Department of Computer Science

Chair of the Department: Robert Boothe
Professors: Alagic, Welty; *Associate Professors:* Boothe, Briggs, MacLeod; *Assistant Professor:* Congdon; *Adjunct Faculty:* Bantz, El-Taha, Heath, Houser

The Department of Computer Science offers a four-year program leading to a B.S. in computer science. Computer science courses concern the theory and practice of solving problems by computer. More specifically, computer scientists build and analyze tools that allow complex problems to be solved. A component of computer science is the study and use of various programming languages, but computer science consists of much more than programming. The mathematical theory of computer science aids in determining the efficiency and correctness of algorithms and programs. In addition, a computer scientist must understand how computers are built and operate. The systematic application of general methods and computing technology to actual problems is also part of computer science.

The undergraduate degree in computer science prepares students both for careers in the computing profession and for graduate study. Course requirements ensure that students receive instruction in both practical and theoretical aspects of computer science. The B.S. degree in computer science is accredited by the Computing Accreditation Commission (CAC) of ABET, the national board that accredits computer, engineering, and technology programs (see www.abet.org).

Program Objectives

At the time of graduation, USM computer science students will be prepared for careers and/or graduate school. In three to five years, graduates of the USM computer science program will

1. have successful professional careers
2. be valued, ethical members of their profession and society
3. be actively involved in continuing their professional education

Programs and Requirements

Bachelor of Science in Computer Science

All students are reminded that, in addition to meeting departmental requirements for a major, they must also meet the University Core curriculum requirements.

The total number of credits for graduation is 120.

Courses used to fulfill major requirements in sections A through F below must be passed with a grade of C– or better. The accumulative grade point average of all courses applied to the major must be at least 2.0.

The specific course requirements are as follows.

A. Computer Science:

COS	160	Structured Problem Solving: Java
COS	161	Algorithms in Programming
COS	170	Structured Programming Laboratory
COS	250	Computer Organization
COS	255	Computer Organization Laboratory
COS	285	Data Structures
COS	350	Systems Programming
COS	360	Programming Languages
COS	450	Operating Systems
COS	485	Design of Computing Algorithms
COS	499	Ethical Conduct and Social Responsibility

B. Software Design:

COS	420	Object Oriented Design
		or
COS	430	Software Engineering

C. Completion of four additional COS courses numbered 300 and above, excluding COS 498.

Graduate courses in the Computer Science Department can be used to fulfill the requirements in section C.

D. Mathematics requirement

(1) Completion of:

MAT	152D	Calculus A
MAT	153	Calculus B

MAT 145 Discrete Mathematics I
COS 280 Discrete Mathematics II
MAT 380 Probability and Statistics (MAT 281 and MAT 282 may together substitute for MAT 380)

(2) One additional mathematics course from the following courses:
MAT 252 Calculus C
MAT 292 Theory of Numbers
MAT 295 Linear Algebra
MAT 350 Differential Equations
MAT 352 Real Analysis
MAT 355 Complex Analysis
MAT 364 Numerical Analysis
MAT 366 Deterministic Models in Operations Research
MAT 370 Non-Euclidean Geometry
MAT 383 System Modeling and Simulation
MAT 395 Abstract Algebra
MAT 460 Mathematical Modeling
MAT 461 Stochastic Models in Operations Research
MAT 490 Topology
MAT 492 Graph Theory and Combinatorics

E. (1) Completion of a two-semester sequence of either
CHY 113K with CHY 114K and CHY 115 with CHY 116
or
PHY 121K with PHY 114K and PHY 123 with PHY 116
or
BIO 105K with BIO 106K and BIO 107

(2) Two additional courses (except introductory courses such as ELE 100) from any of the departments of Engineering, Biological Sciences, Chemistry, Physics, Geosciences, or Environmental Science and Policy, provided that the course is a course in science that can be used for credit toward the degree offered by that respective department. Courses from other departments may also be approved to satisfy this requirement, provided they have a strong emphasis on quantitative measures and the application of the scientific method. For a course with an associated lab to satisfy this requirement, the lab must also be taken. ELE 172 and ELE 271 are excluded because of similarity to COS courses.

F. Communication skills requirement:
(1) Completion of THE 170F
(2) Completion of ITP 210

G. Successful completion of 30 credit hours in the humanities, arts, or social sciences. Courses in these disciplines that satisfy Core curriculum requirements also satisfy this requirement.

