1867
HOWARD
UNIVERSITY

Computer
Science
Program

Undergraduate Handbook

1016 Downing Hall
2300 Sixth St. NW
Washington, DC 20059
202-806-6585
www.eecs.cea.howard.edu

UNDERGRADUATE STUDY IN COMPUTER SCIENCE

Department of Electrical Engineering and Computer Science
Howard University
Washington, DC 20059

Revised June 2018

Table of Contents

OUR PROGRAM AND THE COMMUNITY

The University and Community

Howard University is a comprehensive, research-oriented, historically black private university providing an educational experience of exceptional quality to students of high academic potential. Further, the University is dedicated to attracting and sustaining a cadre of faculty who are, through their teaching and research, committed to the development of distinguished and compassionate graduates and to the quest for solutions to human and social problems in the United States and throughout the world. This mission of the University is central to everything we do and can be found in a 1989 resolution of the Board of Trustees. With its reputation for providing high-quality education at an affordable price, Howard is consistently ranked one of the nation's very best universities. Of the approximately 26,000 students enrolled, around 10,000 were graduate and professional students.

Washington DC (population 582,049) is the capital city of the United States of America. The city and the surrounding area offer many cultural advantages, including its well-known monuments and inspiring memorials, and an excellent array of theater and music.

Our Program

The Computer Science Program in the Department of Electrical Engineering and Computer Science at Howard is one of the first to be created in a Historically Black College and University (HBCU). The Program's primary mission is to expand and diversify the pool of qualified individuals in the computing profession and to advance knowledge in computer science by providing high quality instruction and conducting research that addresses technical challenges and societal problems.

The Computer Science Program will continue to be the program of choice for students seeking high-quality undergraduate and graduate degree programs in computer science. The program will be recognized across the nation and the global community for research and education that produces diverse and versatile professionals. Within a few years of graduating from the Computer Science Program, graduates will be able to attain the **Program Educational Objectives**. Graduates will be able to:

1. Utilize their knowledge, problem-solving, and communication skills in professional careers in government agencies and the private sector;
2. Conceptualize, analyze, design, implement, and evaluate a computational solution to a real-life problem using appropriate tools independently and as members or leaders of multidisciplinary teams;

3. Will have earned or will be pursuing graduate and professional degrees and/or professional development opportunities, including those required to pursue and maintain professional certifications.

The program offers a traditional B.S. degree in computer science, a computer science minor option for non-engineering disciplines at Howard, a graduate certificate course in cybersecurity, a traditional M.S. degree in computer science, and an accelerated 1-year M.S. degree in computer science, and a Ph.D. program in Computer Science. The B.S. program is accredited by the Computing Accreditation Commission of ABET, 111 Market Place, Suite 1050, Baltimore, MD 21202-4012 – telephone: (410) 347-7700. The Program Educational Objectives embrace outcomes-based learning. Students are prepared to achieve the following Student Outcomes:

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
d) An ability to function effectively on teams to accomplish a common goal
e) An understanding of professional, ethical, legal, security and social issues and responsibilities
f) An ability to communicate effectively with a range of audiences
g) An ability to analyze the local and global impact of computing on individuals, organizations, and society
h) Recognition of the need for and an ability to engage in continuing professional development
i) An ability to use current techniques, skills, and tools necessary for computing practice.
j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices; and
k) An ability to apply design and development principles in the construction of software systems of varying complexity.

Currently, our faculty includes 10 tenured and tenure-track faculty and 4 adjuncts faculty/lecturers. We also have a technical and administrative support staff person. Most of our undergraduate students are full time. Students contribute to nearly every aspect of the program’s operation. In addition to taking a wide variety of courses, they have the opportunity to participate in undergraduate research, and internships and co-op opportunities with our Industry Affiliates. The local chapter of the Association of Computing Machinery (ACM), and the Upsilon Pi Epsilon (UPE) honor society sponsors both professional and social events. Students are strongly encouraged to seek membership of these organizations; however, membership of UPE is by invitation only.

THE FACULTY AND THEIR RESEARCH

Our Faculty

For more details: http://www.eecs.cea.howard.edu/faculty

Peter Keiller, Associate Professor; D.Sc., George Washington University, 1996. Software engineering process, reliability engineering, software testing, software fault tolerance, statistical modeling and analysis, performance modeling.

Todd Shurn, Associate Professor; Ph.D., Southern Methodist University, 1994. Combinatorial optimization, heterogeneous data communication networks, web services and interoperability, interdisciplinary multimedia applications, gaming.

Harry Keeling, Associate Professor and Director, Computer Science Program; Ph.D., George Mason University, 1998. Intelligent tutoring systems, machine learning, artificial intelligence, intelligent agents, web-based technology, and expert systems.

Legand L. Burge III, Professor and Chair; Ph.D., Oklahoma State University, 1998. Parallel and distributed computing, operating systems, and computer networking.

Gedare Bloom, Assistant Professor; Ph.D., George Washington University, 2013. Embedded Systems, Computer Security, Cyber-physical systems, operating systems, computer architecture.

Danda Rawat, Assistant Professor; Ph.D., Old Dominion University, 2013. Cyber security, machine learning and wireless networking for emerging networked systems including cyber-physical systems

Moses Garuba, Professor and Director of Graduate Admissions; Ph.D., University of London, 2000. Information security, database security, secure electronic transactions, distributed algorithms, formal methods, computer forensics.

Jiang Li, Associate Professor; Ph.D., Rensselaer Polytechnic Institute, 2003. Computer networking, network security, network simulation, data communications, wireless/mobile networking, and sensor networks.

Gloria Washington, Assistant Professor; Ph.D., George Washington University, 2011. Human-Computer Interaction, Human-centered computing, affective computing, biometrics.

Chunmei Liu, Professor; Ph.D., University of Georgia, 2006. Bioinformatics, computational biology, algorithms, graph theory.

Adjunct Faculty

Reginald Hobbs, Assistant Professor; Ph.D., Georgia Institute of Technology, 2005. Software engineering, programming languages, knowledge engineering, knowledge management.

Bernard Woolfolk, Instructure, M.S., George Washington University, 1990. Object-oriented design, software engineering.

Emeritus

Don Coleman, Professor Emeritus; Ph.D., University of Michigan, 1971. Fault tolerant software, systems engineering, software engineering, software reliability, software metrics, simulation of parallel processes, user interfaces.

Ronald Leach, Professor Emeritus; Ph.D., University of Maryland, College Park, 1971. Software engineering, software reuse, software measurement and metrics, software fault-tolerance, software testing, operating systems, and computer science education.

Arthur Paul, Professor Emeritus; Ph.D., University of Virginia, 1983. Large-scale systems design, systems engineering, and technology transfer and commercialization.

John Trimble, Professor Emeritus; Ph.D., Georgia Institute of Technology, 1992. System dynamics, expert systems, software engineering, modeling and simulation, knowledge management, appropriate technology.

Wolsey Semple, Professor Emeritus; MS., Massachusettes Institute of Technology, 1961. Computer Graphics, Systems Operations.

Research Areas

Our faculty conducts research in a broad range of research areas. Following is a quick reference index to their interests.

