COLLEGE OF ENGINEERING AND ARCHITECTURE

Catalog Year: 2022 - 2026

COMPUTER SCIENCE

PROGRAM HANDBOOK

BACHELOR OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

1867
HOWARD UNIVERSITY

1016 Downing Hall
2300 Sixth St. NW
Washington, DC 20059

202-806-6595

https://cea.howard.edu/

Excellence in Computing.
Impact in the World.

UNDERGRADUATE STUDY IN COMPUTER SCIENCE

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

Catalog 2022 - 2026

Table of Contents

THE UNIVERSITY AND PROGRAM OVERVIEW

The University and Community

Howard University is a comprehensive, private, historically Black university classified among Research 1 (R1) institutions, reflecting the highest level of research activity in the United States. Founded in 1867, the University is committed to providing an educational experience of exceptional quality to students of high academic potential. Central to its mission is the cultivation of faculty excellence in teaching, research, and service, with a focus on developing graduates who are both intellectually distinguished and socially conscious.

Howard's mission—formally articulated in a 1989 resolution of its Board of Trustees—emphasizes leadership in addressing critical human and social challenges in the United States and globally. The University continues to be recognized for its strong academic programs, leadership in producing Black professionals (particularly in medicine, law, engineering, and the sciences), and its role in advancing equity in higher education.

Howard enrolls approximately 13,000 students, including undergraduate, graduate, and professional students across 14 schools and colleges. The University maintains a competitive academic profile while continuing its legacy of access and opportunity.

Washington, DC, the capital of the United States, serves as a global center for government, policy, research, and international affairs. The Washington, DC region offers a rich ecosystem for academic, professional, and cultural engagement. Students and faculty benefit from proximity to federal agencies, research institutions, think tanks, and leading industries. The city is home to world-renowned museums (many within the Smithsonian Institution), national monuments, performing arts venues, and a vibrant culinary and cultural scene.

This dynamic environment provides Howard University students with unparalleled opportunities for internships, research collaborations, public service, and global engagement.

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. Conduct research and/or create innovative and impactful solutions to solve societal and technical challenges.

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, MD21202-4012-telephone: (410)347-7700. The Program Educational Objectives embrace outcomes-based learning. Students are prepared to achieve the following six Student Outcomes:

1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Currently, our faculty includes 12 tenured and tenure-track faculty, 4 faculty lecturers, and 4 adjunct faculty. We also have 2 shared technical support staff and 2 administrative support staff (4 people). Most of our undergraduate students are full-time. Students contribute to nearly every aspect of the department's operation. In addition to taking a wide variety of courses, they have the opportunity to participate in undergraduate research, 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: https://profiles.howard.edu/node/66726

Anietie Andy, Assistant Professor; Ph.D., Howard University, 2016. Artificial Intelligence, Natural Language Processing, Language Translation, Algorithms.

Jeremy Blackstone, Assistant Professor; Ph.D., University of California San Diego, 2021. Behavioral cybersecurity, hardware security, network security, and artificial intelligence.

Legand L. Burge III, Professor; Ph.D., Oklahoma State University, 1998. Parallel and distributed computing, operating systems, computer networking, CS Education, Tech Innovation, Applied AI/ML.

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

Noha Hazzazi, Assistant Professor; Ph.D., George Mason University, 2017. Software Engineering.

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

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

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

Linwei Niu, Associate Professor; Ph.D., University of South Carolina, 2006. real-time embedded systems, design automation, energy-efficient computing, low power design, software/hardware co-design, cyber-physical systems, ubiquitous and pervasive computing, and parallel and distributed systems.

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

Danda Rawat Professor and Associate Dean for Research; Ph.D., Old Dominion University, 2010. cybersecurity, machine learning, big data analytics, and wireless networking for emerging networked systems.

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

Lecturers

Vijay Chaudhary, Lecturer; Ph.D., Howard University, 2023. Artificial Intelligence, Data Science, Algorithms, Cybersecurity.

Glenwood Elam, Jr., Lecturer; MBA., Penn State University, 1996. Information Technology.

