Welcome to our 2009–2010 annual report. This year has seen great growth and change in CSS. Undergraduate enrollment was up 30% in Fall 2009. We started the Master of Science in CSS, a career transition degree that has gone from zero to 70 students in only its second year of operation. So, at the start of the 2010–2011 academic year, total CSS enrollment stands at around 300 — as high as it has ever been.

To accommodate this enrollment increase, we’ve been hiring new faculty, with two new Assistant Professors, Hazel Asuncion and David Socha, joining us in fall 2010 and an ongoing search for a full-time Lecturer to start in fall 2011. We created a new Advanced Systems Engineer position and welcomed Josh Larios to help us update our computing infrastructure, starting with a redesign of our Windows laboratory with help from funding from the Student Technology Fee. Among faculty transitions, Carol Zander was promoted to Principal Lecturer — one of only a handful at all of UW — and Frank Cioch has become Professor Emeritus.

We’re also expanding our contribution across the Bothell campus. CSS faculty and staff were intimately involved in development of the broader STEM environment on campus. CSS faculty and staff provided the initial leadership and support for creation of Bothell’s Electrical Engineering degree and start up of our new Science & Technology Program.

We’ve been able to accomplish all of these things despite a difficult economy due to strong campus leadership, creativity in seeking alternative funding models, and the hard work of staff and faculty who have dealt with increased class sizes, more student advisees, and greater demands on our computing infrastructure. We also get by with a little help from our friends and alumni who have given us financial support and devoted their time and energy to helping students with their professional goals, sponsoring internships, and working with faculty to develop exciting new partnerships. It takes all of these extraordinary people to build and grow an extraordinary education and research program.

Mike Stiber
Professor and Director
ABOUT OUR PROGRAM

The CSS Program offers broad and innovative approaches to computer science, software engineering, and the design of applications. Our multidisciplinary curriculum enables students to develop a wide range of competencies, making CSS students particularly valued in today's marketplace.

Enrollment

CSS enrollment continues its strong growth from its dot-com bust low point. We admitted a total of 66 new undergraduate students into the major this fall (Fig. 1). As you can see from the figure, we are holding admissions steady, rather than allowing it to increase further, so that CSS can take on a more prominent role to provide a campus-wide computing curriculum and so we can devote attention to the growth of our graduate program. As the UW Bothell student population and our degree offerings in science and technology evolve over the next few years, CSS is prepared to grow in new ways:

• As can be seen in Fig. 1, the number of “native” UW Bothell students (students who are already at UW Bothell when they apply to one of our majors) entering the major is growing rapidly, now representing one-third of our total undergraduate intake (the remaining two-thirds being students transferring from some other higher education institution). Given continued increases in the size of the UW Bothell freshman class, and their increasing interest in science and technology majors, we anticipate even larger numbers entering our degree programs over the next few years.

• Fig. 2 illustrates that we are now teaching more than at the peak of the dot-com boom (here, “CSS FTE”, the purple curve, means “full-time equivalent students”, aggregates all student credit hours as the equivalent number of students if all were attending school full time). While we only have national data (the “Taulbee survey” on average, to produce one FTE) of how many undergraduate program majors it takes, we offer more courses for non-majors, we will teach more credits even without increasing the number of our own majors.

• Not shown in these graphs is enrollment in our Masters degree, which commenced in fall 2009. In that short time, our graduate enrollment has gone from an initial intake of 27 students to a current enrollment of 70.

We are in the midst of exciting times in CSS as we see a revolution in computing education that moves us beyond our majors into the broad sweep of the UW Bothell educational experience. The CSS faculty are committed to an educational future that responds to the ubiquitous presence of computers in everyone’s life.

Figure 1: New undergraduate majors admitted each fall for last 10 years. The solid line shows total students admitted; dashed indicates what part of that total were UWB students at application time (“native” students). Smaller admissions for winter and spring quarters not included in the figure.

Figure 2: Comparison of CSS full-time equivalent undergraduate enrollment versus the Taulbee survey of undergraduate enrollment at North American computer science departments. For both curves, enrollment in the 1998-99 academic year is normalized to 100.