Algorithms and Complexity Theory
Chunmei Liu

Artificial Intelligence
Harry Keeling, Reginald Hobbs

Bioinformatics and Computational Biology
Chunmei Liu, Legand Burge

Computer Architectures
Legand Burge, Jiang Li, Gedare Bloom

Cyber Security
Danda Rawat, Gedare Bloom

Databases and Data Visualization and Analysis
Moses Garuba, Peter Keiller

Data Communications and Networking
Jiang Li, Gedare Bloom, Legand Burge, Todd Shurn,

Distributed/Parallel Computation and Operating Systems
Legand Burge, Moses Garuba, Gedare Bloom, Jiang Li, Torrence Fennell

Information Assurance and Computer Security
Gloria Washington, Gedare Bloom, Jiang, Li, Moses Garuba, Legand Burge

Mobile Computing
Legand Burge, Gloria Washington, Jiang Li

Multimedia Systems, Gaming, and WWW Applications
Todd Shurn, Legand Burge, Gloria Washington

Human-Computer Interaction ,Human-Centered Computing
Gloria Washington, Jamika Burge

Performance Modeling and Simulation
Peter Keiller, Legand Burge, Gedare Bloom, Gloria Washington, Jiang Li

Software Engineering and Environments
Peter Keiller, Gloria Washington

Systems Engineering
Peter Keiller

Faculty Contact Information

Program Director

Harry Keeling, Associate Professor	hkeeling@howard.edu http://www.cs.ceacs.howard.edu/users/hkeeling	(202) 806-4830	1016 Downing Hall

Faculty:

Peter Keiller, Associate Professor	pk@scs.howard.edu http://www.cs.ceacs.howard.edu/users/pkeiller	(202) 806-4828	2112 Downing Hall
Jiang Li, Associate Professor	lij@scs.howard.edu http://www.cs.ceacs.howard.edu/users/jli	(202) 806-4861	2038B Downing Hall
Danda Rawat, Associate Professor	Danda.Rawat@howard.edu http://www.eece.cea.howard.edu/users/drawat	202-806-2209	2120B Downing Hall
Todd Shurn, Associate Professor	shurn@scs.howard.edu http://www.cs.ceacs.howard.edu/users/tshurn	(202) 806-4824	1110 Downing Hall
Legand Burge, Professor	blegand@scs.howard.edu http://www.cs.ceacs.howard.edu/users/blegand	(202) 806-4852	2040 Downing Hall
Moses Garuba, Professor	moses@scs.howard.edu http://www.cs.ceacs.howard.edu/users/mgaruba	(202) 806-4371	B36B Mackey
Gloria Washington, Assistant Professor	gwashington@scs.howard.edu http://www.cs.ceacs.howard.edu/users/gwashington	(202) 806-7417	1110 Downing Hall

Chunmei Liu, Professor	chunmei@scs.howard.edu http://www.cs.ceacs.howard.edu/users/cliu	(202) 865-0056	2038A Downing Hall

Overview of Degree Programs and Requirements

The Computer Science Program provides an undergraduate program leading to the award of the Bachelor of Science in Computer Science, and a computer science minor option for non-engineering disciplines at Howard University, and instruction and research leading to the Master's degree in Computer Science.

About Computer Science

Computer Science, generally defined, is the study of problem-solving procedures, computability and computation systems. Computer engineers and computer science professionals are proficient in several programming languages and are familiar with advanced mathematical concepts in subjects such as linear algebra, matrix theory, topology and Boolean algebra. Often, they work with individuals from other disciplines to coordinate new developments in hardware and software. Computer utilization has made the solutions to complex problems, once considered intractable, feasible because of the speed, accuracy and versatility of the modern computer.

Undergraduate Program

The Program offers a program of study geared to students who wish to pursue careers in the emerging field of software engineering. The undergraduate curriculum provides students with a comprehensive knowledge of the theory, design and application of digital computers, information processing technologies and systems engineering. The program is accredited by the Computing Accreditation Commission of ABET, 111 Market Place, Suite 1050, Baltimore, MD 21202-4012 – telephone: (410) 347-7700.

There is an emphasis on the engineering of computer software, as well as engineering with computers, with particular stress on software and the software/hardware interface. The first two years of instruction prepare students in the engineering fundamentals, while the last two years of instruction focus on systems engineering and computer science and a concentration area of their own interest. In addition to formal course work, students are encouraged to serve an informal internship of at least one summer in a computer-oriented laboratory within the University or at another computing facility.

Students seeking the Bachelor of Science degree in Computer Science must complete a minimum of 120 credit hours including core courses in Computer Science, and Liberal Arts. Elective courses in Computer Science, Mathematics, Chemistry, Biology, Physics, Computer Engineering, Electrical Engineering, Humanities/Social Science, and African American Studies are also required. The curriculum breakdown is listed below.

Concentration	Credit Hours
Computer Science Core	53
Technical Electives	12
Liberal Arts Core Curriculum	40
Social Sciences and Humanities Elective Courses	12
African American Cluster Core	3
Total	**120**

Computer Science Core (53 credit hours)

CSCI-100 Introduction to Computer Science, 3
CSCI-135 Computer Science I, 4
CSCI-136 Computer Science II, 3
CSCI-354 Computer Science III, 3
CSCI-201 Computer Organization I, 3
CSCI-202 Computer Organization II , 3
CSCI-211 UNIX Lab, 1
CSCI-363 Large Scale Programming, 3
CSCI-341 Theory of Computation, 3
CSCI-470 Fundamentals of Algorithms, 3
CSCI-375 Software Engineering, 3
CSCI-350 Structures of Programming Languages, 3
CSCI-432 Database Systems, 3
CSCI-376 Operations Research, 3
CSCI-401 Operating Systems, 3
CSCI-410 Modeling and Simulation, 3
CSCI-472 Systems Management Analysis, 3
CSCI-491 Senior Design Project I, 2
CSCI-492 Senior Design Project II, 2

University Requirements

The University-wide core curriculum course requirements are met by the curriculum. The core themes and Computer Science courses that satisfy the requirements are listed in the following table.

Core Theme	Computer Science
Intellectual Openness and Cultural Diversity	Introduction to Engineering EGPP-101 Introduction to Computer Sci. CSCI-100

Historical Awareness	African American Elective Humanities/Social Science Elective
Empirical Analysis	CSCI-491 Senior Design Project I CSCI-492 Senior Design Project II
Quantitative Literacy & Statistical Reasoning	MATH-189 Probability & Statistics (3)
Social and Human Relations	Humanities/Social Science Elective
Other Core Experiences	University Events – lectures, convocation, College/Department/Program Lecture series, CEACS Student Leadership Institute, Exhibitions, Study/travel Program, Middle & High School Education Programs, etc.

In addition to formal course work, students are encouraged to serve an informal internship of at least one summer in a computer-oriented laboratory within the University or at other computing facilities when deemed appropriate by their advisor. Students are eligible for internships after completion of at least three semesters of course work or in special instances upon recommendation of the advisor.

