Software Development Program Curriculum

Associate in Science Degree – Software Development

Whatcom COMMUNITY COLLEGE

Table of Contents

Program Overview

The software development program is designed to prepare individuals to work as entry-level programming positions within an organization. Students will acquire the core skills to design, code, implement, and maintain programs and database systems that provide programming solutions that meet industry needs. The two year Associate in Science (AS) degree in Software Development includes courses in operating systems, networking, programming, mobile devices, cloud computing and embedded systems. Successful completion of this degree prepares students to enter directly the workforce as junior software developers, testers or programmers.

The degree has been designed with opportunity to transfer to select Washington State community college Bachelors of Applied Science degrees in Software Development in mind. Whatcom Community College intends to also develop a Bachelors of Applied Science in Software Development, to provide seamless educational pathway for students to bolster the local software development workforce.

Program Outcomes

Program outcomes are overarching skills that are emphasized and reinforced throughout several courses in a specific program. They are measurable statements that define the skills the college expects its students to develop by the end of a certificate or degree at WCC.

As a result of completing the software development degree, graduates should be able to...

1. Write and test applications that meet specifications for industry standard platforms.
2. Apply data management concepts to databases.
3. Design solutions utilizing software tools.
4. Apply steps of the software development lifecycle.
5. Analyze systems to develop recommendations for projects.
6. Communicate professionally with clients and coworkers.
7. Conduct software development research.

Work-based Learning

This degree will integrate paid work-based learning opportunities to students in the form of a final quarter academic internship. Students can take this internship course(s) that can range from five to ten credits, resulting in 165 to 330 hours on the job site. These internships will include direct supervision, with meaningful projects that results in high-impact learning. Internships help students develop skills and qualities for success in the workplace, such as work ethic, critical thinking, and problem solving. Specific and measurable learning outcomes are developed by the student, faculty mentor, and employer within an internship learning contract.

These second year software development internship positions can range from testing and bug-fixes, to researching and developing new projects based on employer needs. By entering into a second year internship the software developer student will have been trained in the following topics.

- Proficient in Java, with experience in Javascript, and C programming languages

- Skilled in basic database queries
- Familiar with Agile / Scrum design methodologies
- Familiar with Networks networks and operating systems
- Able to effectively communicate with supervisor and coworkers
- Able to work independently, with strong attention to detail

Academic interns will require both an on-site supervisor and will have a faculty mentor from the college that oversee academic assignments linked to the course.

Industry Alignment

As a professional/technical degree, this program has been developed based upon industry needs through a formal industry input campaign and curriculum development process. Local industry leaders will continue to provide oversight of the curriculum through a professional/technical program advisory committee. Industry members will also be invited to participate in multiple methods within the program. Industry partner participation will be elicited in the following activities:

- Leadership in program advisory committee membership
- Consistent curriculum evaluation and review
- In-class guest speaker and lecture opportunities
- Hosting tour of job site (in-person or virtual)
- Hosting students to job-shadow
- Participate in information interviews with prospective students
- Host second year students in paid internships
- Conduct mock interviews with students
- Recruitment events or marketing for internships and jobs
- Participation in on-campus and virtual career fairs

These activities are intended to have mutually beneficial outcomes for both the program, students, and industry partners. Faculty will engage with industry to ensure students are exposed early on to employer standards and expectations. Students will benefit from opportunities to network and increase professionalism skills, as well as gain work-experience within their program. Employers will be able to promote career opportunities, while screening and recruiting from high caliber students.

Open Access

Whatcom Community College is an open access institution that allows students to enter and pursue degrees, without undue financial or artificial barriers. The software development program will not include a selective entry application, in order to remove access barriers and promote the diversity of students pursuing this career pathway.

Degree Template

The associate in science (AS) degree offers practical training to prepare students to enter the work force. Associate degree requirements are overseen by the State Board for Community and Technical Colleges and Whatcom Community College is accredited by the Northwest Commission on Colleges and Universities. Professional technical degrees must meet a minimum of 90 credits and include at least fifteen credits of General Education Requirements (GERs). These GER classes must have one course focusing on communication (CM), one on computation (CP), and oral communication (OC).

