

Software Engineering and Quality

Free Program Overview

Software Development (including Software Engineering and Quality)

This free event is an informal information session for new or returning students who are interested in our software development programs, including Computer Programming, Internet Programming and Development, Software Engineering and Quality, Linux Programming and Administration, and Database Systems. During this interactive session, instructors and staff will highlight new content developments in each area. They will also explain the curriculum, requirements, recommended course sequences, and certificate completion timeframes. You will learn how these programs and help you advance your current career or break into a new field.

**Course 3085
and 22404**



ENGINEERING AND TECHNOLOGY



Comprehensive Training for Silicon Valley Software Developers

The Software Engineering and Quality certificate program at UCSC Extension is a program for Silicon Valley software developers who want to grow in their specific areas of expertise and broaden their skills. It blends foundation-level coursework in software engineering with cutting-edge developments in software quality, testing, and software development practices.

Unique Curriculum Tailored to Three Areas of Expertise

The program is unique in that it brings together software engineering, testing and management courses under a single umbrella. This provides our students with an integrated view across key disciplines so that they better understand their position in the overall software development process. This broad perspective is also helpful in identifying new career options. Areas of focus include:

- Engineering—covers software architecture and software modeling
- Quality and Testing—offers courses in the state-of-the-art testing strategies and quality assurance methods
- Software Development Practices—covers the practices and skills required to manage the entire software product life cycle

Benefits of Studying at UCSC Extension

- Unique and practical courses enhance job skills for software professionals
- Certificate signifies advanced knowledge and accomplishment
- Courses are taught by expert working professionals
- Instructor-led learning includes interactive Q and A
- University of California quality reputation
- Networking with fellow students
- Convenient course schedule and Silicon Valley location

Who Should Attend This Program

The Software Engineering program at UCSC Extension Silicon Valley is geared toward professional engineers seeking to expand their know-how and career prospects. Recent engineering graduates who want a leg up as they enter the job market also benefit from the practical training our programs provide, which is not typically part of academic engineering curricula. Senior developers or project leaders can learn the best software practices and modeling techniques to help them manage software projects to successful completion. SQA and testing professionals can acquire hands-on knowledge using the latest testing tools and methodologies



About UCSC Extension Silicon Valley

The vital learning community at UCSC Extension Silicon Valley is well known for its collegial atmosphere and rigorous preparation. Our faculty of expert practitioners teaches state-of-the-art solutions to the everyday problems confronting technology professionals working in Silicon Valley. The professional education programs we offer build expertise, open doors to new opportunity, and deliver tangible value. Our broad portfolio of open-enrollment courses and certificates, affordable pricing, experience-based instruction, and central location in Silicon Valley help turn jobs into careers.

Software Engineering and Quality Certificate

Certificate Requirements

Total: 14 units

Required: Choose from one of three core courses

GPA: 3.0 (C or better in all courses)

Prerequisites

A degree in computer science or engineering or equivalent experience in software development, testing, QA or project management is required. For details see full course descriptions at ucsc-extension.edu/engineering.

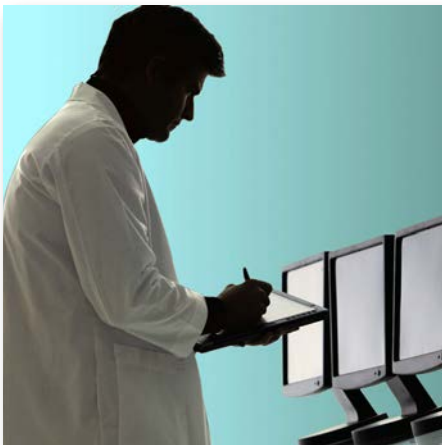
Recommended Course Sequence

We recommend you take courses in the order presented on the course chart. However, the sequence may vary based on student background and professional interest.

Note: Some courses may be listed in more than one program, however, only one course may be shared between two Engineering and Technology certificate programs unless otherwise noted.

