uniform standards after taking a required sequence of professional development courses.
You will learn the essential concepts and practices involved in using software architecture effectively.

Although the term software architecture is used frequently in today's software industry, its meaning is not universally understood.

- What is software architecture?
- How do you use software architectures in practice?
- What does a software architect do for an organization?
- What value does software architecture provide?

This course introduces the essential concepts of software architecture. A software architecture is an abstract view of a software system distinct from the details of implementation, algorithms, and data representation. Architecture is, increasingly, a crucial part of a software organization's business strategy. Software architectures can

- provide flexibility and adaptability in changing markets
- allow for interoperability with other players in the marketplace
- provide leverage of control in a marketplace
- help developers focus on a niche in the marketplace
- be used as a sales and marketing tool
- help reduce maintenance costs and amortize development costs
- assist in workforce organization and with project oversight and control
- establish a common corporate vocabulary
- shorten learning time

This two-day course emphasizes the importance of the business (or mission) context in which systems are designed and introduces participants to software architectures in a real-world setting. "Industrial-strength" case studies illuminate the key technical and organizational issues regarding software architectures. This course is based on the book *Software Architecture in Practice, 2nd Edition*.

**Who should attend?**
This course is targeted at those professionals who design, develop, or manage the construction of software-reliant systems. This course is designed to help practicing software professionals quickly gain insight into the latest concepts of what software architecture is and how to use it successfully.

**Topics**
- definition and overview of software architecture
- the architecture business cycle: what influences software architects and software architecture
- understanding and achieving quality attributes
- attribute-driven design
- documenting software architecture
- evaluating software architecture
- architecture reuse
Objectives
This course provides attendees with a thorough overview of software architectures. After attending this course, participants will have a better understanding of
- the relationships between system qualities and software architectures
- software architectural patterns and their relationship to system qualities
- software architecture evaluation
- attribute-driven design
- software architecture documentation
- architectural reuse

This course is the most fundamental of the six courses in the SEI Software Architecture Curriculum. At the conclusion of the course, attendees will be eligible to take the SEI's Software Architecture Design and Analysis and Architecture Tradeoff Analysis Method (ATAM) Evaluator Training courses. Attendees will also be better prepared for the SEI's Documenting Software Architectures and Software Product Lines courses. This course is the first course required in the SEI's Software Architecture Professional, SOA Architect Professional, and ATAM Evaluator certificate programs.

Materials
Participants receive a copy of the lecture slides and the book *Software Architecture in Practice, 2nd Edition*.

Schedule
This 2-day course meets at the following times:
Days 1–2, 8:30 a.m.–5:00 p.m. (U.S. locations)
Days 1–2, 9:00 a.m.–5:30 p.m. (non-U.S. locations)

Related Web Sites
www.sei.cmu.edu/training/p35.cfm
www.sei.cmu.edu/training/certificates/architecture

For Course Registration
www.sei.cmu.edu/training/registration/

For General Information
For information about the SEI and its products and services, contact
Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
customer-relations@sei.cmu.edu
www.sei.cmu.edu

Prerequisites
Before registering for this course, participants should have
- experience in the development of software-reliant systems
- some familiarity with modern software engineering concepts
You will learn how to document architecture to communicate effectively with all project stakeholders.

Software architecture has become a widely accepted conceptual basis for the development of nontrivial software in all application areas and by organizations of all sizes. However, the treatment of architecture to date has largely concentrated on its design and, to a lesser extent, its validation. Effectively documenting an architecture is as important as crafting it; if the architecture is not understood (or worse, if it is misunderstood), it cannot be analyzed effectively or meet its goal as the unifying vision for system and software development. Documentation-based architecture strategies stop short of prescribing documentation standards.

- How do you document architectures using well-known styles such as service-oriented, client-server, multi-tiered, layered, and so forth?
- What are effective architecture documentation guidelines?
- How do you represent architectural elements and the relations among them?
- How do you document interface semantics and architectural rationale?
- How do you provide relevant architectural information to important stakeholders?
- Are there templates for architecture documentation?