Bachelor’s Degree Requirements

Bachelor of Science in Computer Science

College of Engineering and Architecture

Howard University

(Effective for Students Entering in Fall 2012)

Date	Number	Course Title	Cr	Grade	Year	Date	Number	Course Title	Cr	Grade
					Freshmen					
	EGPP-101	Intro to Engineering	2				CSCI-135	Computer Science I	4	
		Non- Technical Elective	3				CSCI-211	UNIX Lab	1	
		Science Lec A	4				MATH-156	Calculus I	4	
		Science Lab A	0							
	ENGL-002	Freshmen Composition	3				ENGL-003	Freshmen Composition	3	
	CSCI-100	Intro to CS	2				SLMC 101	Principles of Speech	3	
			14						15	
					Sophomore					
	CSCI-136	Computer Science II	3				CSCI-354	Computer Science III	3	
	CSCI-201	Computer Organization I	3				CSCI-363	Large Scale Prog.	3	
	MATH-157	Calculus II	4				CSCI-202	Computer Organization II	3	
	MATH-181	Discrete Structures	3				MATH-180	Intro to Linear Algebra	3	
		Science Lec B (1)	4					Science Lec B (2)	4	
		Science Lab B (1)	0					Science Lab B (2)	0	
			17	-					16	-
					Junior					
	CSCI-341	Theory of Computation	3				CSCI-350	Structure of Programming Languages	3	
	CSCI-375	Software Engineering	3				CSCI-376	Operations Research	3	
	CSCI-470	Fundamentals of Alg.	3				CSCI-432	Database Systems	3	
	MATH-189	Probability and Statistics I	3				ENGL-009	Technical Writing	3	
		Technical Elective	3					Technical Elective	3	
			15						15	
					Senior					
	CSCI-401	Operating Systems	3				CSCI-472	Systems Mgt. Analysis	3	
	CSCI-410	Modeling and Simulation	3				CSCI-492	Senior Project II	2	
	CSCI-491	Senior Project I	2					Technical Elective	3	
		Non-Technical Elective	3					Non- Technical Elective	3	
		Technical Elective	3					Non- Technical Elective	3	
			14						14	

Bachelor of Science in Computer Science

College of Engineering and Architecture

Howard University

(Effective for Students Entering in Fall 2015)

Date	Number	Course Title	Cr	Grade	Year	Date	Number	Course Title	Cr	Grade
					Freshmen					
	EGPP-101	Intro to Engineering	2				CSCI-135	Computer Science I	4	
		Non- Technical Elective	3				CSCI-211	UNIX Lab	1	
		Science Lec A	4				MATH-156	Calculus I	4	
		Science Lab A	0							
	ENGW		3				ENGW		3	
	CSCI-100	Intro to CS	2				SLMC 101	Principles of Speech	3	
			14						15	
					Sophomore					
	CSCI-136	Computer Science II	3				CSCI-354	Computer Science III	3	
	CSCI-201	Computer Organization I	3				CSCI-363	Large Scale Prog.	3	
	MATH-157	Calculus II	4				CSCI-202	Computer Organization II	3	
	MATH-181	Discrete Structures	3				MATH-180	Intro to Linear Algebra	3	
		Science Lec B (1)	4					Science Lec B (2)	4	
		Science Lab B (1)	0					Science Lab B (2)	0	
			17	-					16	-
					Junior					
	CSCI-341	Theory of Computation	3				CSCI-350	Structure of Programming Languages	3	
	CSCI-375	Software Engineering	3				CSCI-376	Operations Research	3	
	CSCI-470	Fundamentals of Alg.	3				CSCI-432	Database Systems	3	
	MATH-189	Probability and Statistics I	3				ENGL-009	Technical Writing	3	
		Technical Elective	3					Technical Elective	3	
			15						15	
					Senior					
	CSCI-401	Operating Systems	3				CSCI-472	Systems Mgt. Analysis	3	
	CSCI-410	Modeling and Simulation	3				CSCI-492	Senior Project II	2	
	CSCI-491	Senior Project I	2					Technical Elective	3	
		Non-Technical Elective	3					Non- Technical Elective	3	
		Technical Elective	3					Non- Technical Elective	3	
			14						14	

Bachelor of Science in Computer Science

College of Engineering and Architecture

Howard University

(Effective for Students Entering in Fall 2016)

Date	Number	Course Title	Cr	Grade	Year	Date	Number	Course Title	Cr	Grade
					Freshmen					
	EGPP-101	Intro to Engineering	2				CSCI-135	Computer Science I	4	
		Non- Technical Elective	3				CSCI-211	UNIX Lab	1	
		Science Lec A	4				MATH-156	Calculus I	4	
		Science Lab A	0				ENGW	English II	3	
	ENGW	English I	3				SLMC 101	Principles of Speech	3	
	CSCI-100	Intro to CS	3							
			15						15	
					Sophomore					
	CSCI-136	Computer Science II	3				CSCI-354	Computer Science III	3	
	CSCI-201	Computer Organization I	3				CSCI-363	Large Scale Prog.	3	
	MATH-157	Calculus II	4				CSCI-202	Computer Organization II	3	
		Science Lec B (1)	4				MATH-181	Discrete Structures	3	
		Science Lab B (1)	0					Science Lec B (2)	4	
								Science Lab B (2)	0	
			14	-					16	-
					Junior					
	CSCI-341	Theory of Computation	3				CSCI-350	Structure of Programming Languages	3	
	CSCI-401	Operating Systems	3					Technical Elective	3	
	CSCI-470	Fundamentals of Alg.	3				CSCI-432	Database Systems	3	
	CSCI-450	Data Communications and Network Programming	3				ENGL-009	Technical Writing	3	
	CSCI-453	Intro to Cybersecurity 1	3				MATH-180	Intro to Linear Algebra	3	
			15						15	
					Senior					
	CSCI-491	Senior Project I	3					Technical Elective	3	
	CSCI-375	Software Engineering	3				CSCI-492	Senior Project II	2	
	CSCI-473	Applied Data Science	4					Technical Elective	3	
		Technical Elective	3					Non- Technical Elective	3	
		Technical Elective	3					Non- Technical Elective	3	
			16						14	

Computer Science Option for Non-Engineering Majors

The Computer Science option for non-engineering students is intended for use by departments that use it as a minor. Students must take the following three courses:

CSCI 100	Intro to Computer Science	3
CSCI 135	Computer Science I	4
CSCI 136	Computer Science II	3

At least two additional courses (more if required by the student’s own departments) must be taken from the following list, or courses from the department approved by the chair and course instructor:

CSCI 201	Computer Organization I	3
CSCI 202	Computer Organization II	3
CSCI-263	Web Makers	3
CSCI 165	Scientific Computing for Eng.	3
EECE 211	Intro to Digital Systems	3
CSCI 432	Database Systems	3
MATH 181	Discrete Structures	3
CSCI 354	Computer Science III	3
CSCI 363	Large Scale Programming	3
CSCI-421	Computer and Video Game Dev	3
CSCI-402	Mobile Application Development	3
CSCI-410	Modeling and Simulation	3
CSCI-453	Introduction to Cybersecurity I	3

Any other selection must be approved in advance by the chair of the department.

Intro to Digital Systems (EECE-211) should not be taken until the student has completed Computer Science II (CSCI 136) and Computer Organization I (CSCI 201).

Discrete Structures (MATH 181) should not be taken until the student has completed Computer Science II (CSCI 136). Calculus is also a prerequisite for MATH 181. Minors based on this option may choose to count a calculus course taken from the department of Mathematics as part of the minor.

Large Scale Programming (CSCI-363), Computer and Video Game Development (CSCI-421), Mobile Application Development (CSCI-402) should not be taken until the student has completed Computer Science III (CSCI 354).

Prerequisite Structure for Core Computer Science Courses

For simplicity, only the most direct prerequisite is shown both a table and flow chart below.

