Computer Science

PROGRAM OBJECTIVES

Students in the Computer Science program prepare for a future in software development and the use of computer technology to solve complex problems. An initial core of classes introduces students to general principles of programming and software development.

Students pursuing the computer science degree will learn to design and develop software systems for industrial, scientific, and commercial applications. They will acquire an understanding of computer operating systems, programming, data structures and algorithms, and systems analysis. Graduates will be prepared to work in the private or public sectors as programmers, analysts, or software engineers, or to proceed to advanced study.

LEARNING OUTCOMES
All program graduates will demonstrate achievement in the following areas:

- *Content Knowledge:* demonstrate factual and conceptual grasp of the field of computing.
- *Integrated Learning and Communication:* demonstrate the ability to incorporate learned skills to design, develop, and evaluate software systems of varying complexity to meet desired user requirements.
- *Problem Solving:* demonstrate proficiency in using one or more industry-standard programming languages and scripting languages to solve problems;
- *Inquiry, Critical Thinking, and Analysis:* demonstrate ability to apply conceptual knowledge for analysis and problem solving.
- *Teamwork and Civic Engagement:* demonstrate teamwork ability to work collaboratively with end users and other developers.

MEANS OF ASSESSMENT
The outcomes for each class will be clearly stated on the syllabus. Assessments for courses will address both the conceptual and applied aspects of the class. Means of assessment include projects, quizzes and exams. The objectives for projects and other assigned work tie directly into course outcomes.

In addition to course-level assessment, the program provides for assessment of the students' abilities to integrate concepts from the entire spectrum of coursework. Each student is required to develop a capstone project prior to graduation. The precise nature of the capstones varies according to specific student interests, but generally include the complete design documents for a software product and the finished product itself.

REQUIREMENTS FOR THE BACHELOR OF SCIENCE & BACHELOR OF ARTS IN COMPUTER SCIENCE
(Foreign language proficiency is a requirement for the B.A. degree.)
In addition to General Education requirements, B.S. and B.A. degree candidates for Computer Science should complete a total of 76 credit hours. A grade of "C-" or better is required for each course. A minimum overall GPA of 2.00 for all courses is required for completion of the degree.

COMPUTER SCIENCE CORE:
(44 Credits). (For B.A. and B.S.)
CS 121 Introduction to Software Development (1)
CS 161 Foundations of CS I (4)
CS 162 Foundations of CS II (4)
CS 221 C/C++ Programming (4)
CS 260 Data Structures (4) **UWR*
CS 311 Operating Sys (3)
CS 315 Software Design (4) **UWR*
CS 318 Algorithm Analysis (4)
CS 330 Database Management Systems (4)
CS 360 Object-Oriented Programming with C++ (4)
CS 380 Software Engineering (3)
CS 401 Capstone (3) **UWR*
CS 407 Seminar (2)

COMPUTER SCIENCE ELECTIVES
(choose at least 16 credits from the list below)
COM 252 Intro Web Authoring (3)
CS 316 Authoring Environment Programming (4)
CS 321 Computing Theory (3)
CS 325 Applied 3-D Graphics/Animation (4)
CS 328 Intro Video Game Design/Dev (4)
CS 335 Networking & Network Administration (4)
CS 369 Mobile App Development (4)
CS 409 Practicum (1-12)
CS 425 Computer Graphics (4)
CS 440 Artificial Intelligence (4)
CS 310 Special Topics (1-5)
CS 410 Special Topics (1-5)
RELATED AREA REQUIREMENTS (16 credits)
MATH 231 Discrete Math (4)
MATH 251 Calculus I (4)
MATH 252 Calculus II (4)
MATH 341 Linear Algebra (4)

MINOR IN COMPUTER SCIENCE
The Computer Science minor is intended for students who seek a basic understanding of software engineering and systems analysis. It includes the required introductory courses in the foundations of CS and data structures, operating systems, user interface design and the theory of object oriented programming. Students also have the opportunity to select one or two electives in upper division CS courses of interest.

- A minimum of 32 graded credits as follows:
  CS 161 Fndtns of Computer Science I (4)
  CS 162 Fndtns of Computer Science II (4)
  CS 221 C/C++ Programming (4)
  CS 260 Data Structures (4)
  CS 360 Object-Oriented Programming (4)
  Upper Division CS Electives (12)
- A minimum grade of "C-"required for each course with a cumulative average GPA of 2.00 or more for all courses required for the minor.
- A minimum of 10 hours required the minor must be completed at Eastern Oregon University.

CS CERTIFICATES
The CS Certificate program consists of two certificates. The first, Certificate in Computer Programming I, is a prerequisite for the second. The certificates are designed to provide entry-level skills and knowledge in practical applications of computer science.

REQUIREMENTS

- Prior to beginning the Certificate in Computer Programming I, students must take the Programming Aptitude Test and the Math Accuplacer. Students who do not place into MATH 111 or above must complete MATH 111 before beginning the certificate coursework.
- Students must complete all courses with a "C or better" to earn credit for certificates.
- Students must complete the Certificate in Computer Programming I before starting on any of the other three certificates.