Figure 3: CSS undergraduate enrollment over time; comparison of headcount (actual number of CSS program majors) with full-time equivalent students (FTE; both curves correspond to left scale). Headcount/FTE (yellow curve) is an indication of how many undergraduate CSS program majors it takes, on average, to produce one FTE (right scale).
Over the past five years, CSS faculty have earned over $400,000 in research grant support. These grants support faculty and student scholarship in areas as diverse as computer science education, signal processing, game development, computing workforce issues, computer vision, Mars lander navigation, distributed computing, and brain modeling.
HAZELINE ASUNCION

Dr. Asuncion’s research emphasis is on traceability and she has developed a novel software traceability approach that automatically links distributed and heterogeneous information. She has investigated the tracing of software license conflicts in heterogeneously composed software systems. Dr. Asuncion is also interested in investigating the traceability challenges in other domains such as e-Science and health care.

FRANK CIOCH

Dr. Cioch’s area of expertise is software engineering. His technical interests derive from his basic interest in software comprehension, both as it relates to the software’s internal characteristics and to its utilization in a particular environment. Dr. Cioch’s specialty is assessing the degree of fit of software requirements, tools and methods to any given situation, and tailoring their application to enhance their effectiveness. He is particularly interested in information sharing, communication difficulties and the type of learning that occurs during software development.

WILLIAM ERDLY

Prof. Erdly’s research entails organizational analysis techniques, workflow management systems, human computer interaction, software risk management, database design.

MUNEHiro FUKUDA

Dr. Fukuda’s research involves self-migrating computations, job coordination, parallel and distributed simulations, shared memory multiprocessor systems, grid computing environments.

CHARLES JACKELS

For many years, Dr. Jackels’ research had focused primarily on application of large-scale computational science methods to chemical and physical problems involving the ground and excited state properties of small molecules, especially those that are of importance in Earth’s atmosphere. Ab initio quantum chemical studies in his lab have focused on the overtone vibrational spectrum of ethanol and the photolysis of C2H2O2. These studies employed large-scale CASSCF configuration interaction, and perturbation theory calculations. Recent activity has moved in an entirely different direction, involving collaboration in an international project to conduct service-based chemistry research for improvement of coffee quality with Nicaraguan small-holder coffee farmers. This project has involved field work on farms in Nicaragua and laboratory studies in both Seattle and Managua. This project is being accomplished through partnership with students, faculty, and staff of Seattle University (SU), the University of Central America Managua, and Catholic Relief Services/Nicaragua (CRS/NI).

MARK KOCHANSKI

Prof. Kochanski is involved in a number of research and development projects. Outside of UWB, his company is involved in the research and development of ‘shrinkwrap’ software and web development with several small software vendors. His research interests are in a variety of areas including software engineering techniques practical for small software teams and ontologies and other semantic structures for encoding, searching, and navigating taxonomies of knowledge.

ALAN LEONG

Prof. Leong’s area of research includes design team process and outcome; voice of the customer as it relates to product design.

CLARK OLSON

Dr. Olson’s research is comprised of computer vision, particularly object recognition, mapping and localization for mobile robots, and indexing in multimedia databases. His is currently working with NASA/JPL on computer vision for Mars rovers.

DAVID SOCHA

Dr. Socha’s interests have consistently been on how to effectively design software and human systems, with the focus on the human and social aspects of software development. He has worked in a variety of software organizations as a programmer, architect, manager, teacher, ScrumbMaster, product designer, change agent, and agile coach.

MICHAEL STIBER

Dr. Stiber’s research pertains to underlying biological neural function for application to machine intelligence: including areas of artificial intelligence, neural networks, bioinformatics, robotics & scientific computing.

KELVIN SUNG

Dr. Sung’s area of research comprises teaching and learning of computer graphics and foundational concepts in programming based on computer games. He enjoys learning about the role technology plays in facilitating human communication of ideas and how to distribute communication workload to a loosely coupled network of computers. He is also interested in Computer Graphics, all aspects of image generation, especially in visibility determination, and algorithms concerning temporal anti-aliasing; Computer Hardware and Architecture; Information Visualization; and Virtual Reality.

CAROL ZANDER

Dr. Zander’s interests include object-oriented programming and design, programming languages, and computer science education. After her original Ph.D. work in Distributed Artificial Intelligence, she was drawn to the education aspect and currently focuses on Computer Science Education research. Her primary research group includes colleagues from the US, Sweden, and Wales.

FACULTY PUBLICATIONS

CLARK OLSON

Robust Registration of Aerial Image Sequences

Game-Theored Instructional Modules: A Video Case Study

Game-Theored CS Education: Empowering the Faculty

Game-Theored Instructional Assignments for Faculty: A Case Study


Computer Games and Traditional Computer Science Courses


CAROL ZANDER

Student Transformations: Are They Computer Scientists Yet?