Suggested Schedule

The following schedule of mathematics and computer science courses is typical for the freshman and sophomore years.

	Fall	*Spring*
First year	COS 160	COS 161
	COS 170	MAT 152D
	MAT 145	
Second year	COS 280	COS 250
	COS 285	COS 255
		MAT 153

Double Majors

Students who opt to pursue a double major with computer science as one of the major fields of study must satisfy all computer science degree requirements. The general requirements for a double major are listed under the heading Double Major in this catalog.

Minor in Computer Science

A minor in computer science may be obtained by successfully completing the following courses with an accumulative grade point average of 2.0 in these courses: COS 160 and COS 170; COS 161; COS 250 and COS 255; COS 285, and two additional COS courses numbered 250 or greater, excluding COS 498 and COS 499.

Course Fees

Course fees to cover the cost of materials and supplies are assessed in some computer science courses.

COS 120E Deductive Logic
This course will teach the formal techniques that allow one to infer valid conclusions from valid premises in ordinary English discourse. These rules of propositional and predicate logic are based solely on the logical form of statements, independent of their content, and can be used to analyze rational arguments in any discipline. English sentences will be translated into a more precise symbolic language that elucidates their structure. Prerequisite: successful completion of the USM mathematics proficiency requirement. Cr 3.

COS 131 Web Programming
In this course students develop actual Web pages and acquire basic skills in Web programming languages such as JavaScript, VBScript, and Active Server Pages. Other topics include: object model, event model, dynamic HTML, Web-enabled databases, and XML. Prerequisite: a programming course or instructor permission. Cr 3.

COS 141 Visual Basic I
Visual Basic is used to introduce students to the fundamental skills of problem solving and programming. The class includes both classroom presentation and instructor-guided laboratory sessions. Small to medium size programming projects are completed. Prerequisite: a working knowledge of the Windows operating system. Cr 3.

COS 142 Visual Basic II
The concepts of COS 141 are extended to present programming in an application development environment. Topics covered in this course include static and dynamic arrays, user defined data types, class modules and dynamic link library development, serial and random access files, database connectivity, advanced Visual Basic controls, ActiveX controls, and Web programming using Active Server Pages. Component Object Modeling (COM) topics will be included. Small to medium size programming projects and a comprehensive final project will be completed. Prerequisite: COS 141 or permission of the instructor. Cr 3.

COS 160 Structured Problem Solving: Java
An introduction to the use of digital computers for problem solving, employing the Java programming language as a vehicle. Content includes elementary control structures and data representation methods provided by Java and the top-down programming methodology. Course requirements include a substantial number of programming projects. This course must be taken concurrently with COS 170. Prerequisite: successful completion of the USM mathematics proficiency requirement. Cr 3.

COS 161 Algorithms in Programming
The development of algorithms and their implementations in a higher-level programming language, with emphasis on proper design principles and advanced programming concepts. Introduction to the performance analysis of algorithms. Course requirements include substantial programming projects. Prerequisites: COS 160, and working knowledge of word processing and Web browsing. Cr 3.

COS 170 Structured Programming Laboratory
Computational experiments will be designed to teach students how to construct reliable software using Java. Topics to be covered include: Windows system, conditional program flow, iteration, procedures and functions, and symbolic debugging. This course must be taken concurrently with COS 160. Cr 1.

COS 211 The C Programming Language
A first course in the C programming language. This course should provide students with fundamental skills of C programming. Small to medium size programming projects will be written. Prerequisite: A previous course in problem solving and programming (e.g., COS 160) or instructor's permission. Cr 3.

COS 212 The UNIX Operating System
Introduction to the UNIX system, system commands, standard editors, shells, and more. Prerequisite: Experience with some computer operating system. Cr 1.

COS 214 C++ for Programmers
Introduces the basics of C++ programming. Covers types, expressions, control structures, functions, and a brief introduction to classes and objects. This course will prepare students for further object-oriented courses. Prerequisite: a previous college-level course in problem solving and programming or instructor's permission. Cr 3.

COS 215 Introduction to C++ for C Programmers
The changes in the syntax and semantics of C that resulted in C++ are presented. C++ will be covered up to and including a brief introduction to classes and objects. This course will prepare students for further object-oriented courses. Prerequisite: a previous course in problem solving and programming in C (e.g., COS 211) or instructor's permission. Cr 1.