CERTIFICATE IN COMPUTER PROGRAMMING I
The Certificate in Computer Programming I provides basic knowledge and skills in computer programming. Completion of this certificate is minimal preparation for entry-level programming positions.

Required Courses:
CS 161 Foundations of Computer Science I (4)
CS 162 Foundations of Computer Science II (4)
CS 260 Data Structures (4)

Total Credits: 12

CERTIFICATE IN COMPUTER PROGRAMMING II
The Certificate in Computer Programming II builds on the basic skills provided by the first certificate and provides a solid skill set for entry-level programming work.
Required Courses:
CS 221 C/C++ Programming (4)
CS 344 System Analysis & Design (3)
CS 360 Object-Oriented Programming (4)
CS 380 Software Engineering (3)

TYPICAL FIRST YEAR CURRICULUM
Fall
CS 121 Intro Software Development (1)
CS 161 Foundations of CS I (4)

Winter
CS 162 Foundations of CS II (4)

Spring
CS 260 Data Structures (4)

TYPICAL SECOND YEAR CURRICULUM
Fall
CS 221 C/C++ Programming (4)

Winter
CS 315 Interactive Software Design (4)
CS 318 Algorithm Analysis (4)

Spring
CS 360 Object-Orient Prog w C++ (4)

TYPICAL THIRD YEAR CURRICULUM
Fall
CS 311 Operating Systems (3)
Elective

Winter
CS 330 Database Management System (4)
Elective

Spring
CS 380 Software Engineering (3)
Elective

TYPICAL FOURTH YEAR CURRICULUM
Fall
CS 401 Capstone (3)
CS 407 Seminar (2)
Elective

Winter
Elective

Spring
Elective

TRANSFER WITH AAOT/CS

TYPICAL FIRST YEAR CURRICULUM
Fall
CS 221 C/C++ Programming (4)
Elective

Winter
CS 315 Interactive Software Design (4)
CS 318 Algorithm Analysis (4)

Spring
CS 360 Object-Orient Prog. C/C++ (4)
Elective

TYPICAL SECOND YEAR CURRICULUM
Fall
CS 311 Operating Systems (3)
CS 401 Capstone (3)
Elective

Winter
CS 330 Database Management System (4)
Elective

Spring
CS 380 Software Engineering (3)
Elective

COMPUTER SCIENCE COURSE DESCRIPTIONS

CS 110 - Selected Topics (Credits: 1 to 6)
An in-depth presentation of a topic of interest to both students and faculty. Topics will vary from year to year depending on the interests and availability of faculty. Prerequisites: May be required for some topics.

CS 121 - Intro Software Development (Credits: 1)
This survey course introduces computer software, the process of its development, and its uses in contemporary society. Topics include data representation, basic computer architecture, and categories of software including multimedia products, end-user applications, process-control, and scientific computing.

CS 140 - Microcomputer Syst (Credits: 3)
Emphasis is placed on the technical details of the microcomputer system as a whole in order to produce sophisticated users. This course focuses on microcomputer operating systems, their structures and relations to the microcomputer architecture, a technical understanding of information flow through the microcomputer and its hardware interfaces. This course also introduces the Internet, networking, and communications protocols such as TCP/IP.

CS 161 - Foundations of CS I (Credits: 4)
Introduces basic data representation, branching and iteration, memory management, computer architecture, and the analysis and design of problem solutions.

CS 162 - Foundations of CS II (Credits: 4)
Introduces some common algorithms for searching and sorting, the analysis of algorithm complexity, exception handling, and file output. Prerequisites: MATH 111, CS 161.

CS 209 - Field Placement (Credits: 1 to 15

CS 210 - Selected Topics (Credits: 1 to 6)
An in-depth presentation of a topic of interest to both students and faculty. Topics will vary from year to year depending on the interests and availability of faculty. Prerequisites: May be required for some topics

CS 221 - C/C++ Programming (Credits: 4)
An introduction to the basics of programming as used in C and C++, including selection statements, loops, arrays, string handling, pointers, registers and functions. Practical exercises will require the construction, compilation, debugging, and execution of complete programs that implement given algorithms to solve simple problems. The emphasis in this course will be on the common features of C and C++; however memory allocation and the use of pointers will be discussed. Prerequisite: CS 162

CS 260 - Data Structures (Credits: 4)
Institutional Graduation Requirement - UWR
An introduction to various implementations of commonly used data structures and their applications. Topics include lists, stacks, queues, trees and heaps. Prerequisite: CS 162.

CS 310 - Selected Topics (Credits: 1 to 5)
An in-depth presentation of a topic of interest to both students and faculty. Topics will vary from year to year depending on the interests and availability of faculty. Prerequisite: May be required for some topics. Student must have at least sophomore standing to register for this course.

