Computer science is the study of information systems, their representation, architecture, and implementation, used for a variety of practical and theoretical purposes. Computer science addresses methods by which data is accessed, stored, and retrieved, which include areas such as representational computation, programming languages, algorithmic modeling, and software design, testing and development. Computer scientists apply their knowledge of mathematics, physics, and logic to solve a variety of problems using diverse technology.

Students learn practical methods of reasoning, problem-solving, and theoretical analysis to develop their skills in computer science. While exploring general courses in programming, systems analysis, mathematics, and physics, students apply their skills to core challenges within the field. PCC offers students the opportunity to earn an Associate of Arts Oregon Transfer (AAOT) degree, or Associate of Science (AS) degree. Students may also complete courses as preparation for a bachelor’s or advanced degree or update skills to industry standards. Students wishing to transfer credits must check the specific requirements of the college/university to which they intend to transfer, and should seek Academic Advising for help in tailoring a transfer degree to accommodate the requirements of their chosen major. Computer Science transfer guides exist for Portland State University, Oregon State University, and other Oregon institutions offering Computer Science degrees.

**CS 133G. Introduction to Computer Games. 4 Credits.**
Introduces fundamentals of computer game development, including a survey of computer game categories and platforms, major game components, the game development process, and game graphics. Design and development of elementary two-dimensional computer games. Prerequisites: (WR 115 and RD 115) or IRW 115 and MTH 20 or equivalent placement. Audit available. This course fulfills the following GE requirements: Science, Math, Computer Science/AAS, Science, Math, Computer Science/AGS, Science, Math, Computer Science/AAOT.

**CS 133U. C Programming. 4 Credits.**
Introduces computer programming through development of C programs to solve practical problems. Recommended: CS 160. Audit available.

**CS 140U. Introduction to UNIX. 4 Credits.**
Introduces the UNIX/Linux operating system, including: task scheduling and management, memory management, input/output processing, internal and external commands, shell configuration, and shell customization. Explores the use of operating system utilities such as text editors, electronic mail, file management, scripting, and C/C++ compilers. Discusses trends in UNIX/Linux, including use of graphical user interfaces. Recommended: CS 160. Audit available.

**CS 160. Exploring Computer Science. 4 Credits.**
Explores the field of computer science. Provides an overview of computer architecture, software development engineering, data organization, problem-solving strategies, ethics, and theory of computation. Explores career options and develops rudimentary software development skills. Recommended: Computer Literacy (such as completion of CS 120); MTH 65 and RD 115. Audit available. This course fulfills the following GE requirements: Science, Math, Computer Science/AS, Science, Math, Computer Science/AAOT, Science, Math, Computer Science/AAS, Science, Math, Computer Science/AGS.

**CS 161. Computer Science I. 4 Credits.**
Introduces the concepts of computer science. Explores problem solving, algorithm and program design, data types, loops, control structures, subprograms, and arrays. Introduces writing programs in a high level programming language. Surveys current social and ethical aspects of computer science. Recommended: MTH 111, WR 121, and CS 160. Audit available. This course fulfills the following GE requirements: Science, Math, Computer Science/AAS, Science, Math, Computer Science/AGS, Science, Math, Computer Science/AAOT, Science, Math, Computer Science/AS.

**CS 162. Computer Science II. 4 Credits.**
Explores classes, pointers, dynamic memory, linear linked lists, multi-dimensional arrays, program correctness, verification, and testing. Recommended: MTH 112 and WR 121. Prerequisites: CS 140U and CS 161. Audit available. This course fulfills the following GE requirements: Science, Math, Computer Science/AAS, Science, Math, Computer Science/AGS, Science, Math, Computer Science/AAOT, Science, Math, Computer Science/AS.

**CS 201. Computer Systems. 4 Credits.**
Explores computer systems from a software perspective. Provides an overview of C and assembly language programming and reading skills. Explores basic systems programming skills and tools to measure and improve program performance based on an understanding of key aspects of machine architecture. Prerequisite: CS 162 and CS 140U. Audit available.

**CS 233G. Game Programming. 4 Credits.**
Introduces object-oriented architectures and software design patterns used for game design. Explores a game engine software framework to design and implement several kinds of games, animation techniques, physics simulation, user controls, graphical methods, and intelligent behaviors. Recommended: one term of a programming language such as C, C++, Java or C#. Audit available.

**CS 250. Discrete Structures I. 4 Credits.**
Introduces discrete structures and computational techniques in the areas of first-order logic, discrete proofs, number theory, sequences, induction, recursion, and set theory. Prerequisite: MTH 251 and CS 162. Audit available.

**CS 251. Discrete Structures II. 4 Credits.**
Introduces discrete structures and computational techniques in the areas of functions, relations, probability, graph theory, algorithm analysis, and finite state automata. Prerequisite: CS 250. Audit available.

**CS 260. Data Structures. 4 Credits.**
Explores stacks, queues, lists, vectors, hash tables, graphs, trees and algorithms including sorting, searching, iterating over data structures and recursion. Prerequisite: CS 162. Audit available.

**CS 261. Programming Systems. 4 Credits.**
Explores the theory and practice of object-oriented programming as embodied in both Java and C++. Introduces inheritance, polymorphism, virtual functions, templates, exceptions, operator overloading and the extensive libraries that are available as a standard part of Java and C++. Prerequisite: CS 260. Audit available.