Associate in Science Degree- Software Development

Course #	Course title	Credits
GENERAL EDUCATION REQUIREMENTS/RELATED INSTRUCTION		
ENGL&101	English Composition I (CM)	5
MATH& 141	Precalculus I (CP) (5 credits) or any higher level course in calc sequence	5
CMST	Any CMST or CMST& course designated "OC" 5 credits	5
Subtotal		**15**
CORE REQUIREMENTS		
CS& 141	Computer Science I Java	5
CIS 105	Computer Operating Systems I	5
CS 120	HTML Fundamentals	5
CS 145	Computer Science II	5
CIS xxx*	Introduction to Database Design	5
CIS 236	Cisco Networking I	5
CS 240	Data Structure and Algorithm Fundamentals	5
CIS xxx*	Cloud Foundations	5
SD 130	Systems Analysis and Design	5
SD 230	Web Programming	5
SD 235	Programming for Mobile Devices	5
SD 247	Applied Computer Architecture	5
SD 180	Technical Internship and Career Preparation	2
SD 250	Software Security Principles	3
Subtotal		**65**
ELECTIVES/COOPERATIVE WORK EXPERIENCE		
SD 299 **and/or** SD 290	Software Development Capstone Software Development Internship	5 5-10
Subtotal		**10**
Total		**90**

*Course numbers will be selected by Computer Information Systems program at a future date.

Quarterly Schedule

Whatcom Community College follows the eleven week quarter schedule. The software development degree will offer a fall quarter entry point and initially offer a summer break between years one and two. In future years as the program expands, there will be multiple entry points within the year for students to begin course work and the program will also include summer offerings.

Quarter 1	**Course Name**	**Credits**
SD 130	Systems Analysis and Design	5
CS &141	Computer Science I Java	5
ENGL&101	English Composition I (CM) 5 credits	5

Quarter 2	**Course Name**	**Credits**
CIS 105	Computer Operating Systems I	5
CS 145	Computer Science II	5
MATH& 141	Precalculus I (CP) (5 credits) or any higher level course in calc sequence	5

Quarter 3	**Course Name**	**Credits**
CS 240	Data Structure and Algorithm Fundamentals	5
CS 120	HTML Fundamentals	5
CIS xxx*	Introduction to Database Design	5

Quarter 4	**Course Name**	**Credits**
CIS 236	Cisco Networking I	5
SD 230	Web Programming	5
SD 180	Technical Internship and Career Preparation	2
SD 250	Software Security Principles	3

Quarter 5	**Course Name**	**Credits**
CMST OC	Any CMST or CMST& course designated "OC"	5
SD 247	Applied Computer Architecture	5
SD 235	Programming for Mobile Devices	5

Quarter 6	**Course Name**	**Credits**
CIS xxx*	Cloud Foundations	5
SD 299	Software Development Capstone	5
	and/or	
SD 290	Software Development Internship	5-10
		Total Credits: 90

*Course numbers will be selected by Computer Information Systems program at a future date.

Course Descriptions

Credit Hour Overview

Each lecture credit equates to one hour of class time either face-to-face or online per week. Each lab credit equates to two hours of time spent in the lab classroom, receiving hands-on instruction, learning activities, or projects. One work site credit equates to three hours in the field at an employer site within a structured internship experience.

General Education Requirements (GER)

The fifteen credits of related instruction/general education requirements in professional-technical degrees and certificates are automatically satisfied for students who have completed a direct transfer agreement (DTA) associate degree from an accredited institution within the state of Washington or a baccalaureate degree from an accredited institution within the United States. Other degrees will be considered on a case-by-case basis.

ENGL& 101 – English composition I
Prerequisite: ENGL 095 or placement into ENGL& 101.
Credits: 5.00 (5.00 lecture, 0 lab, 0 work site)
Hours: 55 (55.00 lecture, lab, work site)

Course description
This course helps students become more effective writers in academic and professional settings. Students learn to enter ongoing academic conversations, analyze and use secondary sources to formulate, develop, revise, and communicate ideas in writing, and shape their message to different purposes, audiences, and media.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Use rhetorical knowledge to compose texts.
2. Think critically about texts (print, media, data, etc.).
3. Use multiple composing processes to conceptualize, develop and finalize writing projects.
4. Analyze the formal rules and informal guidelines that define genres.
5. Evaluate information needs including source location, credibility, and relevance.
6. Explain how their understanding of composing has developed from prior experience and will continue to develop.

