GEORGE MASON UNIVERSITY | The Volgenau School of Engineering

COMPUTER SCIENCE, B.S.

2014-2015

The objectives of the B.S. program in Computer Science relate to the abilities of the graduates several years after graduation. The objectives include

- Foundation for successful careers in industry: Graduates of the program will have a broad understanding of the fundamental concepts, methodologies and tools, and applications of computer science. They will have the educational foundation that leads to successful careers in the computing industry.
- Foundation for graduate study: Graduates of the program will have the academic preparation for successful completion of rigorous graduate programs.
- Professional preparation: Graduates will have effective written and oral communication skills, and be able to work collaboratively with others in a professional and ethical manner.

This bachelor's degree program is accredited by the Computing Accreditation Commission of ABET, http://www.abet.org.

Admission Requirements

Admission to George Mason is competitive. Each candidate who presents sufficient admission qualifications is reviewed in the context of other qualified applicants. An offer of admission is valid only for the semester for which the student applied. Application for undergraduate admission should be made to the Office of Admissions. Please consult http://admissions.gmu.edu for additional information. (See the last page of this document for details about admission to the BS/Accelerated MS programs.)

Freshman Admission Requirements

The following factors are considered when reviewing applications for admission:

- Cumulative high school grade point average for course work completed in grades 9 through 12.
- Level of difficulty of course work elected throughout the high school years, particularly in English, mathematics, laboratory science, social studies, and foreign language.
- Scores from Scholastic Aptitude Test I (SAT I) or the American College Test (ACT).
- Test of English as a Foreign Language (TOEFL) where applicable.
- Essays, list of extracurricular activities, and teacher and guidance counselor recommendations.

Transfer Admission Requirements

The university accepts qualified students who wish to transfer from other colleges. Transfer applicants must submit official transcripts from each collegiate institution attended. Transfer applicants with fewer than 30 semester hours of transferable credit must also submit a copy of their secondary school record and test scores. All non-native English speakers are also required to submit a TOEFL or IELTS score or acceptable grades (C or better) in at least two English composition or literature classes taken at another U.S. university or college.

Undecided Students

Students who are undecided but interested in pursuing a career in Computer Science should seek the advice of a departmental faculty advisor. Sample schedules of the majors within the Volgenau School of Engineering are also available from each individual department.

Change of Major

Students requesting a change of major to computer science must have a GPA of at least 2.75 and successfully completed CS 112 or 211 and MATH 113, 114, or 125.

Advanced Placement, Credit by Exam

A score of 3 on the Advanced Placement (AP) computer science exam qualifies the student for credit in CS 112. An AP score of 5, together with demonstrated competence in the programming language used in CS 211, qualifies students for credit in CS 211. A score of 4 on the International Baccalaureate (IB) computer science exam qualifies students for credit in CS 112, and a score of 5 or more qualifies students for credit in CS 211.

Degree Requirements

Undergraduate degree work in computer science provides students with essential background for studying the design and implementation of computer system software, computer architecture, and computer software applications for science and business. The program emphasizes both computer system fundamentals and computer software applications. Required areas of study include data structures, analysis of algorithms, low-level programming, computer architecture and language translation, ethics and law for the computing professional, and software design and development. Evolving software technologies are a major concern. The BS in Computer Science program also requires 12 credits of natural science and 20 credits in mathematics and statistics, including calculus, discrete mathematics, linear algebra, and applied probability theory.

A sample schedule that fulfills degree requirements for a Bachelor of Science in Computer Science degree is shown below. With prior approval of department advisors, some courses may be taken out of the indicated sequences, particularly in the case of Mason Core [MC] courses.