CS 311 - Operating Systems (Credits: 3)
The principles and problems involved in the development of a computer operating system. Overview of the development of operating systems, sequential and con-current processes, cooperation, communication and mutual exclusion, synchronization constructs: monitors, conditional critical regions, semaphores; deadlocks, resource allocation, scheduling policies, storage management. Prerequisite: CS 221 and CS 260. Student must have at least sophomore standing to register for this course.

CS 315 - Interactive Software Design (Credits: 4)
Institutional Graduation Requirement - UWR
Covers models of software development, with emphasis on the prototyping model, and user interface design. Students will design an interactive product, producing deliverables for each stage of design up to the development of a working prototype. Prerequisites: CS 162: Foundations of Computer Science II.

CS 316 - Authoring Environment Programming (Credits: 4)
Students learn to apply procedural and object-oriented programming methodologies to create interactive products for informational, educational, and entertainment applications for web or stand-alone delivery Prerequisites: CS 162: Foundations of Computer Science II.

CS 318 - Algorithm Analysis (Credits: 4)
The analysis of a variety of algorithms that arise frequently in computer applications. Basic principles and techniques for analyzing and improving algorithms in areas such as list searches, sorting, pattern recognition, polynomial and matrix computations. Prerequisite: CS 260. Student must have at least sophomore standing to register for this course.

CS 321 - Computing Theory (Credits: 4)
Includes automata, complexity, Turing machines, and unsolvable problems. Prerequisite: CS 260. Student must have at least sophomore standing to register for this course.

CS 325 - Applied 3-D Graphics and Animation (Credits: 4)
Covers the three dimensional computer modeling tools for the creation of still and moving images. Topics include creation of models using a variety of techniques including spline and vertex editing; animation using keyframes, skeleton rigging and morph targets; virtual lighting, and texture maps. Prerequisites: CS 162: Foundations of Computer Science II. Junior standing or consent of instructor.

CS 328 - Intro to Video Game Design & Development (Credits: 4)
The design, implementation, and testing of video games. Includes incremental game engine development, simple graphics, user input, animation, sound, music, and artificial intelligence. Prerequisites: CS 221, CS 260.

CS 330 - Database Mgmt System (Credits: 4)
Analysis, design, and implementation of data systems in relation to information transfer. Prerequisite: CS 260. Student must have at least junior standing to register for this course.

CS 335 - Networking/Network Admin (Credits: 4)
An introductory examination of the Open System Interconnection Reference Model (OSI). Topics covered include network architecture, data flow control, transmission control, path control, recovery, and routing techniques. Prerequisite: CS 162. Student must have at least sophomore standing to register for this course.

CS 360 - Object-Orient Prog With C++ (Credits: 4)
A study of object oriented programming with C++. Beginning and intermediate concepts are covered including classes, objects, member functions, overloading, inheritance, polymorphism, templates, and virtual functions. Prerequisite: CS 221, 260. Student must have at least sophomore standing to register for this course.

CS 369 - Mobile Application Development (Credits: 4)
Development of applications for phones, tablets, and other mobile devices, with an emphasis on the constraints facing mobile application design and development from both a hardware and user perspective. Introduction to current mobile app frameworks, events, and user interfaces. Prerequisites: CS 360.

CS 380 - Software Engineering (Credits: 3)
Emphasis is on the specification, organization, implementation, testing, and documentation of software. Inherent problems, challenges, tools, and methods of a large software project. Presents methods and tools used in the various stages of software production. This course should prepare students for the problems they will encounter as software professionals. Prerequisite: CS 260, CS 315. Student must have at least sophomore standing to register for this course.

CS 381 - Programming Languages (Credits: 4)
Concepts of high-level programming languages. Syntax and semantics of several existing languages. Compilers, interpreters and formal syntax specification. Prerequisite: CS 360. Student must have at least sophomore standing to register for this course.

CS 401 - Capstone (Credits: 1 to 6)
Institutional Graduation Requirement - UWR
Prerequisite: Upper-division standing or consent of instructor. Student must have at least junior standing to register for this course.

CS 407 - Seminar (Credits: 1 to 6)
Student must have at least junior standing to register for this course.

CS 409 - Practicum (Credits: 1 to 12)
Students gain practical experience in a professional or pre-professional setting. Prerequisites: Upper-division standing and consent of instructor. Student must have at least junior standing to register for this course.

CS 410 - Selected Topics (Credits: 1 to 5)
An in-depth presentation of a topic of interest to both students and faculty. Topics will vary from year to year depending on the interests and availability of faculty. Prerequisites: May be required for some topics. Student must have at least junior standing to register for this course.

CS 425 - Computer Graphics (Credits: 4)
This course studies the principles underlying the generation and display of 3-D computer graphics. Topics include geometric transformations, 3-D viewing and projection, lighting and shading, color, camera models and interaction, and standard graphics APIs. Prerequisites: CS 221 and junior standing.

CS 440 - Artificial Intelligence (Credits: 4)
Basic concepts of intelligent systems and artificial intelligence programming, representation, control, communication, and perception. Prerequisites: CS 318, CS 360, and junior standing