Who Should Attend This Program

- Software engineers and developers
- Software project leaders and managers
- Quality assurance and testing professionals
- Aspiring or practicing software architects
- Business professionals interfacing or working with software

Program Summary

The Software Engineering and Quality certificate program at UCSC Extension Silicon Valley is ideal for software developers who want to grow in their areas of expertise and broaden their skills. It blends foundation-level course work in software engineering with cutting-edge developments in software quality, testing, and software interface design. These topics are essential in the software industry but are not commonly covered in traditional computer programming curriculum. Established and aspiring engineers will develop new skills, network with fellow students, and learn from our expert faculty. Software developers, architects, and project leaders will find that this certificate program provides the next level training they need for professional growth.

Join us for a FREE Event to learn more.
Details inside…

ucsc-extension.edu/engineering
Certificate Requirements

Total: 14 units
Required: Choose 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 extensionprogram@ucsc.edu

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

Data Structures and Algorithms Using C++

In this course, you will 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. You will 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 4732

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.

Copyright © 2013 The Regents of the University of California. All rights reserved.
**Software Development Events** (covers Computer Programming, Internet Programming and Development, Software Engineering and Quality, and Linux Programming and Administration) Offered quarterly in two formats:

**Program Overview** | This in-depth FREE presentation is an opportunity to meet program staff and instructors, and learn about this comprehensive curriculum; including requirements, recommended course sequences, and certificate completion timeframes. Get answers to your questions during an informal Q&A.

**Open House** | Which of our 4 certificates fits your career goals? Find out in this FREE Open House presentation. Gain insight from our program staff, learn about new developments in this dynamic field, and get answers to your questions.

*For information and to register, visit uscsextension.edu/events.*

---

### Java Programming for Beginners

This course is an introduction to Java programming, starting with programming concepts and Eclipse IDE. The instructor introduces basic and intermediate Java syntax, and then addresses abstraction, object-oriented paradigm, procedural programming, elementary data structures, and more. Other useful topics include graphics user interface, collections and generics. You will gain a strong conceptual foundation in these areas while starting to write programs for real applications. The course includes programming exercises.

**Course 5185**

### 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. You 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 the 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**

### Machine Learning and Data Mining: Clustering Methods

This survey course is a continuation of “Introduction to Machine Learning and Data Mining” (2612). It covers concepts and principles of unsupervised clustering methods, which divide the data into meaningful groups. The applications of these methods can be found in bioinformatics, marketing research, image processing, information retrieval, and many other fields. You will use the statistical language R to demonstrate clustering algorithms such as k-means, expectation maximization, hierarchical (both agglomerative and divisive), and density-based clustering. The course grade is based on a guided final project.

**Course 30164**

### 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 anyone new to this field or interested in moving into such a role. You will 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**

### Mobile Interface Design

There are many variations of mobile apps in the App Store, but only a limited number of them have innovative design principles, friendly user interfaces, and most importantly, widespread adoption by users. In this hands-on lab and lecture course, you will learn the core principles for creating effective user interfaces for mobile devices incorporating GPS, gestures, augmented reality, tactile experiences, sensors, commerce, tagging and more. You will discover useful patterns for developing mobile products and design a mobile user experience using an iterative and user-centered design process.

**Course 30126**

### 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). You will 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 focuses on the most widely used architectures in enterprises for development and integration such as MVC, EDA, SOA, and Messaging, and covers a wide range of design patterns that are used to construct modern software systems. 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. It concludes with object-oriented framework design, software product lines, software components, and an overview of aspect-oriented programming.

**Course 6633**

### Perl Programming, Comprehensive

This comprehensive hands-on course covers all the basic Perl syntax and programming constructs. You will develop programs in class and as homework assignments. In addition to the basics, the course covers the file handle, filters, testing and system interface. It also includes an introduction to the object-oriented features in Perl, standard libraries, and how to package and modularize Perl programs. By the end of the course, you will be able to develop sophisticated Perl scripts in several applications.

**Course 2110**

*Courses continue on reverse…*
Python Programming for Beginners
This hands-on lab-based 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.

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. The course is not language specific. Upon successful completion, you 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 Design and SQL Programming
This course covers the concepts and design for Relational Database Management Systems (RDMs) and the Structured Query Language (SQL) needed to define and manipulate data. You will learn how to create conceptual, logical and physical designs of relational databases. You will use Oracle 11g to design the entity-relationship diagrams (ERD) and implement a working database. The course covers methods for producing readable output, creating and manipulating tables and creating and managing constraints using SQL. The concepts and SQL language learned here apply to all major RDMS.

Course 30215

Ruby and Ruby on Rails
Ruby on Rails is a popular framework for creating dynamic Web 2.0 database applications. It delivers working, bare-bones Web applications out of the box, ready to be developed into your application. The focus is on the three main topics: Active Record for database persistence, Action Controller for HTTP request routing and processing, and Action View for Web page and form generation. The course also covers configuration, debugging, testing, performance, AJAX on Rails, RESTful architecture, and other advanced topics.

Course 21342

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 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. You will gain hands-on testing and analysis experience with sample code using state-of-the-art software analysis and testing tools.

Course 20501

User Experience Design Fundamentals for the Web
In order to create an effective Web site, you need to address the needs of both the business and the end-users. This course demonstrates how to balance the requirements of both by applying a user-centered design process. The course covers three phases of the process: discovery; definition; and interaction design with an emphasis on the interaction design phase. Topics include user and business requirements; site audits; personae; scenarios and task flows; interaction design; wireframing; prototyping and testing; and information architecture.

Course 30031

Earn Master’s Degree Credit
UCSC Extension and the University of Wisconsin, Platteville, have established an articulation agreement for this certificate program. Students completing the Software Engineering and Quality certificate program can apply their course credits toward an online Master of Science in Engineering degree at UW Platteville upon completion of additional online credits. See details at uwplatt.edu/disted/engineering