Sample Schedule for B.S. in Computer Science

First Semester		Second Semester	
CS 101 Preview of Computer Science	2	CS 105 Computer Ethics and Society	1
CS 112 Intro Computer Programming	4	CS 211 Object-Oriented Programming	3
MATH 113 Calculus I	4	MATH 114 Calculus II	4
ENGH 101 Composition [MC]	3	Arts [MC]	3
Western Civilization [MC]	3	Literature [MC]	3
Total Hours	16	Total Hours	14
Third Semester		**Fourth Semester**	
CS 262 Intro Low-level Programming	2	CS 310 Data Structures	3
MATH 125 Discrete Mathematics	3	COMM 100 Public Speaking [MC]	3
MATH 213 Calculus III	3	ECE 301 Digital Electronics	3
Social and Behavioral Science [MC]	3	Natural Science with lab	4
Natural Science with lab	4	Elective	3
Total Hours	15	Total Hours	16
Fifth Semester		**Sixth Semester**	
CS 330 Formal Methods & Models	3	CS 321 Software Requirements and Design	3
CS 367 Computer Systems & Programming	3	CS 465 Computer Systems Architecture	3
MATH 203 Linear Algebra	3	STAT 344 Probability and Statistics	3
ENGH 302 Advanced Composition [MC]	3	Natural Science	4
Global Understanding [MC]	3	Humanities	3
Total Hours	15	Total Hours	16
Seventh Semester		**Eighth Semester**	
CS 306 Synthesis - Ethics & Law [MC]	3	Senior CS course	3
CS 483 Analysis of Algorithms	3	Senior CS course	3
Senior CS Course	3	Senior CS course	3
Senior CS Course	3	CS-related Elective	3
CS-related Elective	3	Elective	1
Total Hours	15	Total Hours	13

B.S. DEGREE IN COMPUTER SCIENCE
FROM 2014-2015 CATALOG

(The GMU catalog is the official reference – see catalog.gmu.edu)

- **MASON CORE [MC] REQUIREMENTS** (24 credits)

FOUNDATION

- ✓ ENGH 101 Composition [Non-native English speakers see catalog about using ENGH 100]
- ✓ ENGH 302 Advanced Composition (Natural Science section)
- ✓ COMM 100 Public Speaking
- ✓ [Information Technology is satisfied by major requirements]
- ✓ [Quantitative Reasoning is satisfied by major requirements]

CORE [see university catalog for currently approved classes for these requirements]

- ✓ Literature
- ✓ Western Civilization/World History
- ✓ Social and Behavioral Science
- ✓ Global Understanding
- ✓ Arts
- ✓ [Synthesis is satisfied by the major requirement - CS 306]
- ✓ [Natural Science is satisfied by major requirements]

- **MAJOR REQUIREMENTS** (92 credits)

Required Computer Science

- ✓ CS 101 Preview of Computer Science
- ✓ CS 105 Computer Ethics and Society
- ✓ CS 112 Introduction to Computer Programming
- ✓ CS 211 Object-Oriented Programming
- ✓ CS 262 Introduction to Low-Level Programming
- ✓ CS 306 Synthesis of Ethics and Law for the Computing Professional
- ✓ CS 310 Data Structures
- ✓ CS 321 Software Requirements and Design Modeling
- ✓ CS 330 Formal Methods and Models
- ✓ CS 367 Computer Systems and Programming
- ✓ CS 465 Computer Systems Architecture
- ✓ CS 483 Analysis of Algorithms
- ✓ **Senior CS:**
  - CS 463 or CS 471 or CS 475;
  - Four additional courses chosen from: CS 425, 440, 450, 451, 455, 463, 468, 469, 471, 475, 477 480, 482, 484, 485, 490, 499; MATH 446 (or OR 481)

  At most three credits of CS 499 can be used toward the Senior CS requirement.

Mathematics and Engineering

- ✓ MATH 113, MATH 114, MATH 125: Calculus I, II, and Discrete Mathematics
- ✓ MATH 203, MATH 213, ECE 301, STAT 344*: Linear Algebra, Calculus III, Digital Electronics, Probability and Statistics for Engineers & Scientists I

* Students may replace STAT 344 with both MATH 351 and 352 (while also satisfying one CS-related elective.)

Computer Science Related Electives – Any two of the following (not used to satisfy other requirements):

ECE 280, 431, 447, 450, 511; ENGH 388; OR 335, 441, 442; PHIL 371, 376; STAT 354, SWE 432, 437, 443; SYST 371, 470; any MATH course numbered above 300 except MATH 351; any CS course above 300.

Natural Sciences [see next page]

Humanities [see next page]

- **GENERAL ELECTIVES** (4 credits)

Additional academic hours beyond Mason Core and Major requirements.

Humanities Requirement for CS Majors

One additional course (3 credit hours) is required in the Humanities beyond what is required by the Mason Core [MC] requirements. This can be fulfilled by any Mason Core course **except** those listed under Information Technology, Synthesis, Quantitative Reasoning, or Natural Science. Students wishing to substitute alternate courses for this requirement must obtain departmental approval.