ALAN LEONG

Medarex-Bristol Meyers Squibb Company: Realizing Its Potential?


MIKE STIBER

Work in Progress — Collaborative Multi-Disciplinary J-DSP Software Project


Game Programming in C#0: A Scaffolded Approach

TEACHING

The number of course sections taught has increased 14% from the 2009–10 to the 2010–11 academic year. This includes an 80% increase in courses at the Masters level.
UNDERGRADUATE
CSS 385: Introduction to Game Development
Examines the fundamental issues in designing and developing computer video games, such as creative and artistic elements, story narration, software architecture, interaction model, mathematical, physics, special effects, and in-game AI logic.

CSS 427: Intro to Embedded Systems
Introduction to the process of specifying and designing embedded systems. Covers the embedded systems development; software and hardware partitioning, processor selection, real-time operating systems, coding in assembly language and C, debugging, and testing. Lab experiments reinforce fundamental concepts using embedded design and debug tools.

CSS 473: Entrepreneurship Seminar
Creation of new venture proposals, presentations for various stakeholders, proof of concept, and issues especially presented for various stakeholders, software architecture, interaction model, mathematical, physics, special effects, and in-game AI logic.

GRADUATE
MASTERS PREP SEQUENCE
CSS 501: Data Structures & Object-Oriented Programming 1
Covers data structures and object-oriented programming. Basic and advanced data types, their uses, and implementations are studied. Students design solutions to programming problems using object-oriented techniques with various data types. Algorithms and their tradeoffs are examined in-depth, while continuously using and applying software engineering practices.

CSS 502: Data Structures & Object-Oriented Programming 2
Covers advanced data structures including trees, balanced trees, heaps, graphs, and hash tables along with associated algorithms. Covers object-oriented programming with a focus on design and implementation of problems using inheritance and polymorphism, while also introducing formal automata theory.

CSS 503: Systems Programming
Examines the logical design and programming aspects of operating systems and network communication. Topics covered include processes, threads, synchronization, deadlocks, memory management, virtual memory, file systems and client-server network programming.

CSS 504: Software Development Processes
Provides a foundation in software engineering processes, methods, and practices associated with prescriptive and agile software process models. Students create artifacts commonly used to communicate, justify, and manage computing projects.

CSS 505: Software Modeling Techniques
Provides the concepts and skills needed to use modeling in software analysis and design to foster understanding and communication of a problem and its potential solution(s). Students will create modeling artifacts for projects by hand and using CASE tools.

CORE COURSES
CSS 543: Advanced Programming Methodology
Builds on knowledge of data structures and operating systems, introducing thread based and component based multi-tier programming. Reviews synchronization mechanisms and design/implementation of concurrent applications, discusses language/system independent software reuse, component technology, and multi-tier application design and development.

CSS 555: Evaluating Software Design
Studies best software engineering practices and methods used in prescriptive and agile approached to create and evaluate software design from an quality principles point-of-view. Considers design from quality dimensions such as performance, scalability, maintainability, usability, and security.

CSS 565: Research Methods in Software Development
In-depth study of research design and data analysis techniques for computing-related research activities. Students prepare a research proposal; examine experimental, quasi-experimental, and qualitative design strategies; perform meta-analytic research, define and collect appropriate software metrics; and perform appropriate advanced statistical analyses.

NEW COURSES
CSS 385: Introduction to Game Development
Examines the fundamental issues in designing and developing computer video games, such as creative and artistic elements, story narration, software architecture, interaction model, mathematical, physics, special effects, and in-game AI logic.

CSS 427: Intro to Embedded Systems
Introduction to the process of specifying and designing embedded systems. Covers the embedded systems development; software and hardware partitioning, processor selection, real-time operating systems, coding in assembly language and C, debugging, and testing. Lab experiments reinforce fundamental concepts using embedded design and debug tools.

CSS 473: Entrepreneurship Seminar
Creation of new venture proposals, presentations for various stakeholders, proof of concept, and issues especially related to technology ventures.

CSS 501: Data Structures & Object-Oriented Programming 1
Covers data structures and object-oriented programming. Basic and advanced data types, their uses, and implementations are studied. Students design solutions to programming problems using object-oriented techniques with various data types. Algorithms and their tradeoffs are examined in-depth, while continuously using and applying software engineering practices.