Course content
Students produce several essays across multiple drafts based on analyses of various texts, including print and visual media; they study the conventions of academic writing, including but not limited to incorporating the ideas of others into their own work, synthesizing source material, and joining an on-going conversation; they study and reflect upon the rhetorical effectiveness of their own work, including the presentation and development of ideas, as a means of making choices that govern revision.

MATH& 141 – Precalculus I

Prerequisite: MATH 099 with a minimum grade of C
Credits: 5.00 (5.00 lecture, 0 lab, 0 work site)
Hours: 55 (55.00 lecture, lab, work site)

Course description

The basic properties and graphs of functions and inverses of functions, operations on functions, compositions; various specific functions and their properties including polynomial, absolute value, rational, exponential and logarithmic functions; applications of various functions; conics. A graphing calculator is required.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Analyze the graphs of polynomial, rational, exponential, logarithmic, and piecewise functions.
2. Analyze relationships between real and complex zeros, linear factors, and x-intercepts of a polynomial function.
3. Solve polynomial, exponential, and logarithmic equations.
4. Perform function composition.
5. Analyze the relationships between graphs of conic sections and their standard equations.

Course content

Complex #'s and operations

Polynomials

- Graphs of Polynomials (End behavior, turns, zeros)
- Finding zeros of Polynomials (real and complex)
- Theorems: Fundamental Theorem of Algebra, Complex Zeros Theorem

Rational Graphs

- Vertical Asymptotes, Horizontal Asymptotes, Holes, Zeros
- Oblique Asymptotes

Piecewise Functions

- Graphing Piecewise Functions
- Absolute Value as a Piecewise Function

Function Composition

- Algebra of Compositions
- Graphing Compositions
- Inverse functions

Exponentials and Logs

- Graphs of Exponentials and Logs
- Properties of Logs
- Change of base
- Solve Exponential and Logarithmic Equations
- Applications/Models of Exponentials and Logs

Conics

- Graphs in Standard Form
- Change and graph non-standard form
- Properties of Conics – Center, Vertices, Directrix, Foci

Core Requirements

CS&141 – Computer Science I Java

Recommended Preparation: Computer Science 101
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, 0 work site)

Course description
This course provides an introduction to computer programming with Java. It covers computer architecture, machine instruction processing, basic data types, program control structures, methods, classes, and fundamental data structures. Recommended for math, science, engineering, computer science and software development majors.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Develop and implement algorithms in an object-oriented programming language.
2. Identify elements of a programming language, including data types, operators, variables, expressions, basic input and output, and arrays.
3. Describe object-oriented terminology and concepts.
4. Demonstrate programming concepts by designing, coding, testing and debugging programs.

Course content

- Data types and data type representation
- Digital logic structures
- Instruction processing
- Program control structures
- Subroutine structures
- Fundamental data structures
- Basic file input and output
- Debugger operations
- Introductory method analysis

CIS 105 Computer operating systems I

Recommended preparation: CIS 100
Credits: 5.00 (3.00 lecture, 2.00 lab, 0 work site)
Hours: 77 (33.00 lecture, 44.00 lab, 0 work site)

Course description
This course introduces the fundamentals of computer operating systems, including history, evolution, and design, as well as support, maintenance, and troubleshooting. Lab work included.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Describe the boot process and the basic interactions between hardware and software.
2. Install and update an operating system.
3. Perform the daily tasks involved in managing and troubleshooting operating systems.
4. Use the command line to perform routine tasks.
5. Identify and interpret system events logged on a system.
6. Describe and demonstrate file security and file sharing methods.

7. Backup, modify, and recover critical files.
8. Carry out the recovery of a failed system.

Course content

- Relationships between computer software and hardware
- Operating system installation and configuration
- Operating system updates and routine system maintenance
- Performance and event monitoring
- Malware identification and removal
- Implementation of local system security policies
- Workgroup networking and secure file sharing configuration
- System recovery

CS 120 – HTML fundamentals

Recommended preparation: Windows file management and keyboarding skills
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, 0 work site)

Course description

Teaches the fundamentals of web page design and implementation. Emphasizes text formatting, web page layout, links, lists, tables, frames and forms using HTML, scripting, and database connectivity.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Compare and contrast the origins, evolution, protocols, and architecture of the Internet and World Wide Web.
2. Develop a multi-page website using the current HTML standard.
3. Incorporate cascading style sheets (CSS) to format and style a website.
4. Incorporate tables, forms and form input elements into a web page.
5. Develop websites that include audio and video multimedia elements.
6. Use fundamental JavaScript capabilities to add interactivity and functionality to web pages.
7. Discuss various topics related to Internet search engines (e.g., web bots--spiders and crawlers, search engine databases, meta tags, keyword and description tags, search engine results page and search engine optimization concepts).