Desta Haileselassie Hagos, Lecturer; Ph.D., University of Oslo, 2020. Artificial Intelligence, Data Science, Algorithms, Machine Learning, Deep Learning, Mobile Systems.

Anamika Paul Rupa, Lecturer; Ph.D., University of Maryland Baltimore County, 2023. Information Systems, Artificial Intelligence, Data Science.

Adjuncts

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

Torrance Fennell, Adjunct Lecturer Ph.D., Capella University, 2016, Information Technology, Computer Networking, Telecommunications, Wireless Networking.

Guy Lingani, Adjunct Lecturer Ph.D., Howard University, 2015, Cybersecurity, Artificial Intelligence, Cloud Computing.

Bernard Woolfolk, Adjunct Lecturer, 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.

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

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, Danda B. Rawat, Anietie Andy, Vijay Chaudhary,

Artificial Intelligence / Machine Learning / Data Science
Harry Keeling, Reginald Hobbs, Danda B. Rawat, Gloria Washington, Anietie Andy, Jeremy Blackstone, Vijay Chaudhary, Anamika Paul Rupa, Guy Lingani

Bioinformatics and Computational Biology
Chunmei Liu, Legand Burge, Guy Lingani

Computer Architecture
Jiang Li, Linwei Niu, Legand Burge, Danda Rawat

Databases and Data Visualization and Analysis
Moses Garuba, Danda B. Rawat, Chunmei Liu, Anietie Andy, Gloria Washington

Computer Networks
Jiang Li, Danda B. Rawat, Torrance Fennell, Legand Burge, Todd Shurn, Guy Lingani

Distributed/Parallel Computation and Operating Systems
Legand Burge, Danda B. Rawat, Moses Garuba, Linwei Niu,, Jiang Li, Torrance Fennell, Guy Lingani

Information Assurance and Computer Security
Jeremy Blackstone, Gloria Washington, Danda Rawat, Jiang, Li, Moses Garuba, Legand Burge

Mobile Computing
Legand Burge, Danda B. Rawat, Gloria Washington, Jiang Li, Vijay Chaudhary, Anamika Paul Rupa

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

Human-Computer Interaction, Human-Centered Computing
Gloria Washington, Legand Burge, Danda B. Rawat

Wireless Networking and Security
Danda B. Rawat, Jiang Li, Vijay Chaudhary

Internet of Things, Cyber Physical Systems and Resiliency
Danda B. Rawat, Jiang Li

Robotics
Jeremy Blackstone, Harry Keeling, Linwei Niu, Reginald Hobbs, Danda Rawat

Performance Modeling and Simulation
Legand Burge, Jeremy Blackstone, Danda Rawat, Gloria Washington, Jiang Li, Linwei Niu

Software Engineering
Legand Burge, Gloria Washington, Harry Keeling, Noha Hazzazi

CS Education / EdTech
Harry Keeling, Gloria Washington, Legand Burge, Danda B. Rawat

Faculty Contact Information

Program Director

Harry Keeling	hkeeling@howard.edu https://profiles.howard.edu/harry-keeling	2120B Downing Hall

Graduate Program Director and Director of Information Security Certificate Program

Jeremy Blackstone, Grad Director	jeremy.m.blackstone@howard.edu https://profiles.howard.edu/jeremy-blackstone	2012B Downing Hall
Danda Rawat, Certificate Program Director	danda.rawat@Howard.edu https://profiles.howard.edu/danda-rawat	2120B Downing Hall

Undergraduate Program Director

Harry Keeling, Associate Professor	hkeeling@Howard.edu https://profiles.howard.edu/harry-keeling	1012C Downing Hall

Faculty:

Anietie Andy, Assistant Professor	anietie.andy@Howard.edu https://nlp.howard.edu/	2036A Downing Hall
Jeremy Blackstone, Assistant Professor	jeremy.m.blackstone@howard.edu https://profiles.howard.edu/jeremy-blackstone	2012B Downing Hall
Legand Burge, Professor	lburge@Howard.edu https://profiles.howard.edu/legand-burge	2040 Downing Hall
Moses Garuba, Professor	mgaruba@Howard.edu https://profiles.howard.edu/moses-garuba	3015 Downing Hall
Noha Hazzazi, Assistant Professor	noha.hazzazi@Howard.edu https://profiles.howard.edu/noha-hazzazi	2112C Downing Hall
Harry Keeling, Associate Professor	hkeeling@Howard.edu https://profiles.howard.edu/harry-keeling	1012C Downing Hall
Jiang Li, Associate Professor	jli@Howard.edu https://profiles.howard.edu/jiang-li	2038C Downing Hall
Chunmei Liu, Professor	chunliu@howard.edu https://profiles.howard.edu/chunmei-liu	2038B Downing Hall
Linwei Niu, Associate Professor	linwei.niu@Howard.edu https://profiles.howard.edu/linwei-niu	2038A Downing Hall
Todd Shurn, Associate Professor	tshurn@Howard.edu https://profiles.howard.edu/todd-shurn	1110D Downing Hall
Danda B. Rawat, Professor	danda.rawat@Howard.edu https://profiles.howard.edu/danda-rawat	3007 Downing Hall
Gloria Washington, Associate Professor	gloria.washington@Howard.edu https://sites.google.com/view/affective-biometrics-	1110B Downing Hall

	lab-resea	
Vijay Chaudhary, Lecturer	vijay.chaudhary@howard.edu	1110C Downing Hall
Glenwood Elam, Jr., Lecturer	glenwood.elam@howard.edu	1016 Downing Hall
Desta Haileselassie Hagos, Lecturer	desta.hagos@howard.edu https://profiles.howard.edu/desta-haileselassie-hagos	411 HUIRB
Anamika Paul Rupa, Lecturer	Anamikal.Rupa@howard.edu https://profiles.howard.edu/anamika-paul-rupa	1012A Downing Hall
Bernard Woolfolk, Lecturer	bernard.p.woolfolk@Howard.edu https://profiles.howard.edu/bernard-woolfolk	1016 Downing Hall
Reginald Hobbs, Adjunct Lecturer	reginald.hobbs@Howard.edu https://profiles.howard.edu/reginald-hobbs	1016 Downing Hall
Torrance Fennell, Adjunct Lecturer	torrance.fennell@Howard.edu http://www.eece.cea.howard.edu/users/torrancefennell	1016 Downing Hall
Guy Lingani, Adjunct Lecturer	gmlingani@Howard.edu https://profiles.howard.edu/guy-lingani	2112A Downing Hall

OVERVIEW OF DEGREE AND REGUIREMENTS

The Department of Electrical Engineering and Computer Science 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 and Ph.D. 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 Department 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	58
Technical Electives	15
Liberal Arts Core Curriculum	38 (41)
Social Sciences and Humanities Elective Courses (African American Cluster Core)	9
Total	**120 (123)**

Computer Science Core (58 credit hours)
CSCI-100 Introduction to Computer Science, 3
CSCI-120 Exploring Computer Science, 2
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-213 Linux 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-401 Operating Systems, 3
CSCI-449 Intro to Computer Networks, 3
CSCI-453 Intro to Cybersecurity I, 3
CSCI-473 Applied Data Science, 3
CSCI-491 Senior Design Project I, 3
CSCI-492 Senior Design Project II, 3

University Requirements
University-Wide Requirements for graduation include a minimum GPA: 2.0 for all undergraduate coursework. In addition, students must complete last 30 credits at Howard University.

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	Exploring Computer Science CSCI-120 Introduction to Computer Sci. CSCI-100
Historical Awareness	African American Cluster Elective Humanities/Social Science Elective
Empirical Analysis	CSCI-491 Senior Design Project I CSCI-492 Senior Design Project II
Quantitative Literacy & Statistical Reasoning	Applied Data Science CSCI-473
Social and Human Relations	Non-Technical Electives - Humanities/Social Science
Other Core Experiences	University Events – lectures, convocation, College/Department 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 the departmental 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.

