Computer Science

Chair: John Stratton
Janet Davis, (on Sabbatical, 2019-2020)
Andrew Exley

Students of computer science gain insight into a technology on which we increasingly rely, while learning new ways of thinking and tools to solve problems in many domains. Central to computer science is the concept of an algorithm—a precise, repeatable procedure for solving a well-defined problem. Computer scientists discover, define and characterize computational problems; they design, implement, and evaluate algorithmic solutions. Studying computer science in the context of a liberal arts education enables graduates to approach problems from multiple perspectives and communicate effectively with diverse colleagues and stakeholders.

Computer Science 167 is suitable for both potential majors and non-majors who have no prior computer science experience. Students with prior experience should discuss their placement with a computer science faculty member.

Distribution: Some courses completed in Computer Science apply to the quantitative analysis distribution area. See General Studies Program for lists of courses that apply.

Total credits required to complete a Computer Science major: A student who enters Whitman College with no prior experience in computer science will need to complete 33 credits.

Learning Goals: Upon graduation, a student majoring in Computer Science will be able to:

- Understand and apply fundamental algorithms and data structures;
- Understand the abstractions supporting modern software systems, and how the construction of those mechanisms affects the supported systems;
- Apply mathematical techniques to justify computational solutions and explore the limitations of computers;
- Communicate computational ideas through speech, writing, diagrams, and programs;
- Work with a team to design and implement a substantial, integrative project;
- Propose and compare multiple solutions to computational challenges, with consideration for the context and impact of each solution on the creators, maintainers, and users of that solution;

The Computer Science major: Twenty-nine credits in Computer Science at the 200 level or higher, including Computer Science 310, 320, 327, 370, 495, and 496.

A student will typically take Computer Science 167, 210, 220, and 270 as prerequisites to the explicitly required courses, along with three additional elective credits at the 200 level or higher, for a total of 33 credits. Students granted advanced placement may proceed directly to coursework at the 200 level. Students with a 4 or 5 on the APcomputer science (A) test are considered to have completed the equivalent of Computer Science 167 and receive four credits in computer science.

No more than 10 credits earned in domestic or foreign study programs, transfer credits, and/or AP or IB credits may be used to satisfy the course and credit requirements for the major.

Students without advanced placement in mathematics and statistics will also take Mathematics 125 (3 credits) as a prerequisite to Computer Science 220.

Computer science majors will typically fulfill the quantitative analysis distribution requirement through program prerequisites. Students considering graduate study are encouraged to take additional courses in Mathematics and Statistics, particularly Mathematics 240 (Linear Algebra) or 247 (Statistics with Applications).

Senior assessment: The senior assessment consists of the capstone project (Computer Science 495 and 496), a one-hour oral examination, and satisfactory performance on the written Major Field Test. The oral examination will address topics from Data Structures and the 300-level Computer Science core; the student’s capstone project will be considered as context for some questions.

Honors in the major: Students do not apply for admission to candidacy for honors. Students will be invited to honors candidacy based on achievement of the minimum Cumulative and Major GPAs specified in the faculty code

(3.300 and 3.500, respectively), and strong performance in Computer Science 495. To earn honors in the major, a candidate must additionally achieve distinction on the senior assessment and earn an A or A- in Computer Science 495, 496, and 498. An acceptable digital copy of the Honors Project Report must be submitted to Penrose Library no later than Reading Day; source code and/or data will be deposited electronically.

The Computer Science minor: A minimum of 15 credits in courses numbered 200 or above.
Note: Courses taken P-D-F prior to the declaration of a computer science major or minor will satisfy course and credit requirements. Courses taken P-D-F and independent studies may not be used to satisfy course and credit requirements for the major or minor.

The Data Science minor: A minimum of 19 credits from: Computer Science 167, Mathematics 240, 247. In addition to these three required courses, at least 9 credits from any of Computer Science 351, 357 or Mathematics 248, 339, 347, 349, 350. Students wishing to combine the Data Science minor with the Mathematics major will be allowed to use Mathematics 240 to satisfy both sets of requirements. Students, in this case, will complete a minimum of 51 credits total.

167 Introduction to Computational Problem Solving
Fall, Spring Fall: Loveland, Stratton; Spring: Loveland 4 credits

Students will learn to design, document, implement, test, and debug algorithmic solutions to computational problems in a high-level, object-oriented programming language. We introduce core concepts: algorithms, data structures, and abstraction. We apply foundational constructs common to all programming languages: data types, variables, conditional execution, iteration, and subroutines. Students will gain experience with exploratory and structured approaches to problem solving through collaborative in-class exercises. Frequent programming projects will address applications of computing to problems arising from other disciplines.

200-204 Special Topics in Introductory Computer Science
1-4 credits

A course which examines special topics in computer science at the introductory level. *Prerequisite:* Computer Science 167. Any current offerings follow.