COS 230 Programming in COBOL
A study of the programming language used primarily in business. Prerequisite: COS 160 or analogous experience. Cr 3.

COS 241 Java Programming
Students will develop software using the Java programming language. Some work on connecting Java applets to Web pages using HTML will be included. Prerequisite: a previous course in problem solving and programming, or instructor's permission. Cr 3.

COS 246 Programming Topics
Topics to be covered may include programming languages not otherwise offered (e.g., Ada, Smalltalk), different programming methodologies (e.g., object-

oriented programming), assembly languages, and other specific areas of programming. Prerequisite: COS 161 or permission of instructor. Cr 3.

COS 250 Computer Organization
The basic hardware, architecture, and software of computer systems are covered. Subjects include digital logic design, microprogramming, machine languages, assembly languages, and operating systems. Prerequisite: COS 161. This course must be taken concurrently with COS 255. Normally offered only in the spring semester. Cr 3.

COS 255 Computer Organization Laboratory
Students design, build, and test combinational and sequential logic circuits and write assembly language programs. This course must be taken concurrently with COS 250. Cr 1.

COS 280 Discrete Mathematics II
Concepts of modern algebra, set theory, Boolean algebra, elements of graph theory, and their application to computer science. Prerequisites: MAT 145 and COS 160. Cr 3.

COS 285 Data Structures
Basic abstract data types and their representations, fundamental algorithms, and algorithm analysis. Consideration is given to applications. Specific topics include linked structures, trees, searching and sorting, priority queues, graphs, and hashing. Course requirements include a substantial programming component. Prerequisites: COS 161, MAT 145, and calculus, or their equivalents. Cr 3.

COS 350 Systems Programming
A study of systems programming concepts and software, including the C programming language and the Unix programming environment and operating system interface. Students develop their abilities in these areas through programming exercises and projects. Prerequisites: COS 250, COS 285. Cr 3.

COS 360 Programming Languages
Examination of basic concepts and constructs of high-level languages via consideration of several representative languages. Topics include scope of declarations, binding time of constituents, type checking, and control organization. Study of implementation methods for various programming language features. Study of the methods of formal specifications of programming languages: regular, context-free, and attribute grammars and operational semantics. Exposure to programming in languages that deviate from the imperative style such as Lisp and Prolog. Prerequisites: COS 250, COS 285. Cr 3.

COS 368 Graphical User Interface Design
Principles of graphical user interface design are utilized to build working interfaces. The programming language used may vary from offering to offering. Possible languages include Java and C++. Students will work in an object-oriented, event-driven environment. Prerequisite: COS 285 or instructor's permission. Cr 3.

COS 374 Numerical Analysis
A study of the theory and application of computational algorithms for interpolation, equation solving, matrix methods, integration; error analysis. Prerequisites: MAT 252, MAT 295, COS 160, and permission of instructor. Cr 3.

COS 399 Programming Autonomous Robots
Introduction to the programming concepts involved with autonomous robotic systems. Using off-the-shelf "robot kits" students will design a simple robotic platform to meet specific goals. Then, using a common platform for the remainder of the course, students will develop their programming capabilities. Simple open-ended, feedback, and artificial intelligence systems will be explored throughout the course. Several benchmarks and robot competitions will be used to demonstrate the platform and programming learned in the course. Prerequisite: COS 285. Cr 3.

COS 400 Introduction to Simulation Modeling
Introduction to general principles of discrete event simulation modeling. Topics include design of simulation models, their implementation in a computer simulation language, and analysis of simulation data. Applications will emphasize computer and communication science. Requirements include a substantial programming component. Prerequisites: COS 285 and MAT 380 or equivalent. Cr 3.

COS 420 Object-Oriented Design
This course will focus on the construction of object-oriented software. Students will learn conceptual models for organizing objects and object hierarchies, an object-oriented design notation, the application of design patterns, and the use of software development methodologies such as the Agile development process. The capabilities will be used to solve relatively complex problems in a group setting. Prerequisite: COS 285. Cr 3.

COS 430 Software Engineering
Study of methods applied to large-scale software development, including topics such as requirements analysis and specification, design, validation and verification, and project management, with emphasis on principles of design. Students use methods on a large programming project. Prerequisites: COS 250, COS 285. Cr 3.