Course content

- Introduction to the Internet and WWW
- Web page HTML tag usage
- Web page text formatting
- Web page graphics
- Web page layout design using CSS
- Web page links to other web pages and web sites
- Table usage in web pages
- Frame usage in web pages
- Form usage in web pages
- Introduction to JavaScript
- Web page promotion and Internet search engine concepts

CS 145 – Computer Science II

Prerequisite: CS&141
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, work site)

Course description

This course is a continuation of CS&141. Teaches the fundamentals of computer programming. Covers searching and sorting, object oriented design, error handling, file input and output, event based programming, bitwise operators, multithreaded and network programming. Recommended for math, science, engineering, computer science and software development majors.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Explain the differences between procedural and object-oriented programming.
2. Define inheritance and polymorphism.
3. Define and develop multiple classes in one program.
4. Explain the difference between recursion and repetition.
5. Implement data structures including arrays, sets, lists, collections, and trees.
6. Implement concurrent programming and linear data structures.

Course content

- Class and data type
- Encapsulation
- Class inheritance
- Polymorphism
- Advance file input and output
- Recursive calls
- Dynamic memory allocation
- Advanced data structures
- Intermediate method analysis

CIS XXX* Introduction to Database Design

Hours: 66 (44.00 lecture, 22.00 lab, 0 work site)
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)

Course description

Students will learn how to use Structured Query Language (SQL) to retrieve and organize information from a relational database, filter, modify, group and summarize data, and retrieve joint information from multiple tables in a database.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Demonstrate understanding of the purpose, design, and terminology of a relational database.
2. Explain and use relational schema.
3. Use a relational database management system to enter, edit, and run SQL statements.
4. Access multiple tables and work with unions, subqueries, self joins, inner joins, and outer joins.
5. Use aggregation functions.
6. Build SQL queries to retrieve, store, and modify data.
7. Create and edit tables and enforce data integrity on them.

8. Evaluate and synthesize information in a database.
9. Gather and organize information needed for database creation and maintenance.

Course content

- Introduction to Databases
- The Relational Model
- SQL: Data Manipulation
- SQL: Data Definition
- Entity Relationship Modeling
- Normalization
- Security
- Query Processing
- Web Technology and DBMSs

*Course number will be selected by Computer Information Systems program at a future date.

CIS 236 – Cisco networking I

Prerequisite: CIS 105 with a minimum grade of C or permission of program coordinator
Credits: 5.00 (3.00 lecture, 2.00 lab, 0 work site)
Hours: 77 (33.00 lecture, 44.00 lab, 0 work site)

Course description

First in the three quarter networking sequence. This course introduces the fundamentals of networking, including introduction to the OSI and TCP/IP network models, and IP addressing and sub-netting. Other topics include: network design, topologies, protocols, wiring, network devices, and network security fundamentals.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Explain the layers of communications in data networks using network protocol models.
2. Configure and verify network device interfaces.
3. Configure subnet masks and addresses.
4. Build a simple Ethernet network using routers and switches.
5. Employ basic cabling and network designs to connect devices.
6. Perform basic router and switch configuration and verification using Cisco CLI commands.

Course content

- Fundamentals of networking including introduction to the OSI and TCP/IP network models
- IP addressing and subnetting
- Network design
- Topologies and protocols
- Network Security Fundamentals

CS 240 – Data Structure and Algorithm Fundamentals

Prerequisite: CS 145
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, work site)

Course description

Teaches software development skills that emphasize the study of abstract data types using object oriented programming techniques, Big O algorithm analysis, fundamental data structures such as lists, stacks, queues, and trees, and searching and sorting.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Analyze the runtime performance of code segments and operations using big-O notation.
2. Implement recursive and non-recursive algorithms to manipulate (construct, insert, delete, search, and traverse) binary search trees.
3. Design and use hash tables appropriately in solving problems.
4. Implement recursive and non-recursive algorithms to manipulate (construct, insert, delete, search, and traverse) heaps.
5. Use heaps appropriately in solving problems.
6. Compare and contrast sorting algorithms.