Program Contact

Engineering and Technology Department,
(408) 861-3860 or e-mail
program@ucsc-extension.edu



Curriculum

14-unit minimum

(*choose one of these three core courses)

Software Engineering

Object-Oriented Analysis and Design*	3.00774
Object-Oriented Development Architectures and Design Patterns, Advanced.....	3.06633
Refactoring with Design Patterns	1.56427
Data Structures and Algorithms Using C++	3.04732

Software QA and Testing

Software Quality Assurance and Testing*	2.03396
Software Testing: Techniques, Tools and Practices	3.020501
Software Security Testing	1.020280
JUnit Test Framework	1.56198

Professional Software Practices

Professional Software Development Fundamentals*	3.022868
User Research: User Needs and Usability Assessment for Web and Software Products.....	2.020079
User Experience Design for Web Applications	2.03113
Agile Software Development	1.520285
Managing Software Projects.....	1.50943

Refresher for Software Professionals

Programming with Java for Beginners	1.55185
C# .NET Programming Comprehensive	3.05408
C Programming for Beginners	3.05208
Perl Programming I	2.02856
Python Programming for Beginners	1.520776
Ruby, Introduction.....	0.521341
Relational Database Essentials	3.06195

Courses

Agile Software Development

Agile methodologies are changing the way software developers work. This course will equip you with the concepts, methods, practices and tools needed to use Agile in software development. It covers common Agile practices including extreme programming, pair programming, feature driven development, test driven development, continuous integration, code refactoring, and more. You will also learn how Agile can be applied to larger projects. To experience the real world use of Agile, there will be a laboratory session that simulates an Agile project.

Courses 20285

C Programming for Beginners

This course is for individuals who want to learn C programming language but have little or no programming background. The course begins with an overview of programming and tools. It introduces functions, data types, input/output, strings, operators, precedence, and expressions. It also demonstrates the use of control statements, arrays, and pointers for problem solving. You will receive assignments to write non-trivial programs and learn to create modular programs with efficiency and readability.

Course 5208

C# .NET Programming, Comprehensive

This course introduces beginning and intermediate programmers to .NET programming using Microsoft's C# programming language. The instructor explains the Visual Studio development environment and reviews the basic constructs of C# language with detailed explanations of the C# regular expressions, delegates, events, generics and collections. The course also covers exception handling, threading and synchronization. Sample applications will be used to illustrate core concepts and the instructor will present real-world code examples in class.

Course 5408

Course Descriptions

Data Structures and Algorithms Using C++

In this course, students learn to implement efficient algorithms with powerful data structure in object oriented code using C++. After a quick review of C++, the course delves into complex algorithms using examples. Students learn and implement industry standard container objects such as dynamic array, linked list, stack, heap, hash tables, heap and binary tree, search and sort, as well as graph algorithms using C++ templates. Emphasis is placed on common problems, implementation details, examples, and testing throughout the course.

Course 4743

JUnit Test Framework

JUnit is one of the most popular open source testing frameworks for all types of Java software applications. This course begins with the fundamentals of JUnit, including installation, setup and integration with Eclipse and Apache Ant, two of the major Integrated Development Environments (IDE) that support JUnit. Then the features of JUnit are covered, along with how different versions compare to each other, and how to customize JUnit and create automated tests. Students will learn the real-world uses of JUnit, including test strategies and the concept of Test Driven Development (TDD), also known as “test first, code later.” Other advanced topics include testing presentation layer, server side, JPA and database access. By the end of this course, you will have learned to use JUnit to test Java applications in all life cycle stages and will be able to apply this framework to build automation testing.

Course 6198

Managing Software Projects

According to a six-year study by The Standish Group, 75 percent of all software projects fail to meet schedule, budget, or scope requirements. To reverse this trend, this course teaches fundamental software project management techniques for students new to this field or interested in moving into such a role. Participants learn to select the optimum software development life cycle for the project, identify, articulate, and document the customer’s needs, estimate the scope of work and create schedules with the appropriate level of detail, and control common software risks.

Course 0943

Object-Oriented Analysis and Design

Object-oriented design involves transforming the descriptive analysis models into computational models for coding. During an object-oriented analysis, a descriptive model of the problem domain is developed. Instruction uses the notation specified by the Unified Modeling Language (UML). Students learn Agile and Iterative Development methodologies and use case design and requirements driven design. The course covers the principles of object-oriented design as well as practical considerations for applying these principles. Methods for evaluating and fixing poor designs are also addressed, as well as tools and library issues.