This two-day course provides in-depth coverage of effective software architecture documentation practices that meet the needs of the entire architecture stakeholder community. This course presents the information in the context of prevailing prescriptive models, including the Rational Unified Process (RUP), the Siemens Four Views software approach, the IEEE 1471-2000 standard, and the Unified Modeling Language (UML). The course is based on the book *Documenting Software Architectures: Views and Beyond*.

**Who should attend?**
This course is targeted at
- software architects and software lead designers whose jobs include producing architectural documentation
- software technical managers whose jobs include overseeing and/or managing the architecture definition process
- software engineers who may be expected to use architecture documentation

**Topics**
- principles of sound documentation
- viewtypes, styles, and views
- advanced concepts such as refinement, context diagrams, variability, software interfaces, and how to document interfaces
- documenting the behavior of software elements and software systems
- choosing relevant views
- building a documentation package using a seven-part template

**Objectives**
This course shows software architects how to produce a comprehensive documentation package for a software architecture that is useful to stakeholders.
After attending this course, participants will have a better understanding of

- the basic principles of sound technical documentation
- a stakeholder- and view-based approach to documenting software architectures
- which views are available for documenting an architecture
- how to choose the set of views that will be most valuable to the architecture’s community of stakeholders
- the information needed to document a view
- how to use formal and informal notations (including UML) to represent elements and relations in a view
- how to document a software interface and software behavior
- the information needed to document information that applies across views

**Materials**
Participants receive a copy of the lecture slides, exercises, and the book
*Documenting Software Architectures: Views and Beyond.*

**Schedule**
This 2-day course meets at the following times:
- Days 1–2, 8:30 a.m.–5:00 p.m. (U.S. locations)
- Days 1–2, 9:00 a.m.–5:30 p.m. (non-U.S. locations)

**Prerequisites**
Before registering for this course, participants must

- have experience in designing and developing software-intensive systems
- understand the basic concepts of software architecture. If desired, they can gain this understanding by completing the Software Architecture: Principles and Practices course, which is available as instructor-led classroom training and as eLearning
Software Architecture Design and Analysis

A system's software architecture is widely regarded as one of the most important software artifacts. Software professionals routinely make decisions that impact that architecture, yet many times that impact is not fully considered or well understood.

- Which design decisions will lead to a software architecture that successfully addresses the desired system qualities?
- How do you know if a given software architecture is deficient or at risk relative to its target system qualities?

This two-day course provides in-depth coverage of the concepts needed to effectively design and analyze a software architecture. The essential considerations for defining any architecture are carefully examined and then illustrated through application of the SEI Attribute-Driven Design (ADD) software architecture design method. This course also explores architecture analysis in-depth and introduces the SEI Quality Attribute Workshop (QAW) and the SEI Architecture Tradeoff Analysis Method (ATAM). Through multiple exercises, participants study an application of these methods and get a chance to apply them to sample problems.

This course is based on the books *Software Architecture in Practice, 2nd Edition* and *Evaluating Software Architectures: Methods and Case Studies*.

The prerequisite for this course is the Software Architecture: Principles and Practices course, which is available as instructor-led classroom training and as eLearning.

Who should attend?

- practicing software architects
- designers and developers of software-reliant systems

Topics

- life-cycle view of architecture design and analysis methods
- the QAW, a method for eliciting critical quality attributes, such as availability, performance, security, interoperability, and modifiability
- the ADD method, a method for designing a software architecture
- the ATAM, a method for evaluating a software architecture based on a set of attribute-specific measures of the system such as performance, availability, modifiability, and security

Objectives

This course provides attendees with in-depth coverage of the concepts needed to effectively design and analyze a software architecture. After attending this course, participants will have a better understanding of

- the essential considerations in any architectural design process
- the QAW for eliciting critical quality attributes
SEI Training
Software Architecture Design and Analysis

- the ADD method for designing an architecture
- the role of architecture evaluation
- using the methods within a software development life cycle

Prerequisites
Before registering for this course, participants must
- complete the Software Architecture: Principles and Practices course, which is available as instructor-led classroom training and as eLearning

Materials
Participants will receive a copy of lecture slides and exercises.