COLLEGE OF ENGINEERING AND ARCHITECTURE

COMPUTER SCIENCE

BACHELOR OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Catalog Year: 2022 - 2026

1867 HOWARD UNIVERSITY

4-YEAR PLAN OF STUDY

Students seeking the **Bachelor of Science in Computer Science** must complete a minimum of 120 credit hours comprised of core courses in computer science (58 credits), technical electives (15 credits), liberal arts core curriculum (41 credits), and non-technical electives (9 credits): African American Cluster Core (3 credits), and choirce of - social science elective (3 credits), and/or humanities elective (3 credits).

FALL SEMESTER 1

Number	Course Title	Credits
CSCI 100	Intro to Computer Science	3
CSCI 120	Exploring in Computer Science	2
--	Non-Technical Elective	3
--	Science Lecture A	4
--	Science Lab A	0/1
ENGW --	English First-Year Writing (1)	3
	Total Credits	15/16

SPRING SEMESTER 1

Number	Course Title	Credits
MATH 156	Calculus I	4
CSCI 213	Linux Lab	1
ENGW --	English First-Year Writing (2)	3
COMM 101	Principles of Speech	3
CSCI 135	Computer Science I	4
	Total Credits	15

FALL SEMESTER 2

Number	Course Title	Credits
MATH 157	Calculus II	4
--	Science Lecture B (1)	4
--	Science Lab B (1)	0/1
CSCI 136	Computer Science II	3
CSCI 201	Computer Organization I	3
	Total Credits	14/15

SPRING SEMESTER 2

Number	Course Title	Credits
MATH 181	Discrete Structures	3
CSCI 354	Computer Science III	3
CSCI 375	Software Engineering	3
CSCI 202	Computer Organization II	3
--	Science Lecture B (2)	4
--	Science Lab B (2)	0/1
	Total Credits	16/17

FALL SEMESTER 3

Number	Course Title	Credits
CSCI 341	Theory of Computation	3
CSCI 401	Operating Systems	3
CSCI 470	Fundamentals of Algorithms	3
CSCI 449	Computer Networks	3
CSCI 453	Introduction to Cybersecurity	3
	Total Credits	15

SPRING SEMESTER 3

Number	Course Title	Credits
CSCI 350	Structure of Programming Languages	3
--	Technical Elective	3
CSCI 432	Database Systems	3
MATH 180	Introduction to Linear Algebra	3
ENGL 009	Technical Writing	3
	Total Credits	15

FALL SEMESTER 4

Number	Course Title	Credits
CSCI 363	Large Scale Programming	3
CSCI 473	Applied Data Science	3
--	Technical Elective	3
CSCI 491	Senior Project I	3
--	Technical Elective	3
	Total Credits	15

SPRING SEMESTER 4

Number	Course Title	Credits
--	Technical Elective	3
--	Technical Elective	3
--	Non-Technical Elective	3
CSCI 492	Senior Project II	3
--	Non-Technical Elective	3
	Total Credits	15

Total Credits: 120

More Information :

The 4-Year Plan of Study is a guide for students to successfully complete the program in four years of study. It is not a substitution for academic advising. Students are expected to download and maintain their own CHECKLIST, and check-in with their academic advisor every semester.

The prerequisite structure for courses and technical elective options are available in the program handbook.

Courses may not be offered in semesters in which they do not appear listed on the scheme.

DOWNLOAD CHECKLIST

https://bit.ly/CS-CheckList-2022-2026

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
Lewis K. Downing Hall, 2300 Sixth Street, NW, Suite 1016, Washington, DC 20059
Phone: 202.806.6585 Email: eecs@howard.edu
www.cea.howard.edu

Computer Science Option for Minors

The Computer Science option for all 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-200	Introduction to Mobile Applications	3
CSCI-203	Object-Oriented Programming using Java	3
CSCI-263	Web Makers	3
CSCI-165	Scientific Computing for Eng.	3
EECE 212	Fundamentals of Digital Systems,	4
CSCI-432	Database Systems	3
MATH 181	Discrete Structures	3
CSCI-354	Computer Science III	3
CSCI-375	Software Engineering	3
CSCI-363	Large Scale Programming	3
CSCI-421	Computer and Video Game Development	3
CSCI-402	Mobile Application Development	3
CSCI-410	Modeling and Simulation	3
CSCI-453	Introduction to Cybersecurity I	3
CSCI-473	Applied Data Science	3

Any other selection must be approved in advance by the CS Program Director or Department Chairperson.