Course 0774

Object-Oriented Development: Architectures and Design Patterns, Advanced

This course begins with a discussion of best practices in software architecture and detailed design. Covered next are a wide range of design patterns used to construct modern software systems- patterns which are an integral part of a designer’s engineering toolkit. The course illustrates the reuse of design patterns with an overview of patterns from the gang of four and several domains including refactoring, performance, distribution, lifecycle management, system integration, and message routing. The course concludes with lectures on object-oriented framework design, software product lines, software components, and an overview of aspect-oriented programming.

Course 6633

Perl Programming I

This course introduces users, programmers and system administrators to the popular interpreted language called Perl, the Practical Extraction and Report Language. Perl is hailed as the system administrator’s language and is the de facto standard for writing dynamic Web pages. This practical course is also useful for anyone working with UNIX text files, databases and processes. Although the course is taught on a UNIX system, Perl is very portable. The fundamental topics covered in this course are data types, operators, regular expressions and pattern handling, conditional and looping constructs, file handles and filters, file testing, command-line arguments, subroutines and packages, the UNIX system interface, formatting and database management files.

Course 2856

Professional Software Development Fundamentals

This course provides an essential overview of software product development for people who want to move beyond programming or testing into a lead role or management. The instructor addresses the entire product life cycle, including key make-or-break milestones. The course covers the fundamental practices in software production, along with trade-offs between rigorous and informal approaches. Participants learn measures that help steer progress and determine success. Additional topics include the impact of major industry quality standards, open source opportunities and legal risks.

Course 22868

Programming with Java for Beginners

This course is intended for those who are new to programming or need a refresher. Java, a widely popular programming language, will be used to generate solutions to real, practical problems. The course begins with the concepts of programming, computer science, and software engineering. It introduces the basic Java syntax, and then delves into abstraction, the object-oriented paradigm, procedural programming, elementary data structures, and more. Students will gain a strong foundation and learn to write programs for real applications. The course includes lab exercises.

Course 5185

Python Programming for Beginners

This course is intended for newcomers to programming. The course covers the important concepts and programming mechanisms that exist in all programming languages: reading and writing to standard I/O, using operators, controlling the flow of execution, using functions, reading and writing files, and object-oriented programming concepts. It also includes Python specific facilities such as code re-use, built-in sequence types, and iteration. This is a hands-on lab-based course.

Course 20776

Refactoring with Design Patterns

Software companies prefer to incrementally improve their existing code base instead of undertaking new design and development. This approach to improving the design and implementation of existing code is called refactoring. This course covers refactoring principles, common software problems, reusability, code analysis, and design patterns to solve them. This course is not language specific. Upon successful completion, students will be able to identify, apply, and adapt design patterns to existing designs and implementations, including making trade-offs concerning architectural impacts and using common design patterns to refactor software applications.

Course 6427

Relational Database Essentials

Any application that needs to process and store large amounts of information will likely use a commercial database management system (DBMS). This course provides in-depth knowledge of the concepts behind a DBMS, and then focuses on issues related to practical database design. Students learn how to create conceptual, logical and physical designs of relational databases in response to a set of user requirements. The course will lay a solid foundation for technical professionals and others who want to pursue a career in databases and apply that knowledge to the next level of SQL Server as well as the Oracle series of classes. Topics include relational database concepts, entity-relationship model, normalization, SQL fundamentals, and data warehouse fundamentals.

Course 6195

Ruby, Introduction

Ruby is a dynamically-typed, object-oriented programming language which has recently experienced a surge in interest because the popular Rails Web programming framework is written in Ruby. It can be used for command scripts, system administration, text processing, GUI programs, networked and distributed applications, and Web development. Ruby works well with Test-Driven Development and Agile Methodologies. This course is an introduction to Ruby and will provide a solid foundation for further study. Programming with a dynamic language is different and we'll try to jump-start your learning by emphasizing material not easily found elsewhere.