Schedule
This 2-day course meets at the following times:
- Days 1–2, 8:30 a.m.–5:00 p.m. (U.S. locations)
- Days 1–2, 9:00 a.m.–5:30 p.m. (non-U.S. locations)

Related Web Sites
www.sei.cmu.edu/training/p34.cfm
www.sei.cmu.edu/training/certificates/architecture

For Course Registration
www.sei.cmu.edu/training/registration/

For General Information
For information about the SEI and its products and services, contact Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
customer-relations@sei.cmu.edu
www.sei.cmu.edu
You will learn the basic concepts and practices involved in software product lines.

A software product line is a set of software-reliant systems that share a common, managed set of features satisfying a particular market or mission area and that are built from a common set of core assets in a prescribed way. Producing a set of related products as a product line has allowed organizations to achieve increased quality and significant reductions in cost and time to market. But adopting a product line approach to software is both a technical and a business decision that involves many challenges.

- What are the costs and benefits of a product line approach for your organization?
- How do you manage the production and use of core assets across a product line?
- How do you handle software engineering, technical management, and organizational management issues associated with software product line engineering?
- How do you structure your organization to coordinate the production of core assets and products?
- What are your organization's particular strengths and challenges with respect to a product line approach?
- How do software product lines relate to other current technology and business trends?

This two-day course introduces the world of software product lines and the basic concepts behind it. The course also provides an overview of the essential technical and management practices needed to succeed with software product lines, as well as case studies, guidelines, and patterns for applying product line techniques. This course, which is also included in the SEI Software Architecture Curriculum, is based on the book *Software Product Lines: Practices and Patterns*.

The goal of this course is for participants to understand the fundamental concepts and practices involved in software product lines.

**Who should attend?**

- software engineers and technical managers who are interested in effective reuse strategies
- software engineers and technical managers who are adopting or using a software product line approach

**Topics**

- fundamental concepts of software product lines
- the costs and benefits of using a product line approach
- essential product line activities
- software product line practice areas
- software product line practice patterns
- the SEI Product Line Technical Probe
- case studies of organizations that have achieved success with software product lines
- a phased approach for adopting software product lines
Objectives
This course provides attendees with a thorough overview of software product lines including:

- the essential activities involved in fielding software product lines
- the costs and benefits of adopting a product line approach
- the software engineering, technical management, and organizational management practices necessary for achieving successful software product lines
- product line practice patterns that aid in product line adoption
- a product line diagnostic method and an adoption roadmap
- how a product line approach can be combined with other technology and business trends
- what product line practice patterns best apply to your organization
- a path to software product line adoption

Prerequisites
Before registering for this course, participants must have:

- experience in designing and developing software-reliant systems
- some familiarity with modern software engineering concepts and management practices

In addition, an understanding of basic software architecture concepts is recommended. Participants can gain it by completing the Software Architecture: Principles and Practices course, which is available as instructor-led classroom training and as eLearning.

Materials
Participants receive a copy of lecture slides, exercises, and the book *Software Product Lines: Practices and Patterns.*

Schedule
This 2-day course meets at the following times:

- Days 1–2, 8:30 a.m.–5:00 p.m. (U.S. locations)
- Days 1–2, 9:00 a.m.–5:30 p.m. (non-U.S. locations)

Related Web Sites
www.sei.cmu.edu/training/p36.cfm
www.sei.cmu.edu/training/certificates/architecture/

For Course Registration
www.sei.cmu.edu/training/registration/

For General Information
For information about the SEI and its products and services, contact
Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
customer-relations@sei.cmu.edu
www.sei.cmu.edu