Natural Science Requirement for CS Majors

The BS in CS requires 12 credits of natural science. The courses should be intended for science and engineering students and must include a two-course sequence with laboratories (selected from below). Some acceptable combinations have a total of more than 12 hours. As with all courses, be sure that you have the prerequisites.

ASTRONOMY 111(3) & 112(1) and 113(3) & 114(1).

BIOLOGY 103(4) and 104(4).

CHEMISTRY 211(4) and 212(4).

ENVIRONMENTAL SCIENCE (EVPP) 110(4) and 111(4).

GEOLOGY 101(4) and 102(4).

PHYSICS 160(3) & 161(1) and 260(3) & 261(1). You may substitute PHYS 265 for PHYS 261.

Students who wish to study a natural science in depth can choose to take the coursework that students majoring in that science are required to take; these course are more rigorous. More advanced courses may be substituted with the advisor's approval. Students with a strong interest in Biology should consider the BS in Applied Computer Science.

Grade and Credit Hour Requirements for CS Majors

Students must earn a C or better in any course intended to satisfy a prerequisite for a computer science course. Computer science majors may not use more than one course with grade C- or lower toward departmental requirements. (Any course can be repeated and the new grade is used in the computation of the cumulative GPA; see "repeating a course" in the University Catalog.)

Graduation requires 120 total credit hours (at least 30 at GMU) and 45 upper division hours (at least 12 at GMU).

Repeating Courses

Students may attempt an undergraduate course taught by the Volgenau School of Engineering twice. A third attempt requires approval of the department offering the course. This policy does not apply to STAT 250, which follows the normal university policy for repeating undergraduate courses.

Termination from the Major

No math, science, or Volgenau School of Engineering course, required for the major, may be attempted more than three times. Those students who do not successfully complete such a course within three attempts will be terminated from the major. For more information, see the "Termination from the Major" section under Academic Standing in the University Catalog.

Computer Science Courses (CS)

100 Principles of Computing (3:3:0). *Prerequisite: none*. This course is intended to help students learn to think in the manner necessary to fully grasp the nature and power of the digital world around us. The early era of the Internet and the personal computer led to the need for "computer literacy." Now, the changing nature of our global society requires that students learn new ways to think about problems and how to solve them, regardless of students' specific fields of endeavor. Through this course, students will explore major issues related to the "big ideas" of computational thinking (namely, (i) Creativity, (ii) Abstraction, (iii) Data, (iv) Algorithms, (v) Programming, (vi) Internet, and (vii) Societal Impact), as well as how these issues will impact their future lives.

101 Preview of Computer Science (2:2:0**)** *Co-requisite: CS 112.* All computer science majors are required to take this course within their first year. Offers a broad overview of computer science designed to provide students with an introduction to the field of computer science and an orientation to the computer science department and the computing environment at the university. Includes a project to introduce problem solving using computers.

105 Computer Ethics and Society (1:1:0). *Prerequisite: none.* Intensive introduction to legal, social, and ethical issues surrounding software development and computer use. Stresses professional conduct, social responsibility, and rigorous standards for software testing and reliability. Examines issues such as liability, ownership of information, and computer crime.

112 Introduction to Computer Programming (4:3:1). *Prerequisite: Satisfaction of pre-requisites for MATH 113* Rigorous introduction to problem solving through development of computer programs. Focuses on identifying algorithmic patterns in problems, describing problem solutions in high-level pseudocode, then implementing in a procedural programming language. Basic programming concepts are covered in detail including expressions, control structures, simple data types, and input/output. Program testing and debugging are discussed to verify that problems are solved correctly.

211 Object-Oriented Programming (3:3:1). *Prerequisite: Grade of C or better in CS 112.* Thorough treatment of programming according to object-oriented principles. Introduces classes, interfaces, inheritance, polymorphism, and single dispatch as means to decompose problems. Covers intermediate programming techniques including error handling through exceptions, arrangement of source code into packages, and simple data structures. Intermediate debugging techniques and unit testing are covered.

222 Computer Programming for Engineers (3:3:0). *Prerequisites: Grade of C or better in CS 112.* A second course in computer programming. Introduces object-oriented programming, and elementary data structures. The emphasis will be on problems and language features relevant to engineers. (Intended as terminal course in computer programming.)