COS 444 Software Project Management
The course covers project life cycle, including developing the charter, plans and justification, outsourcing and procurement decisions, scope management, time and cost estimation, quality control, personnel management, risk assessment, and the critical role of communication, both internal and external, to the project. Students will learn to lead and participate in significant software projects. Experts from industry will present case studies of success and failure. Prerequisite: COS 420 or COS 430. Cr 3.

COS 450 Operating Systems
Bottom up construction of a layered operating system beginning with the hardware interface and ending with the user interface. Specific topics covered include concurrent processes, process management, I/O, virtual memory, file management, resource scheduling, and performance measurement. Students are assumed to be familiar with general machine architecture, functions of system software (compilers, loaders, editors, etc.), data structures, and to have some experience with UNIX or another multiprogramming operating system. Prerequisites: COS 250, COS 350. Cr 3.

COS 452 Computer Graphics
A study of the techniques involved in computer graphics systems. Topics include: point-plotting and line drawing in two- and three-dimensional space; clipping and windowing; geometric modeling; algorithmic solutions to the hidden line and hidden surface problems. Prerequisite: COS 285. Cr 3.

COS 455 Computer Architecture
Fundamentals of the design and organization of digital computers. Topics include applications of Boolean algebra to logical design; machine algorithms used in addition, subtraction, multiplication, etc.; types of memory; synchronous and asynchronous operation; minimization of logic circuits. Also, concepts from microprocessors and large parallel computers. Prerequisite: COS 250. Cr 3.

COS 457 Database Systems
Study of the methods and principles of database management systems (DBMS). Topics addressed include DBMS objectives and architecture, data models, data definition and data manipulation languages, and providing Internet access to databases. The entity-relationship and relational models are emphasized and their use required in a design project. Prerequisites: COS 280, COS 285. Cr 3.

COS 460 Computer Networks
An introduction to computer networks. Computer network architecture is described. Other topics include digital data communication, local area networks, wide area networks, internetworks, and the Internet. Specific technologies, including Ethernet and ATM, and protocols, including TCP/IP, will be considered in detail. Prerequisite: COS 285. Cr 3.

COS 465 Distributed Systems
An introduction to the design and operation of distributed systems. Topics include client-server models, interprocess communications, RPC, replication and consistency, online transaction processing, error and fault recovery, encryption and security. Examples will be taken from extant distributed systems. Prerequisites: COS 450 and COS 460, or their equivalents, or permission of the instructor. Cr 3.

COS 467 Performance Analysis of Distributed Systems
The objective of the course is to learn techniques that enable assessing the performance of applications running on distributed systems. This is an important topic because software developers should have a good understanding regarding the performance of the distributed applications they develop. This course presents techniques such that the performance of distributed applications can be evaluated. Topics to be covered include queueing theory, simulation, availability, and performability modeling. Other techniques used to assess the performance of distributed systems will be introduced as needed. Prerequisites: COS 450 or COS 460, and MAT 281 or MAT 380. Cr 3.

COS 469 Compiler Construction
Definition of languages via context-free grammars. Organization of a compiler into phases of lexical analysis, parsing, code generation, and optimization. Students will implement a compiler for a Pascal-like language. Prerequisite: COS 360. Cr 3.

COS 470 Topics in Computer Science
Topics to be covered may include philosophy of computers, history of computers, computers and society, simulation, graphics, and other advanced topics. Prerequisite: COS 285 or permission of the instructor. Cr 3.

COS 471 Advanced Database Systems
This course covers object-oriented and XML database technologies, their interfacing and integration. Object-oriented topics include developments from industrial standards such as ODMG and Java Data Objects, query languages such as OQL, Java database technology, object-relational systems, and language integrated queries such as LINQ. The XML technology is represented by schema languages such as XML Schema and query languages such as XQuery. Object-oriented interfaces to XML include DOM, LINQ to XML, LINQ to XSD, as well as other industrial developments. The course includes hands-on experience with advanced database management systems. The requirements include an object-oriented software and database development project, addressed by teams, and a term paper. Prerequisite: permission of the instructor. Cr 3.

COS 472 Artificial Intelligence
An introduction to the underlying concepts and applications of intelligent systems. Topics include pattern matching, production systems, computer representations of knowledge, heuristic search techniques and computer problem solving, and automatic theorem proving. The programming language Lisp is introduced. Students will develop programs in Lisp applying the concepts and techniques introduced. Cr 3.