Course	Prerequisite(s)
CSCI 135 Computer Science I	'C' or better in **CSCI 100** Intro. To Computer Science
CSCI 136 Computer Science II	'C' or better in **CSCI 135** Computer Science I
CSCI 354 Computer Science III	'C' or better in **CSCI 136** Computer Science II
CSCI 201 Computer Organization I	'C' or better in **CSCI 135** Computer Science I
CSCI 202 Computer Organization II	**CSCI 201** Computer Organization I
CSCI 363 Large Scale Programming	'C' or better in **CSCI 13**6 Computer Science II
CSCI 375 Software Engineering	**CSCI 354** Computer Science III
CSCI 376 Operations Research	**CSCI 375** Software Engineering
CSCI 350 Structure of Programming Languages	**CSCI 20**1 Computer Organization I and **CSCI 341** Theory of Computation
CSCI 472 Systems Management Analysis	**CSCI 375** Software Engineering
CSCI 410 Modeling and Simulation	
CSCI 341 Theory of Computation	**CSCI 136** Computer Science II and **MATH 181** Discrete Structures
CSCI 432 Database Systems	**CSCI 136** Computer Science II
CSCI 401 Operating Systems	**CSCI 202** Computer Organization II and **CSCI 363** Large Scale Programming
CSCI 470 Fundamentals of Algorithms	**CSCI 354** Computer Science III
CSCI 491 Senior Project I	
CSCI 492 Senior Project II	**CSCI 491** Senior Project I

Prerequisite Structure for Non-Core Computer Science Courses

For simplicity, only the most direct prerequisite is shown in table below.

Course	Prerequisite(s)
CSCI 165 Scientific Computing for Engineers	

CSCI 203 Object-Oriented Programming using Java	CSCI 136 Computer Science II
CSCI 364 Web Services	
CSCI 379 Introduction to Human Computer Interaction	CSCI 354 Computer Science III
CSCI 390 Ethical and Social Impact of Computing	
CSCI 391 Patents and Technology Entrepreneurship	Junior Standing
CSCI 402 Mobile Application Development	CSCI 363 Large Scale Programming
CSCI 421 Computer and Video Game Development	CSCI 354 Computer Science III, Software Engineering
CSCI 422 Game Engine Programming	CSCI 421 Computer and Video Game Development
CSCI 440 Object-Oriented Programming	CSCI 354 Computer Science III
CSCI 450 Data Communications and Network Programming	CSCI 354 Computer Science III, CSCI 201 Computer Organization I, CSCI 470 Fundamentals of Algorithms, MATH 181 Discrete Structures
CSCI 451 Applied Wireless Networking	CSCI 450 Data communications
CSCI 453 Intro to Cybersecurity I	Junior Standing
CSCI 454 Intro to Cybersecurity II	CSCI 453 Intro to Cybersecurity I
CSCI 460 Advanced Systems Administration	CSCI 211 Unix Lab
CSCI 474 Computational Biology	CSCI 470: Fundamentals of Algorithms, MATH 189: Probability and Statistics
CSCI 475 Introduction to Machine Learning	CSCI 354: Computer Science III, MATH 189: Probability and Statistics
CSCI 476 Intro to Artificial Intelligence	Pre-req: CSCI 350 Structures of Programming Languages.
CSCI 478 Engineering Economic System Design	CSCI 376 Operations Research
CSCI 493 Lean Launch Pad: Startups	Junior Standing
CSCI 480 Digital Media and Multimedia Applications	Junior Standing
CSCI 498 Special Topics: Robotics Programming	Junior Standing

Undergraduate Courses

EGPP-101 Introduction to Engineering 2 Credits

Provides information on engineering education, the engineering profession, engineering basic concepts and engineering tools. Introduces the engineering design process and provides the opportunity for students to complete engineering design projects.

CSCI-100 Introduction to Computer Science 3 Credits

This course introduces the fundamentals of computer science. A brief examination of critical, creative, scientific thinking, is followed by more details on computational thinking. The focus is hands on computing exercises and group exercises that stress the importance of algorithms and problem solving. Students are exposed to the research agenda of the program, the different track options and career opportunities along the different tracks.

CSCI-140 Programming Team 1 Credit

This course is for participants of the programming team.

CSCI-135 Computer Science I 4 Credits

This course introduces the discipline of computer programming. Closed laboratory to reinforce lecture topics and introduce new topics. Course is designed to expose students to basic programming concepts and to the use of the C++ language. This course is designed to enhance the student's ability to design, develop and test/debug programs. Each student will increase his or her skill in writing correct and maintainable programs. Emphasis will be placed on problem analysis and on the subsequent development of algorithms. Several standard data types will be discussed and the student will gain an understanding of the issues relating to the use, design and implementation of each type in C++. A major focus of the lectures will be to provide an overview of real-world problem-solving concepts and top-down software design. Prerequisite: Intro. to Computer Science (with a grade of 'C' or better).

CSCI-136 Computer Science II 3 Credits

Course exposes students to the software development life cycle with a focus on the concepts and use of the object-oriented paradigm in problem analysis, solution design, software development and implementation. This course is designed to enhance the student's ability to engineer software that is efficient, maintainable and cost efficient over its entire life cycle. Data abstraction is discussed in depth and students gain experience in the use of classes, object and member functions. Students gain an understanding of the development of reusable abstract data types. Software reuse is emphasized and object-oriented concepts are used throughout the course. O-notation and the complexity of algorithms are discussed at relevant points in the course. Prerequisite: Computer Science I (with a grade of 'C' or better).

CSCI-165 Scientific Computing for Engineers 3 Credits

Scientific computing consists of computing using computers to analyze and solve scientific problems which usually concerned with constructing mathematical models and quantitative analysis techniques. Scientists and engineers develop computer programs and software that model systems and run these programs with various sets of input. Typically, these models require intensive computing and are often executed on supercomputers or distributed computing platforms. In this course three software/programming tools (Excel, Maxima, and C) are studied to investigate how scientific problems are solved in their appropriate domain. All of the three tools would be running on a PC platform.

CSCI-354 Computer Science III 3 Credits

The course continues the study of data structures and algorithms, focusing on algorithmic design and problem analysis and the relationships between data representation, algorithm design, and program efficiency. Topics include advanced data structures, key algorithm design techniques, analysis of the time and space requirements of algorithms, and the subsequent development of solution of systems. Concrete examples will be drawn from a variety of domains, such as algorithms for trees and graphs, indexing and search, and real-world problems. Prerequisite: Computer Science II (with a grade of ‘C’ or better).

CSCI-201 Computer Organization I 3 Credits

This course will cover the fundamentals required to understand the relationship between computer hardware and software. Topics include data representation on computers, computer arithmetic, Boolean algebra and digital logic, and assembly programming in MIPS. Prerequisite: Computer Science I (with a grade of ‘C’ or better).

CSCI-202 Computer Organization II 3 Credits

This course will present the relationship between computer hardware and software, and the fundamental knowledge essential for understanding and designing the operations of computer systems. Topics include performance evaluation, non-pipelined and pipelined datapath, memory hierarchies, and I/O devices. Prerequisite: Computer Organization I.

CSCI 211 UNIX Lab 1 Credit

This course will present the basic concepts of LINUX and UNIX operating systems. Topics that will be examined include Vi editor, Linux Command, directories, Disks and File systems, Users and Groups, File Permissions, Processes, file compression, basic network use, manage files, create and modify files, and Shell script.