Course content

- Object-oriented programming and design techniques
- Fundamental searching and sorting algorithms
- Fundamental data structures
- Big-O efficiency analysis

SD 130 Systems Analysis & Design

Prerequisite: None
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, 0 work site)

Course description

This course examines the system-development cycle in depth. Topics include, problem identification, problem solving, and information-gathering techniques. Current structured tools are used to describe business rules and objects, data flow, data structures, and process flow and documentation. Creative problem solving and working in a team environment are stressed.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Explain the process of systems analysis, design, development and implementation.
2. Discuss the components within the Systems Development Life Cycle.
3. Explore tools needed to complete an effective systems analysis and design process.
4. Create types of Unified Modeling Language.
5. Analyze feasibility considerations, cost-benefit techniques, and candidate systems.
6. Describe components of application architecture, input/output design, and user-interface design.
7. Explain various Project Management methodologies.

Course content

- Problem identification
- Problem solving
- Information-gathering techniques
- Project management methodologies
- Unified Modeling Language
- Object Oriented Analysis & Design
- Software Development Life Cycle

SD 247 Applied Computer Architecture

Prerequisite: CIS 105 and CS 145
Credits: 5.00 (3.00 lecture, 2.00 lab, 0 work site)
Hours: 77 (33.00 lecture, 44.00 lab, 0 work site)

Course description

An overview of basic computer architecture where concepts are applied to course labs and projects. Topics include introduction to assembly, introduction to C programming language, data representation, and memory organization and management.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Write and debug simple programs using assembly language and C programming language.
2. Explain how computers represent data.
3. Describe how computers organize and manage memory.
4. Apply knowledge of computer memory organization and management to write more efficient programs.
5. Describe major functional components of a computer.

Course content

- Introduction to assembly language
- Introduction to C programming language
- Data representation
- Memory organization and management
- Introduction to working in a Linux environment
- Major functional components of a computer

SD 230 Web Programming

Prerequisite: CS 120 and CS 145
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, 0 work site)

Course description

Utilizing various scripting languages, students will learn to create interactive and dynamic web pages and applications. Topics include client and server side scripting, basic web security, and writing code on the web that interacts with a database.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Develop interactive and dynamic web pages and applications.
2. Test and debug code for web programming.
3. Explain the basics of how the internet works.
4. Implement basic web security practices for web pages and applications.

Course content

- Web scripting languages
- Client and server side scripting
- Basic web security
- Interacting with databases in web programming
- Interacting with networks

CIS XXX* – Cloud Foundations

Prerequisite: CIS 105 with a minimum grade of C or permission of program coordinator
Credits: 5.00 (3.00 lecture, 2.00 lab, 0 work site)
Hours: 77 (33.00 lecture, 44.00 lab, 0 work site)

Course description

This course introduces foundational cloud computing concepts and best practices. Students will examine the benefits and cost of operating cloud architecture and discuss the financial impact of cloud migration and discuss the general impact of migrating to a cloud-based architecture.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Compare core cloud services and their business application.
2. Assess and evaluate cloud computing.
3. Describe the general impacts of migrating to a cloud-based architecture.
4. Evaluate cloud security.
5. Describe the financial impact of cloud migration.

Course content

- Evaluate cloud platforms
- Government compliance concerns
- Access Control
- Storage Options
- Database Types
- Severless Computing

*Course number will be selected by Computer Information Systems program at a future date.

SD 235 Programming for Mobile Devices

Prerequisite: CIS 105 and CS 145
Credits: 5.00 (4.00 lecture, 1.00 lab, 0 work site)
Hours: 66 (44.00 lecture, 22.00 lab, 0 work site)

Course description
This course teaches the principles of mobile application design and development. Students will learn application development for major mobile platform(s). Topics will include user interface design, memory management, user interface building, input methods, data handling, and network techniques.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Develop applications for deployment on mobile devices.
2. Describe differences between mobile and desktop applications.
3. Design user interfaces for touch oriented input model.
4. Apply development tools to implement mobile applications.
5. Use Software Development Kits (SDK) for mobile applications.
6. Describe security and performance requirements for mobile applications.
7. Control hardware features of a device.