COS 476 Advanced Object-Oriented Design
This course considers developing object-oriented, multi-tier, Web-based applications. Topics will include object-oriented design patterns in distributed environments, software components, and software

frameworks. The course also has a significant hands-on implementation component, and after having completed this course, students will have practical experience with several leading distributed object technologies, including AJAX, Web Services, Enterprise JavaBeans, JDBC, and Servlets. The course is structured so that students will work in teams to develop a medium-sized, multi-tier application that incorporates several of the technologies mentioned above. Lectures will provide an introduction to the technologies and discuss principled ways to apply these technologies. Prerequisite: COS 420 or permission of instructor. Cr 3.

COS 478 Advanced Java Technology
The goal of this course is to provide an in-depth study of the most important and the more advanced components of the Java technology. The course covers topics such as concurrent object-oriented programming in Java, Java Core Reflection, the underlying virtual platform (the Java Virtual Machine), genericity (parametric polymorphism), persistence, and assertions. Programming assignments include concurrent programming, programming with parametric collection types, dynamic loading and compilation, usage of the Java reflective capabilities, and usage of persistent capabilities available in Java and in its extensions. The outcome of this course is a high-level of professional expertise in the overall Java technology. Prerequisite: COS 360. Cr 3.

COS 479 Object-Oriented Software Technology
This course combines formal and practical object-oriented software techniques in developing the following main themes of object-oriented software technology: (i) object-oriented software systems that provide efficiency and reliability based on an advanced type system and (ii) correctness and behavioral compatibility in software re-use based on object-oriented assertion languages and programming by contract methodology. Practical implications and usage of the general notions such genericity, self typing and reflection in complex software systems will be based on type systems of major object-oriented languages such as Java, C# and Eiffel. The assertion languages demonstrating the main themes in this course are JML (Java Modeling Language) and Spec# (an assertion language for C#). The programming assignments are based on a pragmatic methodology for object-oriented software construction (programming by contract) along with the associated tools including program erification techniques and systems. Prerequisite: COS 360. Cr 3.

COS 480 Theory of Computation
Study of the theoretical foundations of computer science, including elements of set theory and logic, the specification of formal languages via finite automata, regular expressions, push-down automata, context free grammars, and Turing machines. Also introduces the concepts of recursive and recursively enumerable sets. Prerequisite: COS 280. Cr 3.

COS 485 Design of Computing Algorithms
An introduction to the design and analysis of algorithms. Techniques for designing algorithms, such as divide-and-conquer, greedy method, dynamic programming, and backtracking are emphasized and illustrated. Many problems of practical importance are covered including: minimum spanning tree, single source shortest path, traveling salesperson, and graph search. The concepts of NP-completeness are also considered. Substantial programming in a high-level language. Prerequisite: COS 285. Cr 3.

COS 495 Advanced Web Architectures
The focus of communication over the Internet is shifting to computer-to-computer interaction. Standards for this interaction (ebXML, SOAP, WSFL) are now in place and maturing, and commercial use is exploding. We will survey these standards and evaluate their security, efficiency, and completeness. We will construct several case studies, including Web-based commerce. As a team, we will acquire and learn how to use available tooling, and we will put together working Web services and test their ability to interact with each other. Prerequisites: COS 285 and junior standing. Cr 3.

COS 497 Independent Study in Computer Science
An opportunity for juniors and seniors who have demonstrated critical and analytical capability to pursue a project independently, charting a course and exploring an area of interest within their major field. At most, three credits of COS 497 can be used to satisfy requirement C, as noted under Programs and Requirements. Prerequisites: junior or senior standing and permission of the Department chair and instructor. Cr 1-3.

COS 498 Computer Science Internship
An opportunity for students to gain practical experience in computer science-related employment. The University internship program provides placement. This course is offered on a pass/fail basis only, does not fulfill any computer science requirement, and a maximum of 6 credits may be taken. Cr 3.

COS 499 Ethical Conduct and Social Responsibility
A study of ethical perspectives and social responsibilities of computer professionals. Assigned readings provide the basis for class discussions of such issues as social control and privacy, computer viruses, ACM code of professional conduct, hacking, limits of correctness in computer software, military influence on computer science research and education. Prerequisite: senior standing. Normally offered in spring semester only. Cr 1.