CSCI-203 Object-Oriented Programming using Java 1 Credit

This course introduces Java programming and object-oriented programming concepts for students with previous programming experience in C/C++. The course provides a comprehensive overview of basic programming concepts in the Java programming language using an object-oriented approach. Prerequisite: Computer Science I.

CSCI-350 Structure of Programming Languages 3 Credits

The course will teach students the basic components of the design and analysis of computer programming languages as well as the fundamental computation theory that is required to understand those concepts. The course will also cover several non-imperative languages (unlike C, such as LISP and Prolog) to expose students to the diversity of programming languages. Prerequisite: Computer Organization I, Theory of Computation.

CSCI-341 Theory of Computation 3 Credits

Introduction to the classical theory of computer science. A study of the formal relationships between machines, languages and grammars; we will cover regular, context-free, context-sensitive, recursive and recursive enumerable languages. Sequential machines and their applications to devices, processes, and programming. Models of computation: finite state automata, push down automata, Turing machines. The role of non-determinism. Prerequisite: Computer Science II and Discrete Structures.

CSCI-363 Large Scale Programming 3 Credits

This course will introduce students to the Java Programming language, and to applications and systems in the large scale. Students will be introduced to the object-oriented method to software design using UML and will apply the object-oriented design/analysis techniques of UML to a realistic Java application. Students will gain familiarity with managing larger projects and OOA/D. Prerequisite: Computer Science II.

CSCI-375 Software Engineering 3 Credits

This course will introduce students to the basic concepts of software engineering and the software development life cycle. The course will cover methodological techniques for software specification, design, implementation, testing, verification, and documentation. The course will also present the use of state-of-the-art tools for computer-aided software engineering (CASE). Prerequisite: Computer Science III (CSCI-354).

CSCI-376 Operations Research 3 Credits

Methodology for planning, analyzing and evaluating optimal systems: identifying and structuring objectives and defining performance requirements that influence the design of the system. Synthesizing and analyzing alternative solutions and applying optimization techniques for the optimum queuing system. Applications to real world systems with open and closed queues with emphasis on computer systems using microcomputer software packages. Prerequisite: CSCI-375 Systems Engineering I.

CSCI-379 Introduction to Human Computer Interaction 3 Credits

Students will learn the fundamental concepts of human-computer interaction and user-centered design thinking, through working in teams on an interaction design project, supported by lectures, readings, and discussions. They will learn to evaluate and design usable and appropriate software based on psychological, social, and technical analysis. They will become familiar with the variety of design and evaluation methods used in interaction design, and will get experience with these methods in their project. Prereq: Computer Science III

CSCI-390 Ethical and Social Impact of Computing 3 Credits

This course will present the foundations of ethics in the context of computing. The broader social impact of computing and technology in general will also be reviewed. Areas of specific focus will include technology and human values, costs and benefits of technology, the character of technological change, and the social context of work in computer science and information technology.

CSCI-401 Operating Systems 3 Credits

This course will present the basic concepts of operating systems. Topics that will be examined include processes and interprocess communication/synchronization, virtual memory, program loading and linking system calls and system programs; interrupt handling, device and memory management, process scheduling, deadlock and the trade-offs in the design of large-scale multitasking operating systems. Prerequisite: Computer Organization II, and Large-Scale Programming.

CSCI-410 Modeling and Simulation 3 Credits

Introduces the fundamentals of system design and modeling. Emphasizes advantages and limitations of various modeling techniques for different applications. Introduces probability distributions typical of queuing models and presents in-depth discussions and experiments with existing simulation packages.

CSCI-450 Data Communications and Network Programming 3 Credits

This is an introductory course on computer networking. It will cover the layering model of the Internet. The upper four layers (application, transport, network and data link) will be discussed in detail with dominant networking protocols and algorithms introduced. Students will also learn how to do basic programming on the Internet. Prereq: Computer Science III, Computer Organization I, Fundamentals of Algorithms, Discrete Structures

CSCI-432 Database Systems 3 Credits

This course will present the basic concepts of database systems. Topics that will be covered include basic relational database theory, relational database modeling, relational database design and implementation, normalization, transaction management, the SQL language and other languages and facilities provided by database management systems. Prerequisite: Computer Science III.

CSCI-472 Systems Management Analysis 3 Credits

This course presents methodology for large-scale system design and analysis using modern semantic analysis techniques. Identification and definition of large-scale (community/industrial-based) problems. Discusses how to select and quantify measures of the severity of the problem. Presents different techniques for modeling alternative solutions to problems. Prerequisite: Software Engineering.

CSCI-478 Engineering Economic System Design⬩ 3 Credits

Presents methodology for system design. Methodology begins with identification and definition of private sector problems to which solutions are justified by economics. Discusses selection of appropriate economic measures for comparing alternative solutions such as present worth, equivalent annual cost, cost/benefit ratio, life cycle cost, return on investment payback period. Presents different techniques for modeling alternative solutions to the problems and predicting cost. Other topics discussed include decision-making, system implementation, operations and retirement. Prerequisite: Operations Research.

CSCI-491 Senior Project I 2 Credits

Allows the senior student the opportunity to demonstrate his or her knowledge of computer science principles by application to a class project of his or her choosing, with the guidance and supervision of a faculty member. The student develops a proposal for the project, followed by an architectural design and detailed design, all of which must be presented in class. Prerequisite: Computer Organization II.

CSCI-492 Senior Project II 2 Credits

In part two, the senior student develops and implements the system solution to the proposed project. The system, most commonly comprising computer software, hardware, procedures, etc., is implemented and tested in the program's Systems Development Laboratory. The student is required to demonstrate the system solution to the faculty and the student body of the program. Prerequisite: Senior Project I.

CSCI-451 Applied Wireless Networking⬩ 3 Credits

From both the conceptual and practical standpoints, this course will present the basics of wireless networking. Topics that will be examined include the connection between wireless networks and the Internet, radio signal transmission fundamentals, wireless LAN/WAN industrial stands, and wireless network administration such as network design, installation, configuration, maintenance and trouble shooting. Prerequisite: Data Communications

CSCI-453 Introduction to Cybersecurity I 3 Credits
Computer Security Overview. Malware and Cyberwarfare. Passwords. Biometrics. Access Controls. Multilevel Security. Multilateral Security. Firewalls. Intrusion Detection. Cryptography Before 1970. Symmetric Key Cryptography. The Data Encryption Standard (DES). The Public Key Paradigm. Knapsacks. The RSA Approach to Public Key Cryptology. Elliptic Curve Cryptography. The Advanced Encryption Standard (Rijndael). Hash Functions. The Digital Signature Standard.

CSCI-454 Introduction to Cybersecurity II 3 Credits
Distributed Denial of Service (DDoS). Hash Functions. SHA-3 and Keccak. Network Security. Network attack and defense. Steganography, Software security I. Software flaws. Malware. Miscellaneous software based attacks. Software security II. Software reverse engineering. Software tamper resistance. Digital rights management. Software development. Stuxnet and Cyberwarfare. Advanced encryption techniques. Elliptic Curves and Elliptic Curve Cryptography. Electronic voting. Quantum cryptography. Prerequisite: Intro to Cybersecurity I

CSCI-421 Computer and Video Game Development 3 Credits
The course will span the software domains embedded in computer and video games. Topics such as game computational infrastructure, design, engines, and motion will be presented through discussion and assignments. Game industry guest speakers will discuss software challenges and opportunities. Students completing this course will understand the software development process required to create a successful game and possess the programming expertise to create a simple game. Prerequisite: Computer Science III, Software Engineering.