200 ST: Machine Learning with Deep Neural Networks
Spring Loveland 3 credits

This course will focus on understanding machine learning algorithms and effective application to a variety of problem classes, ranging from classification and dimensionality reduction to the development of agents for decision-making in dynamic environments. Topics will include supervised methods for classification and regression as well as unsupervised/semi-supervised learning for analyzing unlabeled data. Underlying models will be based primarily on artificial neural networks using deep learning methods for training. Necessary math beyond the prerequisites will be covered in class. The software base layer will be the Python programming language. A broad range of currently relevant application areas will be discussed (e.g. super human game playing, augmented medical diagnoses, self-driving cars). *Prerequisite:* Computer Science 167 and Mathematics 125 (or equivalent). Distribution area: None.

210 Computer Systems Fundamentals
Spring Exley 3 credits

This course integrates key ideas from digital logic, computer architecture, compilers, and operating systems, in one unified framework. This will be done constructively, by building a general-purpose computer system from ground up: from the low-level details of switching circuits to the high level abstractions of modern programming languages. In the process, we will explore software engineering and algorithmic techniques used in the design of modern hardware and software systems. We will discuss fundamental trade-offs and future trends. *Prerequisite:* Computer Science 167.

220 Discrete Mathematics & Functional Programming
Fall Loveland 3 credits

Students will practice formal reasoning over discrete structures through two parallel modes: mathematical proofs and computer programs. We will introduce sets and lists, Boolean logic, and proof techniques. We will explore recursive algorithms and data types alongside mathematical and structural induction. We consider relations and functions as mathematical objects built on set theory and develop idioms of higher-order programming. May be elected as Mathematics 220. *Prerequisite:* Computer Science 167 and Mathematics 125.

267 Human-Computer Interaction
Not offered 2019-20 **4 credits**
How do people interact with computers? And how can we design computer systems that make people's lives better? Students will learn to critique user interfaces using principles based on psychological theories of perception, memory, attention, planning, and learning. Through a semester-long team project, students will practice iterative design including stages of contextual inquiry, task analysis, ideation, prototyping, and evaluation. We will also explore current research on new application areas, design techniques, or interaction paradigms, as well as social implications of computing.

270 Data Structures
Fall, Spring **Fall: Stratton; Spring: Exley** **4 credits**
This course addresses the representation, storage, access, and manipulation of data. We discuss appropriate choices of data structures for diverse problem contexts. We consider abstract data types such as stacks, queues, maps, and graphs, as well as implementations using files, arrays, linked lists, tree structures, heaps, and hash tables. We analyze and implement methods of updating, sorting, and searching for data in these structures. We develop object-oriented programming concepts such as inheritance, polymorphism, and encapsulation. We consider implementation issues including dynamic memory management, as well as tools for programming in the large. *Prerequisite:* Computer Science 167.

300-304 Special Topics in Computer Science
1-4 credits
A course which examines special topics in computer science at the intermediate level. Any current offerings follow.

310 Computer Systems Programming
Fall **Exley** **4 credits**
How does data move from a hard drive to memory to a CPU? How does a computer deal with input from a mouse and keyboard? How does one computer communicate with another, or many others? This class examines how operating systems interact with computer hardware to provide higher-level programming abstractions. Students will use the C programming language to explore topics such as processes, virtual memory, concurrency, threads, and networking. *Prerequisites:* Computer Science 210 and 270.

317 Software Performance Optimization
Not offered 2019-20 **3 credits**
Computers do not execute programs with equal speed, even when theoretical analyses indicate that two programs perform approximately the same amount of work. At the same time, software power efficiency affects the size of mobile devices and the energy consumption of datacenters. This course examines current trends in computer system architecture and draws out insights for developing software that is fast and energy-efficient. Students will work problem sets, write programs, conduct experiments, read and analyze technical articles, and carry out a team project of their choice. Throughout the course, we shall consider how computer system designs affect program structure, and in particular the tensions between efficiency and principled software organization. *Prerequisites:* Computer Science 210 and 270.

320 Theory of Computation
Fall **Exley** **3 credits**
Which problems can be solved computationally? Which cannot? Why? We can prove that computers can perform certain computations and not others. This course will investigate which ones, and why. Topics will include formal models of computation such as finite state automata, push-down automata, and Turing machines, as well as formal

languages such as context-free grammars and regular expressions. May be elected as Mathematics 320. *Prerequisite:* Computer Science/Mathematics 220 or Mathematics 260.