CSS 502: Data Structures & Object-Oriented Programming 2
Covers advanced data structures including trees, balanced trees, heaps, graphs, and hash tables along with associated algorithms. Covers object-oriented programming with a focus on design and implementation of problems using inheritance and polymorphism, while also introducing formal automata theory.

CSS 503: Systems Programming
Examines the logical design and programming aspects of operating systems and network communication. Topics covered include processes, threads, synchronization, deadlocks, memory management, virtual memory, file systems and client-server network programming.

CSS 504: Software Development Processes
Provides a foundation in software engineering processes, methods, and practices associated with prescriptive and agile software process models. Students create artifacts commonly used to communicate, justify, and manage computing projects.

CSS 505: Software Modeling Techniques
Provides the concepts and skills needed to use modeling in software analysis and design to foster understanding and communication of a problem and its potential solution(s). Students will create modeling artifacts for projects by hand and using CASE tools.

CSS 508: Software Testing and Quality
Reviews approaches, concepts and techniques used to validate and verify software and methods used to improve software processes. Students will reflect on the applicability of software engineering and computer science methods as part of a capstone activity.

CORE COURSES
CSS 543: Advanced Programming Methodology
Builds on knowledge of data structures and operating systems, introducing thread based and component based multi-tier programming. Reviews synchronization mechanisms and design/implementation of concurrent applications, discusses language/system independent software reuse, component technology, and multi-tier application design and development.

CSS 555: Evaluating Software Design
Studies best software engineering practices and methods used in prescriptive and agile approached to create and evaluate software design from an quality principles point-of-view. Considers design from quality dimensions such as performance, scalability, maintainability, usability, and security.

CSS 565: Research Methods in Software Development
In-depth study of research design and data analysis techniques for computing-related research activities. Students prepare a research proposal; examine experimental, quasi-experimental, and qualitative design strategies; perform meta-analytic research, define and collect appropriate software metrics; and perform appropriate advanced statistical analyses.

CSS 595: Capstone Project I
First of the two-quarter capstone project course sequence for CSS graduate students.

CSS 596: Capstone Project II
Second of the two-quarter capstone project course sequence for CSS graduate students.

CSS 700: Master’s Thesis
Graduate research to prepare for and complete the requirements for a Master’s Thesis.

ELECTIVE COURSES
CSS 534: Parallel Programming in Grid and Cloud
Exploration of theoretical programming methodology and practical middleware design used for parallel programming in grid and cloud systems. Use of different programming models, parallelizing patterns, and middleware systems for designing application-specific fault-tolerant parallel software.

CSS 552: Topics in Rendering
Studies core algorithms and technologies in synthesizing high quality images, including; camera models, 3D viewing, visibility sampling and approximation, light source models, material property approximation, illumination models, human vision system, and texture synthesis.

CSS 577: Secure Software Development
Augmenting standard software engineering practices with practices to develop applications with low security risks. Covers security risk analysis and assessment, design practices, STRIDE, threat modeling, secure coding practices, fuzz and penetration testing, security response, and security-analysis tools.

CSS 583: Knowledge Management Systems
Explores contemporary theoretical and practical implications of how to create and manage knowledge as acquired using technology. Uses different strategies such as XML, RDF, RDFS, and other approaches to provide methods and structures to organize and reference data for use within a variety of knowledge domains.

CSS 587: Advanced Topics in Computer Vision
Content varies. May include image and video databases, object recognition, video processing, scene reconstruction, and robot vision. Students will implement projects on current topics in computer vision research.

CSS 600: Independent Study or Research
Independent study or research on computing topics conducted under the direction of one or more instructors.
SPECIAL TOPICS

FALL 2009

CSS 390: Programming Languages (Theory and Practice)
Taught different languages/paradigms and compiler/translation theory, such as particular parsing and code generation.

CSS 390: Cryptography (History & Theory)
Investigated the history of codes, the political and economic ends to which they had been put, and the mathematical theory underlying them. Specific mathematical content included prime numbers, modular arithmetic, and elementary number theory. The highlight of the course was the study of the RSA cryptosystem.

CSS 490: Agile Software Development
Examined the agile versus plan-driven controversy between software development. Students were given the opportunity to do in-depth study of project management, requirements engineering, design, programming, or testing for agile methods.

Class was run in an agile manner rather than planned in which the instructor gave a set of introductory lectures and then student interest determined what materials were to be covered in subsequent lectures.