Course content

- Software Development Kits (SDKs)
- Evaluate cloud platforms
- Mobile Application & Device Platforms
- Mobile App Dev. Life Cycle
- Mobile App: Front End
- Mobile App: Back End
- Native Mobile Apps
- Cross Platform Apps
- Hybrid Web Apps
- Progressive Web Apps
- Mobile Security
- Virtualization platforms
- Legal/Privacy Issues

SD 180 Technical Internship and Career Preparation

Prerequisite: CS 145
Credits: 2.00 (2.00 lecture, 0.00 lab, 0 work site)
Hours: 22 (22.00 lecture, 0 lab, 0 work site)

Course description
This course is designed for software development students to prepare for obtaining an internship or job, while fostering career readiness skills. They will highlight their own strengths and skills within a professional application, including resume, cover letter, and technical interview for an internship or job. Common strategies for a successful interview will be discussed while practicing solving interview problems.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Assess personal qualifications, strengths, and skills relating to job description.
2. Research occupational field and employing community.
3. Compose electronic application materials.
4. Demonstrate problem solving under the conditions of a job interview.
5. Respond to a variety of technical and verbal interview questions.

Course content

- Values, strengths, and transferrable skills assessments.
- Employment outlook, career related skills, and job search resources.
- Resume, cover letter, online job materials, professional communication, references, and application process.
- Technical job interviewing problems and problem solving skills.
- Interview questions and quality responses.
- Networking
- Portfolio development and maintenance

SD 250 Software Security Principles

Prerequisite: CS 240 & CIS XXX Introduction to Database Design
Credits: 3.00 (2.00 lecture, 1.00 lab, 0 work site)
Hours: 44 (22.00 lecture, 22.00 lab, 0 work site)

Course description

Students will explore fundamentals of software security, and learn how to write more secure code. Topics include common software and website vulnerabilities, proactive coding practices, and basics of cryptography.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Explain the basics of cryptography.
2. Implement security measures in code to create secure software.
3. Describe common software vulnerabilities.
4. Apply proactive coding practices to software projects.

Course content

- Common software vulnerabilities
- Website vulnerabilities
- Proactive coding practices
- Basics of cryptography

Electives/Cooperative Work Experience

SD 299 Software Development Capstone

Prerequisite: SD 247 or permission of instructor or program coordinator
Credits: 5.00 (2.00 lecture, 3.00 lab, 0 work site)
Hours: 88 (22.00 lecture, 66.00 lab, 0 work site)

Course description
Students will work in teams to develop a real world software application, applying concepts taught in the Software Development program. In developing their application, students will complete the entire application development lifecycle, including analysis, design, specification, implementation, testing, debugging, and deployment.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. Apply the software application development lifecycle to a project.
2. Apply software testing methodologies to test an application.
3. Produce written documentation for an application.
4. Develop a software application.
5. Demonstrate professional communication.

Course content

- Software application development lifecycle
- Software testing methodologies
- Project proposal
- Project implementation
- Presentation/demonstration of project

SD 290 Software Development Internship

Prerequisite: SD 180 and permission of instructor or program coordinator
Credits: 1-10.00 (0 lecture, 0.00 lab, 1-10 work site)
Hours: 33-330 (0 lecture, 0 lab, 33-330 work site)

Course description
The student will complete an academic internship that offers experience-based learning in a structured, supervised, and career-related setting. Measurable internship learning outcomes are collaboratively created by the student, internship supervisor, and faculty mentor. The student will develop professional readiness by fostering skills necessary to work in the software development field. Repeatable with program permission.

Course outcomes: Upon successful completion of this course, each student should be able to...

1. [Create, execute, and assess learning outcomes that align with the software development program.]
2. Communicate with stakeholders in a professional manner.
3. Use software development concepts in a professional environment.
4. Demonstrate professionalism in the workplace.

Course content

In order to guide the internship experience, students will create, execute, and be assessed on specific, measurable, meaningful, and attainable internship learning outcomes, which will be documented in the learning contract.

In order to demonstrate professionalism, students will practice and be assessed on critical thinking and problem solving, oral and written communication, teamwork and collaboration, leadership, work ethic, and cultural respect.