CSCI-422 Game Engine Programming 3 Credits
Game engine programming is introduced as a critical element in compelling game creation. Programming activity will feature input capture, world integration, object motion, collision detection and audio scoring. Game performance metrics, code optimization and quality assurance testing procedures will be emphasized. Code examples will be presented from XNA game studio and Torque. Course game project may be completed using a 2D or 3D game engine of choice including Torque, Gamestudio, Panda3D, or OGRE 3D rendering engine. Prerequisite: Computer and Video Game Development.

CSCI-402 Mobile Application Development 3 Credits
This course will introduce students to developing applications which target mobile devices. Students will be introduced to many issues unique to mobile applications, including synchronization, remote data access, security and sometimes-connected networks. They will research topics in these areas and develop a significant project which demonstrates their knowledge and understanding of these issues. Prerequisite: Large Scale Programming.

CSCI-480 Digital Media and Multimedia Applications⬩ 3 Credits

This course provides an introduction to digital media fundamentals including audio, video formats, storage and delivery. Windows Media and other technology will be extensively utilized as a method for digital content manipulation, rights management and internet transfer. Students will be exposed to basic internet architecture, operations and useful world wide web (WWW) resources. In addition, a practical understanding of digital computational devices, communication ports and connection cables will be acquired. Prerequisite: Junior Standing.

CSCI-364 Web Services⬩ 3 Credits

Presents topics in distributed computing with particular emphasis on Web Services using Microsoft .NET Framework. Also, discussion on layered protocols, the client-server model, remote procedure call. Students program extensively in C# and Visual Basic .NET. Prerequisite: 306-401 Operating Systems.

CSCI-460 Advanced Systems Administration⬩ 3 Credits

Advanced system administration course provides a strong practical experience to Linux and Solaris operating systems. The course includes topics such as Samba (Windows file and print sharing), Email, Web serving with Apache, remote access, networking setup, Internet proxy services, fire wall and security administration, deploy LDAP in a Linux, Solaris and windows environment and also compile, configure and patch a Kernel module. Prerequisite: Unix Lab

CSCI-470 Fundamentals of Algorithms 3 Credits

Techniques for designing efficient algorithms, analyzing their complexity and applying these algorithms to a broad range of application settings. Methods for recognizing and dealing with hard problems are studied. Prerequisite: CSCI 354: Computer Science III.

CSCI-474 Computational Biology 3 Credits

Introduces computational methods for understanding biological systems at the molecular level. Problem areas such as mapping and sequencing, sequence analysis, structure prediction, phylogenic inference, regulatory analysis. Techniques such as dynamic programming, Markov models, expectation-maximization, local search. Prerequisite: CSCI 470: Fundamentals of Algorithms, MATH 189: Probability and Statistics

CSCI-475 Introduction to Machine Learning 3 Credits

Techniques for learning from data and applying these algorithms to application settings. Topics covered include Bayesian methods, linear classifiers such as the perceptron, regression, and non-parametric methods such as k-nearest neighbors. Prerequisite: CSCI 354: Computer Science III, MATH 189: Probability and Statistics

⬩

CSCI-476 Introduction to Artificial Intelligence 3 Credits

This course will introduce students to contemporary topics in artificial intelligence. Topics that will be examined include basic AI concepts, representations, and techniques used in building practical computational systems (agents) that appear to display artificial intelligence, through the use of adaptive information processing algorithms. During the semester students will learn general knowledge representation techniques and problem-solving strategies. Topics will include search, intelligent agents, game playing and rule-based systems. Prerequisite: CSCI 350 Structures of Programming Languages.

EGPP-493 Bison Startup: Technology Entrepreneurship and Lean Startups 3 Credits

This course provides real world, hands-on learning on what it's like to start a high-tech company. This class is not about how to write a business plan. It's not an exercise on how smart you are in a classroom, or how well you use the research library to size markets. And the end result is not a Power Point slide deck for a VC presentation. And it is most definitely not an incubator where you come to build the—hot-idea that you have in mind. This is a practical class–essentially a lab, not a theory or—book class. Our goal, within the constraints of a class room and a limited amount of time, is to create an entrepreneurial experience for you with all of the pressures and demands of the real world in an early stage startup. You will be getting your hands dirty talking to customers, partners, competitors, as you encounter the chaos and uncertainty of how a startup works. You'll work in teams learning how to turn a great idea into a great company. You'll learn how to use a business model to brainstorm each part of a company and customer development to get out of the classroom to see whether anyone other than you would want/use your product. Finally, based on the customer and market feedback you gathered, you would use agile development to rapidly iterate your product to build something customers would actually use and buy. Each day will be new adventure outside the classroom as you test each part of your business model and then share the hard-earned knowledge with the rest of the class. Junior Standing and CEACS major.

EGPP-494 Bison Accelerate: Launch and Iterate 3 Credits

This course is an immersive experience for students serious about launching a technology startup. Students will learn from invited guests and online material on specific activities that must take place to have a successful startup, and execute them through a series of sprints to launch their own technology venture. Student teams will enter the build phase as quickly as possible and develop a Minimal Viable Product (MVP) that will allow them to test clear hypothesis they have about their product/service. Students will measure the impact of their product/service into the marketplace using actionable metrics to analyze customer behavior. Student teams will learn whether their original assumptions about the product/service, process, and customer needs were correct, or whether they need to change strategies and iterate their MVP.

CSCI-498 Special Topics: Robotics Programming 3 Credits

The primary focus of this course will be behavior-based robotics, which uses semi-autonomous artificial intelligence modules for planning. Behavior-based robots use sensor information to react to changes in an environment, instead of complicated internal models. Higher level concepts that will be covered include multi-robot communication, robot localization and path planning. Prerequisite: Junior Standing, MEEG, ECEG, CSCI major.

CS Concentration Tracks

The Program offers students the ability to take prescribed technical electives and science elective courses geared toward their specific interest in the following tracks:

A. Computer Science
B. Gaming
C. Computer Networking
D. Cyber Security
E. Computational Biology
F. Computational Chemistry
G. Computational Mathematics
H. Computational Physics

By default, all students entering into the program are under the Computer Science track.

Science Requirements

Three courses based on the concentration track selected.

<Science A > --1/2 year of any laboratory science course listed must be taken:
<Science B (I) > and <Science B (II) > -- 1 full year of a different laboratory science course than <Science A> must be taken.

<u>Science courses include:</u>

1. BIOL 101 Biology I
2. BIOL 102 Biology II
3. CHEM 003 Chemistry I, and CHEM 005 Chemistry Lab
4. CHEM 004 Chemistry II, and CHEM 006 Chemistry Lab II
5. PHYS 013 Physics for Engineers I, and PHYS 023 Physics for Engineers Lab I
6. PHYS 014 Physics for Engineers II, and PHYS 024 Physics for Engineers Lab II

<u>Science Courses By Track:</u>

A. Computational Biology
   1. Can take any science course not from the Department of Biology (Science A).
   2. BIOL 101 General Biology I (Science B I)
   3. BIOL 102 General Biology II (Science B II)

B. Computational Chemistry
   1. Can take any science course not from the Department of Chemistry (Science A).
   2. CHEM 003 General Chemistry I, CHEM 005 General Chemistry Lab I (Science B I)
   3. CHEM 004 General Chemistry II, CHEM 006 General Chemistry Lab II (Science B II)

C. Gaming
    1. Can take any science course from the (Science A)
    2. Can take any science course from the list different from Science A (Science B I and II).