327 Algorithm Design & Analysis

Spring **Stratton** **3 credits**

How can we be confident that an algorithm is correct before we implement it? How can we compare the efficiency of different algorithms? We present rigorous techniques for design and analysis of efficient algorithms. We consider problems such as sorting, searching, graph algorithms, and string processing. Students will learn design techniques such as linear programming, dynamic programming, and the greedy method, as well as asymptotic, worst-case, average-case and amortized runtime analyses. Data structures will be further developed and analyzed. We consider the limits of what can be efficiently computed. May be elected as Mathematics 327. *Prerequisites:* Computer Science 270; Computer Science/Mathematics 220 or Mathematics 260.

339 Operations Research

Spring **Hundley** **3 credits**

Operations research is a scientific approach to determining how best to operate a system, usually under conditions requiring the allocation of scarce resources. This course will consider deterministic models, including those in linear programming (optimization) and related subfields of operations research. May be elected as Mathematics 339. *Prerequisites:* Mathematics 240 and Computer Science 167.

350 Mathematical Modeling and Numerical Methods

Not offered 2019-20 **3 credits**

This course explores the process of building, analyzing and interpreting mathematical descriptions of physical processes. This may include theoretical models using statistics and differential equations, simulation modeling, and empirical modeling (meaning model building from data). The course will involve some computer programming, so previous programming experience is helpful. May be elected as Mathematics 350. *Prerequisites:* Mathematics 240 and 244.

351 Artificial Intelligence

Not offered 2019-20 **3 credits**

How can a computer defeat a human at chess or go? Can a computer really learn new information? This course will focus on algorithms used to make a computer exhibit some level of what humans call "intelligence". Topics include tree search, graph search, neural networks, decision trees, logical inference, and Bayesian probability models. For the final project, students will select a technique to apply to a classification problem or game of their choice. *Recommended prerequisites*: Computer Science 220, Mathematics 220, or Mathematics 260. *Prerequisite:* Computer Science 270.

357 Natural Language Processing

Fall **Exley** **3 credits**

Computers are poor conversationalists, despite decades of attempts to change that fact. This course will provide an overview of the computational techniques developed in the attempt to enable computers to interpret and respond appropriately to ideas expressed using natural languages (such as English or French) as opposed to formal languages (such as Python or C++). Topics in this course will include signal analysis, parsing, semantic analysis, machine translation, dialogue systems, and statistical methods in speech recognition. *Prerequisite*: Computer Science 270.

370 Software Design

Spring **Stratton** **4 credits**

What makes code beautiful? We consider how to design programs that are understandable, maintainable, extensible, and robust. Through examination of moderately large programs, we will study concepts including object-oriented design principles, code quality metrics, and design patterns. Students will learn design techniques such as Class-Responsibility-Collaborator (CRC) cards and the Unified Modeling Language (UML), and gain experience with tools to support large-scale software development such as a version control system and a test framework. Students will apply these concepts, techniques, and tools in a semester-long, team software development project. Weekly laboratory sessions will include time for design critiques, code reviews, and supervised teamwork. *Prerequisite:* Computer Science 270.

400-404 Special Topics in Computer Science
1-4 credits
A course which examines special topics in computer science at the advanced level. *Prerequisites:* Computer Science 167 and 270. Any current offerings follow.

467 Numerical Analysis
Not offered 2019-20 3 credits
An introduction to numerical approximation of algebraic and analytic processes. Topics include numerical methods of solution of equations, systems of equations and differential equations, and error analysis of approximations. May be elected as Mathematics 467. *Prerequisite:* Computer Science 167. *Pre- or corequisite:* Mathematics 240.

481, 482 Independent Study
Fall, Spring Staff 1-4 credits
Directed study or research in selected areas of computer science. A curriculum or project is designed by the student(s) with the advice and consent of an instructor in the department. Inquiry may emerge from prior course work or explore areas not covered in the curriculum. *Prerequisite:* consent of instructor.

495 Capstone Project I
Fall Stratton 2 credits
First semester of a team project integrating skills and concepts from across the computer science curriculum. Students will develop project management and communication skills. In writing and documenting software, students will consider their responsibilities to future users or developers. Open only to senior computer science majors. *Prerequisite:* a 300-level computer science course.

496 Capstone Project II
Spring Stratton 1 credit
Second semester of a team project integrating skills and concepts from across the computer science curriculum. Students will develop project management and communication skills, culminating in a public presentation. In writing and documenting software, students will consider their responsibilities to future users or developers. All course work will be completed by the second Friday in March. *Prerequisite:* Computer Science 495.

497 Advanced Project
Spring Stratton 1 credit
Students will individually design, implement, document, and present an extension of the team capstone project developed in Computer Science 495 and 496. *Prerequisite:* Computer Science 495. *Corequisite:* Computer Science 496.

498 Honors Project
Spring Stratton 1 credit
Students will individually design, implement, document, and present an extension of the team capstone project developed in Computer Science 495 and 496. Required of and limited to senior honors candidates in computer science. *Prerequisite:* Computer Science 495. *Corequisite:* Computer Science 496.