For grades, students were required to give lectures and develop a portfolio.

CSS 490: Product Development
Covered technology project and product development within the dynamic of time-pressured competition. Discussed how to systematically improve products to beat competition and win the customer. Topics include benchmarking, competitive intelligence, and managing small group product development.

CSS 490: Interactive Game Design
Focused on how computer scientists can best design, build, and implement computer games.

It began with an in-depth review of drivers of game design, transition into game design methodologies/concepts, understanding state-of-the-art game analysis/texting approaches and examined future technologies and design methods.

Also discussed the fundamental issue of why we play games and the real goals of these games. The instructor provided options for students to do sophisticated coding or research.

WINTER 2010

CSS 490: Entrepreneurs Workshop
Provided a simulated experience at building your own business. The final deliverables were rough business plan, proof of concept, and highly polished presentations.

SPRING 2010

CSS 490: Social Computing
Examined how the social, cultural, and cognitive sciences explain/influence next-generation multi-user computing application design and their impact.

Students examined specialized applications including work-flow management systems, simulations, multi-player gaming, education, large database/search tools, instant message, blogs, and other highly interactive applications.

CSS 490: Game Development
Examined the fundamental issues in designing and developing computer video games. The first half of the class was programming intensive where students practiced and implemented the technical knowledge learned by developing games individually. After the mid-term exam, students worked in groups designing and implementing their own games.
CSS students engaged in internships with more than 35 area companies and community organizations. Our Speaker Series events provide continuing education and a link to the Computer Science industry to students and the community. The series sponsors guests from a wide range of areas allowing students to explore, experience, and network.
Owning the Problem in Wireless Research

Since the termination of analog TV transmission in 2009, the bandwidth used to transmit the analog signal is now empty. This unused spectrum is known as white space. The reuse of this white space to provide wireless broadband Internet connectivity has been met with both support and opposition.

Dr. Victor Bahl, a Principal Researcher and founding Manager of the Networking Research Group in Microsoft Research Redmond, discussed the evolution of our thinking on this issue with a societal, policy, technical, and business approach.

Ubiquitous Networking

Ubiquitous networks are expected to be a promising technology for life-supporting medical applications and disaster mitigation. From a networking point of view, ad hoc and sensor networks are an important piece of the ubiquitous network. Dr. Takashi Watanabe, a professor in the Faculty of Informatics from Shizuoka University in Japan, discussed the topic of ubiquitous networking. He talked about smart antenna-based medium controls and routing and well as data aggregations to reduce network traffic and routing protocols for sustainable sensor networks.

Dr. Watanabe received his M.E. and D.E. degrees from Osaka University in 1984 and 1987 respectively. In 1987, he joined Microsoft Research, Tokyo, Japan University as an assistant professor. In 1990, he moved to Faculty of Engineering, Shizuoka University. He was a visiting researcher at University of California, Irvine from 1995 through 1996. He has served on many program committees for networking conferences, IEEE, ACM, IPSJ (Information Processing Societey of Japan), IEICE (The Institute of Electronics, Information and Communication Engineers, Japan). His research interests include mobile networking, ad hoc networks, sensor networks, ubiquitous networks, intelligent transport systems, specially MAC and routing. He is a member of IEEE, IEEE Communications Society, IEEE Computer Society, ACM SIGMOBILE as well as IPSJ and IEICE.

State of Computing Summit

 Advisors and faculty members from nearby community colleges met at the second annual State of Computing Summit hosted by the CSS Program. Attendees discussed their approaches to zero-level programming courses taught at their campuses. Zero-level programming courses are beginner-level courses in which the fundamental concepts and ideas of programming are introduced to students.

In addition, attendees brainstormed ways to interest a more diverse group of students in computer science.

Two approaches to teaching and learning were discussed for introductory programming courses: experiential and unplugged. In experiential learning, instructors teach students about the fundamental concepts of programming by having them take part in it, such as by actively solving problems, understanding program logic, elements, and syntax, and creating basic programs. The unplugged approach adds a step prior to experiential, in which students are first introduced to the concept by lecture or discussion and then implement what they learned.

The aim of this summit was to both share ideas and concepts amongst colleges and colleagues and to also broaden the scope of teaching by incorporating different and new ideas. In doing so, community colleges, along with the CSS Program, will be able to provide a better learning and educational experience to current and future students.