D. Computational Mathematics
    1. Can take any science course from the (Science A)
    2. Can take any science course from the list different from Science A (Science B I and II).

E. Computer Networking
    1. Can take any science course from the (Science A)
    2. Can take any science course from the list different from Science A (Science B I and II).

F. Computational Physics
    1. Can take any science course not from the Department of Physics & Atmospheric Science (Science A).
    2. PHYS 013 Physics for Science & Engineers I, PHYS 023 Physics for Science & Engineers Lab I (Science B I)
    3. PHYS 014 Physics for Science & Engineers II, PHYS 024 Physics for Science & Engineers Lab II (Science B II)

G. Computer Security
    1. Can take any science course from the (Science A)
    2. Can take any science course from the list different from Science A (Science B I and II).

H. Computer Science
    1. Can take any science course from the (Science A)
    2. Can take any science course from the list different from Science A (Science B I and II).

NOTE: CHEM003/004 and CHEM 005/006 will increase your total credits to 122.

Non-Technical Electives

Four courses, each of at least 3 credits, are required. Among these courses must be:

- One from the University's African-American Cluster

ENGL 054/055	African-American Literature
POLS 006	Pan-Africanism
HIST 005/006	Introduction to Black Diaspora
AFST 101	African World: Intro. To Contemporary Africa
AFRO 005/006	Afro-American Studies
MUTP 100	Blacks in the Arts
FASH 102	Perspectives on African-American Dress

ARTH 193 Black Body Dress and Culture

- Any course from AFRO, or AFST will classify as an African-American Cluster
- Any course from the Divisional Studies A, B, C, and D lists of the College of Arts and Sciences
- Any foreign language course at "Level II" or higher, or a "Level I" course provided that a more advanced course in the same language is included
- Any other course recommended in writing by the student's academic advisor and approved by the program director

.

Technical Electives

The objective of this requirement is to expand the student's contact with advanced, "state-of-the-art" topics in the major field geared toward their own interest. Students are prescribed technical electives based on their SCS Concentration track selected. Unless a student selects a track, by default all students are in the (I) Computer Science track.

A. Computational Biology
   1. BIOL 200 Genetics
   2. BIOL 320 Molecular Biology
   3. CSCI 474 Computational Biology
   4. Any technical elective approved for the Computer Science track

B. Computational Chemistry
   1. CHEM 001 General and Applied Chemistry
   2. CHEM 184 Methods in Computational Chemistry
   3. MATH 159 Differential Equations or MATH 164 Introduction to Numerical Analysis
   4. Any technical elective approved for the Computer Science track

C. Gaming
   1. CSCI 421 Computer and Video Game Development

    2. CSCI 422 Game Engine Programming
    3. Any technical elective approved for the Computer Science track

D. Computational Mathematics
    1. MATH 159 Differential Equations
    2. MATH 164 Introduction to Numerical Analysis
    3. Any technical elective approved for the Computer Science track

E. Computer Networking
    1. CSCI 450 Introduction to Computer Networks,
    2. CSCI 451 Applied Wireless networks
    3. CSCI 452 Internet/Web Programming
    4. INFO 393 Network/Internet Security Management

F. Computational Physics
    1. PHYS 154 Computational Physics I
    2. PHYS 155 Computational Physics II
    3. MATH 159 Differential Equations or MATH 164 Introduction to Numerical Analysis
    4. Any technical elective approved for the Computer Science track

G. Computer Security
    1. INFO 395 Information Assurance
    2. INFO 391 Intro. to Information Security
    3. CSCI 453 Intro to Cybersecurity I
    4. CSCI 454 Intro to Cybersecurity II, or INFO 393 Network/Internet Security Management

H. Computer Science
    1. In Junior Fall semester, the Tech. Elective course is Introduction to Computer Networking
    2. Any CSCI course numbered 300 to 499 (excluding required courses).
    3. Any CSCI Graduate courses open to seniors.
    4. Computer Engineering EECE 211 Introduction to Digital Systems 5. Computer Engineering EECE 406 Advanced Digital Systems
    6. Electrical Engineering ELEG-416 Microcomputer Design.
    7. Electrical Engineering ELEG-417 Microprocessor Applications.
    8. Any Mathematics courses not required and having a Calculus prerequisite.
    9. INFO-304 Visual Basic OOP
    10. INFO-330 Data Base Management
    11. Any course approved by the program director.

Honor Code Policy for Programming Projects

Unless otherwise stated, at the time that an assignment or project is given, all work handed in for credit is to be the result of individual effort. (In some classes group work is encouraged; if so, that will be made explicit when the assignment is given.)

I. You (or your group, if a group assignment) may:

- seek assistance in learning to use the computing facilities;
- seek assistance in learning to use special features of a programming language's implementation;
- seek assistance in determining the syntactic correctness of a particular programming language statement or construct;
- seek an explanation of a particular syntactic error;
- seek explanations of compilation or run-time error messages

II. You (or your group, if a group assignment) may not seek assistance from anyone else, other than your instructor or teaching assistant:

- in designing the data structures used in your solution to a problem;
- in designing the algorithm to solve a problem;
- in modifying the design of an algorithm determined to be faulty;
- in implementing your algorithm in a programming language;
- in correcting a faulty implementation of your algorithm
- in determining the semantic correctness of your algorithm.

III. Unless permission to do so is granted by the instructor, you (or your group, if a group assignment) may not

- give a copy of your work in any form to another student;
- receive a copy of someone else's work in any form;
- attempt to gain access to any files other than your own or those authorized by the instructor or computer center;
- inspect or retain in your possession another student's work, whether it was given to you by another student, it was found after other student has discarded his/her work, or it accidentally came into your possession;
- in any way collaborate with someone else in the design or implementation or logical revision of an algorithm;
- present as your own, any algorithmic procedure which is not of your own or of the instructor's design, or which is not part of the course's required reading (if you modify any procedure which is presented in the course's texts but which is not specifically mentioned in class or covered in reading assignments, then a citation with page number must be given);

- incorporate code written by others (such as can be found on the Internet);

IV.You must:

- report any violations of II and III that you become aware of;
- if part of a group assignment, be an equal "partner" in your group's activities and productions, and represent accurately the level of your participation in your group's activities and productions.

Dress Code Policy

No lewd or obscene clothing is allowed in class, including hats, do-rags, etc. Cell phones are prohibited in class, unless being used as recorders.

ADMISSIONS INFORMATION

The University promotes academic excellence through a highly selective admission process. Students who are admitted show strong personal motivation along with backgrounds of consistent academic growth and achievement. The University also attracts and seeks out socially and economically deprived students who show promise of gaining from a Howard University education.

To protect its character and standards of scholarship, the University reserves the right, and the applicant concedes to the University the right, to deny admission to any student at any time for any reason the University deems sufficient.