Fundamentals of Digital Systems (EECE-212) 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.

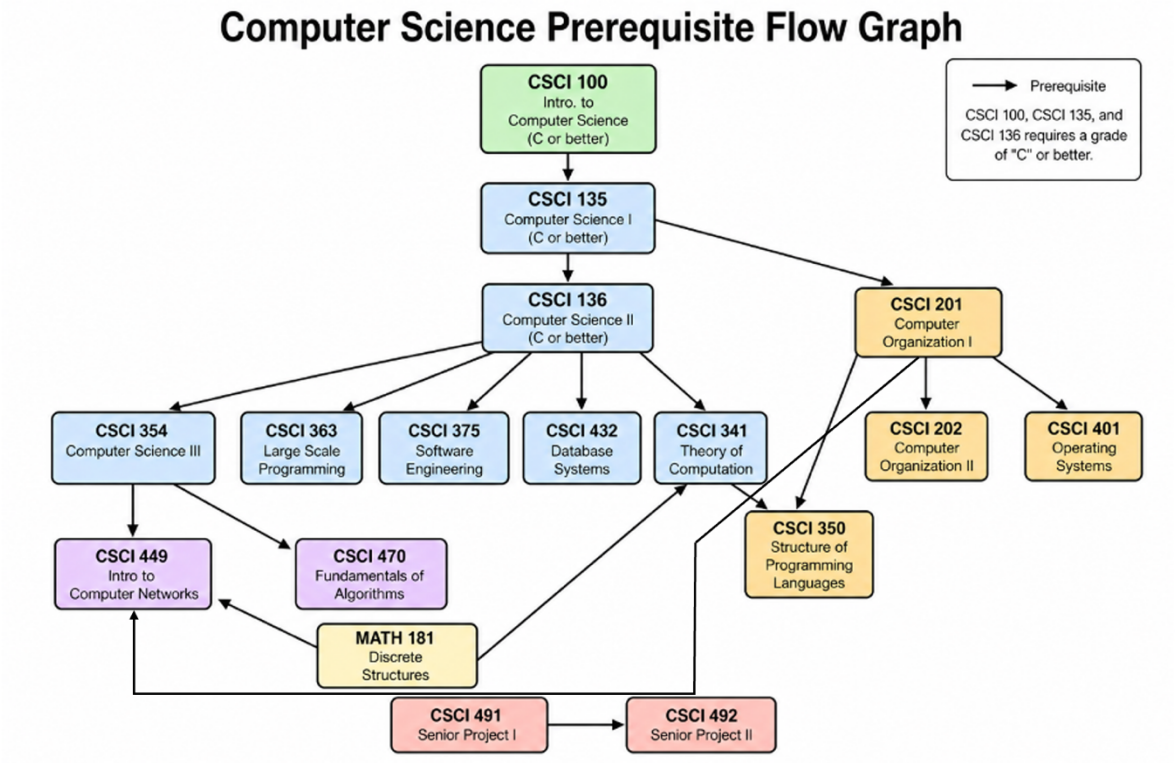
Large Scale Programming (CSCI-363), Computer and Video Game Development (CSCI-421), Mobile Application Development (CSCI-402) should not be taken until the student has completed Computer Science II (CSCI-136).