Women in Computing Dinner

It is no secret that in an industry dominated by men, it can be a difficult choice to pursue a career in computing for women. The CSS Program is proud to have a strong cadre of female students working towards their undergraduate and graduate degree, and recently spent an evening acknowledging the challenges they face.

In winter quarter, the department reinitiated and hosted the ‘Women in Computing’ dinner to bring awareness and strengthen the bond between female students and alumni within the Program and throughout the professional computing community.

This women-only informal dinner created a no-stress environment in which attendees had an opportunity to get to know one another and network with current students and alumni. In addition, it gave them the opportunity to share their ideas, experiences, and issues as women within the computing world.

Due to the large turnout and overall success of this dinner, plans to have another event are underway for spring quarter. Future event announcements will be sent to current and alumni students via the CSS email list.
Adaptis Inc.
Clear One Health Plans - ID Cards and Fulfillment Materials

Advanced Acoustic Concepts
Mapping: Software, Symbology, and Defect Detection

Areva T&D
Walkthrough for Exposing Legacy Systems

Avanade
Avanade IT Intern - Improvement of Laptop Deployment Process

Boeing
Boeing IT Information Security

C. Alex Leigh, Specialized Applications Group
Specialized Applications Group, Master Data Management Platform

Cascadia Community College
Resume Management Tool

Comcast
Conversion from Microsoft SQL 2000 to Microsoft SQL 2005

Cranhart Creative
Contacts Database Web Application

Crystal Commerce
Capturing Market Metrics

CSS Faculty Research
BrainGrid neural network simulator

Evaluating the efficient utilization of parallel hardware resources in a multi-core GPU.

Game Across Multiple Environments

Datality
Datality Development Tools in Eclipse

Internships & Sponsors (CSS 497 Projects)

Electronic Evidence Discovery
Rock Guitar Learning Game

Elektrobit, Inc
User Interface Development for Automotive Radio Head Units

ESAL Kirkland Lab Cogswell Polytechnical College
User interface and Back End for Fog of War Implementation Inside Torque 3D

Engineering Simulation and Animation Lab Internship:Fog of War

Global Crossing
Field Operations Inventory System

Griptonite Games
Console Video Game Development

Art Tools Development

Hewlett Packard
Hewlett Packard web-applications internship

HomeBound Services, Inc.
Website Development & Design

Individual Project
Draco, Episode 1: A legend begins

Physics-based Games and Gameplay

InfoSpace
Automated UI Software Regression Tester

Institute for Systems Biology
ChromEval: A Software Tool for the Evaluation of Separation Columns Using Single Ion Chromatograms

International Sustainability Institute
AlleyArt Project and International Sustainability Institute Database Implementation website upgrade

Ken Craig Software Development Group
Software Update in Tracking Family Demographics

Measurement Technology Northwest
ThermDAC Reporting Tools

Microsoft
Debugging Tools for the Internet Explorer Team

Anti Virus Submission System Dashboard

Implementing Features & Tests for Microsoft SQL Server’s Machine Health Tool

MySpace, Inc.
MySpace QA/test Tools Development

Northpoint Escrow & Title
Analysis and extension of existing and proposed proprietary business applications

Pacific Northwest Ballet
End-to-end Software Development Project: Pacific Northwest Ballet iPhone Application

Applicaton for iPhone

Software Development Project for Local Non-Profit Pacific Northwest Ballet iPhone application

Pandamonium Games
Facebook Game Development

Predictive Solutions
Media Analysis Client Setup

Quantivate LLC.
The Technology Recovery Planning Module (TRPM)

RealNetworks
Logo Overlay Pre-Filter Development

Software Release Engineer Intern

Renewable Energy Technology Group
Agile Development in Startup Company

Home Based Renewable Energy Systems

Renewable Energy Project Perspectives

Siemens Medical Solutions
Working in the Field, a Lesson in Software Project Development

Tulaip Tribe of Washington
Website enhancements for Tulaip Data Services Group

University of Washington Foster Business School
Foster Business School Undergrad Admission Systems

UW Autism Center
Eye tracking device driven valve-game Engine applications for studying autism

UW Bothell Career Services
Resume Management Tool

UW Bothell Science & Technology
Dossier 2.0

Cryptography Spy Dossier 2.0

Vertafore
Software Developer for an Insurance Management System

AMS360 Jupiter release

Zetec, Inc
Internship at Zetec, Inc

Zombie Inc
Video Game from Alpha to Release