225 Culture and Theory of Games (3:3:0) Explores the theory, history, culture, and lore of games with particular emphasis on the varieties of computer game environments

261 Introduction to a Second Language (1:1:0). *Prerequisite: Grade of C or better in CS 211*. Advanced programming using Java programming language; other languages may be offered at times. *This course is not available for CS major credit.*

262 Introduction to Low-Level Programming (1:1:0) *Prerequisite: Grade of C or better in CS 211.* Introduction to the language C, as well as operating system concepts, in UNIX, to prepare students for topics in systems programming.

306 Synthesis of Ethics and Law for the Computing Professional (3:3:0). *Prerequisites: CS 105; junior standing (at least 60 credit hours). Co-requisite: all required Mason Core courses.* Computers science majors may use this course to satisfy the Mason Core synthesis requirement, so long as they have not previously taken CS 305 for credit. Practical course to become effective computer professional. Examines legal, ethical issues surrounding computer technology and its use, as well as the foundation building that is necessary to deal with those challenges. Applies philosophical bases for ethical decision-making to modern concerns raised by computers and technology. Addresses topics covered by CS 105 in a more intensive manner, and focuses on the emerging legal and ethical issues involved in e-commerce and widespread use of the Internet.

310 Data Structures (3:3:0). *Prerequisite: Grade of C or better in CS 211 and MATH 113. Co-requisite: CS 105.* This course continues to focus on object-oriented programming with an emphasis on tools and techniques for developing moderate to large programs. Topics include use and implementation of linear and non-linear data structures and the design and analysis of elementary algorithms.

321 (CS/SWE) Software Requirements and Design Modeling (3:3:0). *Prerequisite: Grade of C or better in CS 211.* An introduction to concepts, methods, and tools for the creation of large-scale software systems. Methods, tools, notations, and validation techniques to analyze, specify, prototype, and maintain software requirements. Introduction to object-oriented requirements modeling, including use of case modeling, static modeling, and dynamic modeling using the Unified Modeling Language (UML) notation. Concepts and methods for the design of large-scale software systems. Fundamental design concepts and design notations are introduced. A study of object-oriented analysis and design modeling using the UML

notation. Students participate in a group project on software requirements, specification, and object-oriented software design.

325 Introduction to Game Design (3:3:0) *Prerequisite: Grade of C or better in CS 211.* 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.

330 Formal Methods and Models (3:3:0). *Prerequisite: Grade of C or better in CS 211 and MATH 125.* Abstract concepts that underlie much advanced work in computer science, with major emphasis on formal languages, models of computation, logic, and proof strategies.

332 (CS/SWE) Object-Oriented Software Design and Implementation (3:3:0). *Prerequisite: Grade of C or better in CS 211.* In-depth study of software design and implementation using a modern, object-oriented language with support for graphical user interfaces and complex data structures. Topics covered will be specifications, design patterns, and abstraction techniques, including typing, access control, inheritance, and polymorphism. Students will learn to the proper engineering use of techniques such as information hiding, classes, objects, inheritance, exception handling, event-based systems, and concurrency.

351 Visual Computing (3:3:0). *Prerequisite: Grade of C or better in CS 262 and CS 310.* 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.

367 Computer Systems and Programming (3:3:0).). *Prerequisite: Grade of C or better in CS262 or 222 and ECE 301 or ECE 331.* Introduces students to computer systems from the perspective of a programmer. Topics covered include data representation, assembly and machine level representation of high-level language programs, the memory hierarchy, linking, exceptions, interrupts, processes and signals, virtual memory, and system-level I/O. This course serves as a foundation for courses on compilers, networks, operating systems, and computer architecture, where a deeper understanding of systems-level issues is required.

390 Research and Project Design Principles in Computing (3:3:0). *Prequisite: Grade of C or better in CS 262; CS 310 and 321 highly recommended*. This course introduces students to the research and project design process within the computing field. Students will learn about the tools of the trade, work through design principles beginning with the articulation of a question, reviewing methods of exploration, gathering evidence, communicating results, and assessing and evaluating research or project outcomes.

