Volgenau School of Engineering

APPLIED COMPUTER SCIENCE, B.S.

Concentrations in Bioinformatics, Computer Game Design, Geography, and Software Engineering
2016 - 2017

The Bachelor of Science degree in Applied Computer Science (BS ACS) has been created for those students who want and need the knowledge and expertise of computer science to work in one of the many disciplines that require advanced computing techniques. These fields do not merely "use" computing but create new and interesting problems for the computer scientist.

The objectives of the BS ACS program are to provide students with the following:

1. The fundamental knowledge regarding theory, methods and applications of Computer Science.
2. A foundation in a second chosen discipline.
3. Knowledge of concepts that integrate Computer Science with the second chosen discipline using senior level classes that focus on the emerging issues.
4. Preparation for employment as a computational expert in a non-computer science discipline.
5. Preparation for graduate studies in fields such as Computer Science, their second discipline and related computational areas.

The curriculum covers the foundational knowledge of two disciplines, and provides a more substantial cross-disciplinary background than if the student pursued a major in Computer Science and a minor in a second discipline (or a major in a second discipline with a minor in Computer Science). At the upper division level, the student will take a set of carefully chosen classes in each discipline along with one or two classes that bring the disciplines together. This provides a solid background in two disciplines so that undergraduate students will be positioned to enter these computational fields immediately or to pursue a graduate degree in an emerging discipline.

Degree Requirements

The ACS degree program can be successfully completed within the normal 120 credit hours. In addition to Mason Core requirements, including humanities, and social science, the BS ACS program requires foundation, core, and concentration courses as described below. These course requirements provide the student with expertise in programming, computer systems, software engineering, formal methods and analysis of algorithms.

ACS Foundation Courses: CS 101, 105, 112, 211; MATH 113, 114, 125, 203.
ACS Core: CS 262, 310, 321, 330, 367, 465, 483; ECE 301.
One CS course numbered above 400.

There are currently four concentration areas for this degree: ***bioinformatics***, ***geography***, ***computer game design***, *and* ***software engineering***. Concentration requirements are briefly described on the next page.

All BS ACS majors must complete at least 36 additional credits to meet the course requirements of one of the concentration areas. These credits will include either STAT 344 (Statistics and Probability) or a course in Statistics relevant to the concentration. See the reverse side for concentration requirements.

For complete information on the requirements for the concentrations in bioinformatics, computer game design, geography, or software engineering, please visit the GMU catalog at http://catalog.gmu.edu or the Computer Science Department web page at http://cs.gmu.edu/programs/undergraduate/#BSACS

We invite requests for additional information. Please contact:

Computer Science Department
Volgenau School of Engineering
George Mason University Mail Stop 4A5
Fairfax, VA 22030-4444

Phone: (703) 993-1530
Fax: (703) 993-1710
E-mail: csadmin@cs.gmu.edu
Internet: http://cs.gmu.edu

(over)

BIOINFORMATICS CONCENTRATION

Foundation:
PHYS 160/161, CHEM 201, BIOL 213,
CS 306 and STAT 344
Core:
BINF 450, BIOL 482, BIOL 580, CS 450
BINF 401 or CS 444
BINF 402 or CS 445
Two approved electives related to bioinformatics

Bioinformatics Specific Courses

CS 444 Introduction to Computational Biology. This course introduces students to computational methods in molecular biology. The course will cover a broad array of topics in bioinformatics and computational biology and will be organized as three four-week modules. The modules are intended to capture the current classification of bioinformatics and computational biology methods and so to provide students with a broad view of the field. The modules will range from sequence-centric bioinformatics, to structure-centric computational biology, and system-centric algorithmic systems biology. The class will give students the ability to obtain hands-on experience and expertise through design and programming projects.

CS 445 Computational Methods for Genomics. Fundamental principles and techniques for implementing computational algorithms to solve problems in biology arising from the need to process large volumes of genomic information. Topics include sequence analysis, alignment, and assembly, gene prediction, and knowledge-based protein structure prediction. Projects involve designing and programming basic alignment and prediction methods.

COMPUTER GAME DESIGN CONCENTRATION

Foundation:
AVT 104; CS 225, 306, 325, 351; STAT 344
Core:
CS 425, 426, 451; AVT 382, 383
One approved elective chosen from:
CS 332, CS 455, CS 475, CS 480, CS 485,
SWE 432, GAME 332, AVT 370,
AVT 374, AVT 487
Natural Science: PHYS 160/161 and
one additional lab science.

Computer Game Design Specific Courses

CS 225 Culture and Theory of Games. Explores the theory, history, culture, and lore of games with particular emphasis on the varieties of computer game environments.

CS 325 Computer Game Design. Game design, in various electronic entertainment technologies, involves a diverse set of skills and backgrounds from narrative and art to computer programming. This course surveys the technical aspects of the field, with an emphasis on programming.

CS 351 Visual Computing. The focus of this course is programming essential mathematical and geometric concepts underlying computer graphics. It covers fundamental topics in computational geometry, 3D modeling, graphics algorithms, and graphical user interfaces using both 2D and 3D implementations. Furthermore, it reinforces object oriented programming practices.

CS 425 Game Programming I. An introduction to the technologies and techniques used in modern computer games. Teams will learn the facets of a complete design, using sophisticated tools. The course will involve a project in which a game is prototyped; this prototype and initial design will serve as the starting point for the project in CS426.

CS 426 Game Programming II. This project-oriented course is a continuation of CS 425 with an emphasis on the implementation of a complete game.

GEOGRAPHY CONCENTRATION

Foundation:
GGS 101, 102, 103, 110, 300;
CS 306; STAT 344.
Core:
GGS 310, 311, 411, 412, 416 and 463
One GGS course numbered above 300

SOFTWARE ENGINEERING CONCENTRATION

Foundation:
CS 306, STAT 344
Core:
SWE 205, 301/401, 332, 437
SWE Related: 15 hours chosen from
CS 455, CS 450, CS 463, CS 468, CS 471,
CS 475, SWE 432, SWE 443
Cross Disciplinary:
ENGH 388 and one of
PSYC 333, COMM 320, COMM 335

Software Engineering Specific Courses

SWE 205 Software Usability Analysis and Design. Explores concepts for objectively and quantitatively assessing the usability of software user interfaces.

SWE 301 Internship Preparation/SWE 401 Internship Reflection. Preparation for, then reflection on, the Internship Educational Experience.

SWE 332 Object-Oriented Software Design & Implementation. In-depth study of software design and implementation using a modern, object-oriented language with support for graphical user interfaces and complex data structures.

SWE 432 Design and Implementation of Software for the Web. This course teaches students how to develop software for web applications.

SWE 437 Software Testing and Maintenance. Concepts and techniques for testing and modifying software in evolving environments.

SWE 443 Software Architectures. This course teaches how to design, understand, and evaluate software systems at an architectural level of abstraction .