Course 21341

Software Testing: Techniques, Tools and Practices

This course covers the fundamentals of software testing with an emphasis on test techniques, test tools, and testing practices. For the various phases of software development, the course introduces testing strategies, or test levels. Open source and commercially available tools will be used to demonstrate concepts such as test generation and test coverage. Students gain hands-on testing and analysis experience with sample code using state-of-the-art software analysis and testing tools.

Course 20501

Software Quality Assurance and Testing

As the software industry evolves, the need for qualified engineers trained in the principles, methodologies, techniques, and tools of software quality assurance has grown. This course presents the specifics of software quality assurance and software testing. The course also describes how these processes fit into the software development process. Topics include process and product quality; building an effective SQA organization; techniques and content of an SQA plan; software quality standards; overview of test cycles; test planning; software inspections; basic concepts of measurement; software development, Total Quality Management, and risk management.

Course 3396

Software Security Testing

With software's critical role in today's economy and infrastructure, hackers are constantly attempting to exploit flaws (bugs) in order to steal data and identities, or route spam across the Internet. Today's software needs to be properly tested for security to prevent negative media exposure or other adverse consequences. You will learn application security principles, security testing frameworks, threat modeling, and methods of attacking software applications. This course is intended for software testing professionals or anyone else needing to ensure that applications behave appropriately.

Course 20280

User Experience Design for Web Applications

Web applications are becoming the de facto standard for delivering software functionality to users. This project-oriented course provides an overview of the design process and best practices for Web app development. In addition to the lectures, you'll create and design your own web app using graphical and Web tools to address a real-world business problem. The course explores the key phases of the user experience design process, including strategy, competitive and user research, persona development, UI (user interface) framework and design, usability testing and rapid prototyping.

Course 3113

User Research: User Needs and Usability Assessment for Web and Software Products

It is commonly reported that more than 60 percent of rework in software products results from problems related to not understanding what users need. In this project-based class, students will learn how to make products more usable through a user-centered design (UCD) process. The instructor will provide real-life industry examples of applying UCD methods to software and Web projects. Topics include business scenarios, user profiles and user task analysis; inquiry methods, including contextual inquiry, focus groups, interviews and surveys; and assessment methods, including usability testing, heuristic evaluations, and inspections.

Course 20079

Program Updates and Enrollment Information

Visit ucsc-extension.edu/engineering, for the most up-to-date information about our courses and programs, including textbooks, instructors, schedules and locations.

Enroll online at ucsc-extension.edu.

Send questions to program@ucsc-extension.edu

UCSCextension
Silicon Valley

Silicon Valley's #1 Professional Engineering Educator

Dynamic, Highly Educated Student Body

Nearly 100 percent of the students who enter our programs have Bachelor's degrees—and more than 40 percent already hold graduate degrees. So when studying at UCSC Extension Silicon Valley, you will learn and network with the best and brightest. Working professionals come here to build their careers rather than earn a degree—yet our certificates' quality and recognition have proven instrumental in helping them advance careers.

Wide Variety of Flexible, Competitively Priced Learning Options

We offer highly practical, real-world instruction in dozens of disciplines of high interest and demand in Silicon Valley. Our applied courses provide a theoretical foundation to enhance on-the-job performance. In addition, our programs are very competitively priced—often less than half that of comparable training available elsewhere—with many offered both in classrooms and online. You can take individual courses in any program or earn a full certificate.

Comprehensive Programs for Technology Industries

Our career-oriented education is organized under eleven certificate programs, each representing a significant technology discipline or industry. Course levels vary from basic to advanced. Students can either be beginners in the field, practicing engineers, or job seekers wanting a rigorous program to broaden their skills and enhance their value in the marketplace. Together these programs offer the broadest technology curriculum in Silicon Valley.

UCSC EXTENSION SILICON VALLEY
2505 Augustine Drive, Suite 100
Santa Clara, CA 95054

How To Get Started

Go to ucsc-extension.edu. You'll find detailed information about the full range of certificates and courses we offer, including class schedules and online registration. Or call us at (408) 861-3860 for more information. Our program representatives will personally help you plan a schedule that meets your needs.

Program Contact

Engineering and Technology Department,
(408) 861-3860 or e-mail
program@ucsc-extension.edu.