391 Advanced Programming Lab (1:0:1). *Co-requisite: Grade of C or better in CS 310 and permission of instructor*. Programming-intensive lab course. Students refine their problem solving and programming skills, while gaining experience in teamwork. Focuses on data structures, recursion, backtracking, dynamic programming, and debugging. Central focus is applying familiar and new algorithms and data structures to novel circumstances.

425 Game Programming I (3:3:0) *Prerequisites: Grade of C or better in CS 310, 325 and 351.* The course will provide an introduction to technologies and techniques used in modern computer games. Teams will explore the various 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 CS 426.

426 Game Programming II (3:3:0) *Prerequisite: C or better in CS 425.* This is a project-oriented course. It is a continuation of CS 425 with an emphasis on the implementation of a complete game.

440 Language Processors and Programming Environments (3:3:0). *Prerequisite: Grade of C or better in CS 310, 330 and 367*. Survey of basic programming language processors and software development tools, such as assemblers, interpreters, compilers. Topics include design and construction of language processors, formal syntactic definition methods, parsing techniques, and code generation techniques.

444 Introduction to Computational Biology (3:3:0). *Prerequisite: Grade of C of better in CS 310.* 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.

445 Computational Methods for Genomics (3:3:0). *Prerequisite: Grade of C or better in CS 310 and STAT 344.* 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.

450 Data Base Concepts (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 330.* This course covers from basics to intermediate knowledge for the design, implementation and use of relational database systems. The main topics include the Entity-Relationship (ER) and Entity-Enhanced Relationship (EER) models for database design, Relational Algebra (RA), Structured Query Language (SQL), SQL programming techniques, functional dependencies and normalization, object- and object-relational databases, and security. Students will practice to design, develop, and implement a relational ORACLE database, and use the database for queries, transaction processing, and report generation.

451 Computer Graphics (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 367, and MATH 203.* Basic graphics principles and programming. Topics include scan conversion, transformation, viewing, lighting, blending, texture mapping, and some advanced graphics techniques.

455 Computer Communications and Networking Systems (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 367, and STAT 344.* Data communication and networking protocols, with study organized to follow layers of the Internet Protocol Suite (TCP/IP family of protocols). Topics include role of various media and software components, local and wide area network protocols, network performance, and emerging advanced commercial technologies.

463 Comparative Programming Languages (3:3:0). *Prerequisite: Grade C or better in CS 330 and CS 367.* Key programming mechanisms described independently of particular machines or languages, including control, binding, procedural abstraction, types, and concurrency. Includes basic programming competence in several different types of programming languages, including a language that provides concurrency. (Students who have taken CS 363 may not receive credit for CS 463.)

465 Computer Systems Architecture (3:3:0). *Prerequisite: Grade of C or better in CS 367.* Computer subsystems and instruction set architectures. Single cycle, multiple-cycle, and pipeline architectures. Memory hierarchy, cache, and virtual memory input-output processing.

468 Secure Programming and Systems (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 367.* Fundamental principles and techniques for implanting secure computer systems. Topics include security and cryptography basics, vulnerability analysis, secure software development and distributed system security. Projects involve designing and programming basic security tools, secure programs and distributed systems.

469 Secure Engineering (3:3:0). *Prerequisite: Grade of C or better in CS 310, 367 and STAT 344.* Modern enterprise computers are constantly under attack. A number of devices and subsystems are deployed in the enterprise defense. Course covers the software subsystems that are involved in defending computer systems. We cover threats and a plethora of architecting solutions against them including but not limited to access control and identity management, network and system security, intrusion detection and recovery systems, monitoring and forensic system.

471 Operating Systems (3:3:0). *Prerequisites: Grade of C or better in CS 310 and 367*. Issues in multiprogramming. Covers concurrent processes and synchronization mechanisms; processor scheduling; memory, file, I/O, deadlock management; performance of operating systems. Projects dealing with synchronization in a multi-programmed OS and virtual memory management.

475 Concurrent and Distributed Systems (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 367.* Practical issues in designing and implementing distributed software. Topics include concurrent programming, synchronization, multithreading, local and wide-area network protocols, distributed computation, system integration, and techniques for expressing coarse-grained parallelism at the application level. Projects involve network programming at the application level.

477 Mobile Application Development (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 367.* This project based course will teach fundamental principles of software development for the mobile device environment, emphasizing the application of numerous academic concepts and the new design and programming paradigms that stem from the use of mobile devices. Topics include user interfaces, event-based programming, interprocess communications, networking, mobile-specific capabilities and performance in a resource restricted environment.