Note to Prospective Students

On September 24, 1983, the Board of Trustees of Howard University adopted the following policy statement regarding applicants for admission: "Applicants seeking admission to Howard University are required to submit accurate and complete credentials and accurate and complete information requested by the University. Applicants who fail to do so shall be denied admission. Enrolled students, who as applicants failed to submit accurate and complete credentials or accurate and complete information on their application for admission shall be subject to dismissal when the same is made known regardless of classification."

Application Procedures

Evaluation of an applicant's qualifications for admission is based on high school course work, grade point average, class rank, test scores, extracurricular activities and letters of recommendation. An essay, audition, portfolio, or interview also may be required. Applicants for regular admission must have completed and be able to document high school graduation or its GED equivalency. Transfer students not holding an Associate Degree are also required to have completed high school graduation or its GED equivalency.

Application Deadlines

Domestic applicants should apply by the following deadlines:

Fall Semester Spring Semester- November 1
November 1- Early Action Summer Sessions- April 1 February 15-
Traditional Action

International applications should apply by the following deadlines:

Fall Semester Spring Semester- September 1
November 1- Early Action Summer Sessions- February 1 February 15-
Traditional Action

Students who plan to enroll at Howard University must complete and submit the following credentials:

- Application for Admission
- Nonrefundable $45 application fee, paid by money order, cashier's check, or credit card only.
  Howard University does not waive application fees.
- Official high school transcript or GED certificate
- Results from the Scholastic Aptitude Test (SAT) or the American College Test Assessment (ACT)
- One letter of recommendation from a high school counselor
- One letter of recommendation from a high school teacher/professor
- Essay
- Resume (optional)

All credentials must be received by EM/Admission by the designated deadline. Admitted students who intend to enroll at Howard University pay a $300 non-refundable enrollment fee by May 1st to secure their place in the class. Students are considered for housing once they have been accepted and have applied for housing and a $50 housing fee.

For more detailed information on undergraduate admissions and to apply on-line go to: http://howard.edu/enrollmentmanagement/admission/Default.htm

FINANCIAL SUPPORT

The Computer Science Program does not offer direct financial aid to undergraduate students other than work study. For information on other forms of financial aid (i.e. grants, scholarships, etc.) should direct their inquiries to the university Office of Financial Aid. The Mission of the Office of Financial Aid, Scholarships and Student Employment is to provide equitable financing options to prospective and current students, through exceptional and confidential customer service, while serving as a responsible fiduciary agent for the federal and state governments, as well as the University and its benefactors.

Although many factors help to determine the amount you receive, your Financial Aid Award is based primarily on your demonstrated financial need. You must complete the FAFSA each year to have your need determined. Your need is the difference between the cost of attendance and the amount you and your family are expected to contribute (EFC - expected family contribution). Once you are admitted to the University, and your file is complete, the Office of Financial Aid will send you a Financial Aid Award Package

You and your family are primarily responsible for financing your education. You and your family are expected to make a maximum effort to assist you with college expenses. You are also expected to contribute to your college expenses from sources that may include savings, summer earnings, monetary gifts from friends and relatives or other sources. Financial aid should be viewed as supplementary to your family’s contribution.

The income and asset information which you (and your parents if you are a dependent student, or your spouse if married) provide on the FAFSA enables the U.S. Department of Education's Central Processing System (CPS) to determine your family's contribution. Certain allowances such as the standard cost of living, retirement needs, and future indebtedness are considered and subtracted from total income and assets.

The Office of Financial Aid, Scholarships and Student Employment does not advance financial aid funds to students. Save and budget for fall semester book, supplies and rent due at the beginning of the school year in the event your aid is not ready for disbursement. Late applications or delays in returning required documents may prevent timely payments.
For more information on financial support go to: http://www.howard.edu/financialaid/

REGISTRATION INFORMATION

It is extremely important that you see your academic advisor prior to registering to ensure that you select the appropriate courses, complete a Request for Registration form, and get a personal identification number (PIN). Our current registration system is designed to prohibit students from registering for classes for which they have not completed the required pre- or co-requisite courses. Therefore, if you encounter a "registration error," this means that the system does not recognize you as having met the prerequisite(s) for the selected course. You must make another selection, or meet with your advisor for a course prerequisite override. If your advisor feels that you have met the prerequisite(s) for a particular course, he/she will approve your course selection.

1. Read over these instructions, or print them out. Once you are finished, go to the bottom of this page and click "Proceed to Bison Web Registration and Students Service".
2. Click LOG IN TO SECURE AREA on the Bison Web homepage.

3. Enter the "@" sign followed by your student identification number. Then enter your PIN. Your PIN must be six (6) numerical digits. Click the "LOGIN" button.

For information on your PIN number, please use one of the following resources:

- Student Reference Manual (page11)
- Your advisor
- Enrollment Management (202-806 2705)
- Courtesy desk in the Blackburn Center Ballroom

4. Type in your PIN again on the Login Verification Page, and click the LOGIN button.
5. If this is the first time you have signed on, a TERMS OF USAGE PAGE will display. Please read and if you accept the terms, click the CONTINUE button. If you do not accept the terms, click the EXIT button.
6. Select the phrase Student Services and Financial Aid.
7. Select the phrase Registration.
8. When the REGISTRATION page displays, click on SELECT TERM.
9. When the SELECT TERM page displays, click on the arrow at the right of the word TERM and select the appropriate term.
10. Click on the SUBMIT TERM button. The system will return you to the REGISTRATION page.
11. Click on CHECK YOUR REGISTRATION STATUS to assure you are able to register. If there are no holds which prevent registration click on the MENU at the top right of the page. If you are not able to register click the exit button at the top of the page.
12. When the registration page displays click on LOOK UP CLASSES TO ADD and follow the instructions.
13. When the classes are displayed, select the courses you want by clicking the boxes on the left side of the courses. When all courses are selected, click the REGISTER button. If there are no errors, you are now registered. If there are errors, you must restart from step 11.

This completes the registration process. Please verify your course selections by printing your schedule and making sure that the appropriate grade mode has been selected. If you need further assistance, call 8062705.

GRADUATION REQUIREMENTS

The curriculum in place at the time of the student’s entrance into the university does, in effect, represent a contract between student and university. Therefore, all parties must agree to any amendment. (As the program understands it, the "contract" is null and void if the student drops out of school for a semester or more.)

Disciplines (and program capabilities) change over time. What the Computer Science Program does is allow the student to graduate if he or she does any one of the following:

1. Meet ALL requirements from the curriculum in place when the student enters.
2. Meet ALL requirements from the curriculum in place when the student wishes to graduate.
3. Meet ALL requirements from any intermediate curriculum between date of entry and date of graduation.

We feel we need all this flexibility because of changes in the discipline. Unfortunately, we have to exercise a great deal of care in order to track these options for students and make sure that no student graduates without fulfilling all requirements and that we have appropriate records for ABET review.

STAFF

Antoinette T. Griffin	atgriffin@howard.edu	(202) 806-6585	2120B Downing Hall

CONTACT US

General Program Information:
Contact: Dr. Harry Keeling, Director, Computer Science
Phone: (202) 806-2207
FAX: (202) 806-4531
Web: http://www.eecs.cea.howard.edu/

Surface Mail:
Department of Electrical Engineering and Computer Science
College of Engineering and Architecture
Howard University
L.K. Downing Hall
2300 Sixth Street, NW
Washington, DC 20059