480 Introduction to Artificial Intelligence (3:3:0). *Prerequisite: Grade of C or better in CS 310 and 330.* Principles and methods for knowledge representation, reasoning, learning, problem solving, planning, heuristic search, natural language processing, and their application to building intelligent systems in a variety of domains. Uses LISP, PROLOG, or expert system programming languages.

482 Computer Vision (3:3:0). *Prerequisite: Grade of C or better in MATH 203, STAT 344 and CS 310.* Basic principles of visual perception and their implementation on computer systems. Topics include early visual processing, edge detection,

segmentation, intrinsic images, image modeling, representation of visual knowledge, and image understanding. Students complete projects involving real images.

483 Analysis of Algorithms (3:3:0). *Prerequisite: Grade C or better in MATH 125 and CS 310 and 330.* Analyzes computational resources for important problem types by alternative algorithms and their associated data structures, using mathematically rigorous techniques. Specific algorithms are analyzed and improved.

484 Data Mining (3:3:0). *Prerequisite: Grade of C or better in CS 310 and STAT 344.* Basic principles and methods for data analysis and knowledge discovery. Emphasizes developing the basic skills needed for modeling and prediction, on one side, and performance evaluation, on the other. Topics include system design; data quality, preprocessing, and association; event classification; clustering; biometrics; business intelligence; and mining complex types of data.

485 Autonomous Robotics (3:3:0). *Prerequisites: CS 310, CS 262, MATH 203, or permission of the instructor.* This course covers various basic software topics in autonomous robotics, including autonomous architectures, elementary kinematics and controls, simulation, localization and mapping, reasoning, and multiagent environments. The course will have several projects involving physical robots.

490 Design Exhibition (3:3:0). *Prerequisite: Grade of C or better in CS 321, 483, two other CS 400-level courses and senior standing.* Capstone course focusing on the design and successful implementation of a major software project, encompassing a broad spectrum of knowledge and skills, developed by a team of students. Requires final exhibition to faculty-industry panel.

498 Independent Study in Computer Science (1-3:0:0). *Prerequisite: 60 credit hours, major in computer science, and permission of instructor.* Research and analysis of selected problems or topics in computer science. Topic must be arranged with and instructor and approved by the department chair before registering. May be repeated for a maximum of 6 credits if the topics are substantially different.

499 Special Topics in Computer Science (3:3:0). *Prerequisite: 60 credit hours and permission of instructor; specific prerequisites vary with nature of topic.* Topics of special interest to undergraduates. May be repeated for a maximum of 6 credits if the topics are substantially different.

BS/Accelerated MS Degree Programs

These programs are for students interested in immediately continuing on to graduate studies in computer science:

BS/Accelerated MS in Computer Science
BS/Accelerated MS in Information Security and Assurance
BS/Accelerated MS in Information Systems
BS/Accelerated MS in Software Engineering

Admission Requirements: Students in the BS program in computer science can apply for a BS/Accelerated MS program if they have earned 90 undergraduate credits with an overall GPA of at least 3.50 for the MS in Computer Science degree program or an overall GPA of at least 3.30 for the remaining MS degree programs. Criteria for admission are identical to criteria for admission to the respective MS program.

Degree Requirements: Students must complete 144 credits that satisfy requirements for the BS program as well as those for the MS program, with 6 credits overlap. Students register for 6 credits of CS 500-level basic courses in place of the corresponding CS 400-level courses required for the undergraduate degree requirements. For example, students in the Accelerated MS CS must register for two of the following courses: CS 540, 550, 551, 555, 571, 580, and 583 in place of the corresponding 400-level courses. Students complete all MS requirements as specified by the respective MS degree program and apply the two courses from the respective approved list toward their MS requirements. Students are permitted to take additional graduate basic courses in their undergraduate programs. In such cases, those classes cannot be counted towards requirements for the MS.

Degree Conferral: Students must apply to have the BS degree conferred the semester before they expect to complete the BS requirements. At the completion of the MS requirements, a master's degree is granted.

Additional information about these and other accelerated degree options is available from the Computer Science Department office or at http://catalog.gmu.edu.

Revised June 6, 2014