Computer and Video Game Development (CSCI-421) 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 graph 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-136 Computer Science II
CSCI-375 Software Engineering	'C' or better in CSCI-136 Computer Science II
CSCI-350 Structure of Programming Languages	CSCI-201 Computer Organization I and CSCI-341 Theory of Computation
CSCI-449 Intro to Computer Networks	CSCI-354 Computer Science III, CSCI-201 Computer Organization I, MATH 181 Discrete Structures
CSCI-341 Theory of Computation	CSCI-136 Computer Science II and MATH 181Discrete Structures
CSCI-432 Database Systems	CSCI-136 Computer Science II
CSCI-401 Operating Systems	CSCI-201 Computer Organization I
CSCI-470 Fundamentals of Algorithms	CSCI-354 Computer Science III
CSCI-491 Senior Project I	Senior Standing
CSCI-492 Senior Project II	CSCI-491 Senior Project I



Undergraduate Courses Catalog

CSCI-100 – Introduction to Comp. 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 department, the different track options and career opportunities along the different tracks.

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-120 – Exploring Computer Science – 2 Credits
Provides an overview of the field of computer science, introducing students to key concepts in computing and "computational thinking." This is a required freshman course introducing CS fundamentals. Course explores CS sub-domains such as artificial intelligence, cybersecurity, data science, machine learning, interactive media (games), human computer Interface (HCI) usability testing, and robotics. In addition, students will be exposed to design thinking for innovation. Class will include presentations from leading experts from academia, government and industry allowing students to ask questions directly. Topical assignments will expose students to introductory topics through individual exercises and team projects.

CSCI-135 – Computer Science I – 4 Credits
This course provides an introduction to 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 a high-level 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 a high-level language. A major focus of the lectures will be to provide an overview of real-world problem-solving concepts and top-down software design. Prereq: CSCI-100 Intro. to Computer Science (with a grade of 'C' or better) or equivalent course.

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. Prereq.: CSCI-135 Computer Science I (with a grade of 'C' or better).

CSCI-140 – Programming Team – 1 Credit

• Course has not been offered in over 2 years.

This course is for participants of the programming team.

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-172 – Interview Prep – 1 Credits
This course will provide students with workshops and mock interviews to prepare them for careers in tech; with a specific interest in technical coding interviews at companies like Google. Topics covered will include interview best practices, problem solving techniques, how to communicate your thought process, and how to diagram coding problems. In addition, this course will cover topics such as: work-life balance, company culture, salary negotiations, relocation, roles within tech companies, and management vs technical tracks.

CSCI-200 – Introduction to Mobile App Dev• – 3 Credits
This course introduces non- and beginning programmers to mobile app development using JavaScript and a modern, widely used framework such as React Native. Students will learn how to design, build, and deploy cross-platform mobile applications without relying on native development tools. Through hands-on projects, the course emphasizes practical skills in user interface design, application logic, and real-world app development workflows. CSCI-135 Computer Science I (with a grade of 'C' or better) or equivalent course or experience.

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. Prereq.: CSCI-135 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. Prereq.: CSCI-201 Computer Organization I.

CSCI-203 – Object-Oriented Programming using Java – 3 Credit
This course provides an introduction to 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. Prereq.: CSCI-135 Computer Science I (with a grade of 'C' or better).

• Course has not been offered in over 5 years.

CSCI-211/213 – Linux Lab (formerly Unix Lab) – 1 Credit
This course will present the basic concepts of LINUX 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-340 – Robotics: Computational Linear Algebra – 3 Credits
Robotics Computational Linear Algebra introduces problems at the "scale of life". It operates on the premise that linear algebra is the fundamental language of autonomous robotics, computer vision, and machine learning. Students learn techniques and write code to solve large linear algebra problems with hundreds of equations and thousands of variables.

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. Prereq.: CSCI-136 Computer Science II (with a grade of 'C' or better), and MATH 180 Discrete Structures.

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. Prereq.: CSCI-201 Computer Organization I, CSCI-341 Theory of Computation.

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. Prereq.: CSCI-136 Computer Science II (with a grade of 'C' or better).

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. Prereq: CSCI-136 Computer Science II (with a grade of 'C' or better).

CSCI-365 – Cloud Computing – 3 Credits
Cloud Computing provides the modern on-demand services for management and usage of large and shared computing resources including storage, computations, and communications. This course will cover in-depth knowledge for Cloud Computing and the practical experience in designing and implementing large-scale and composite business web applications on Cloud Computing platform.

CSCI-375 – Software Engineering – 3 Credits
This course is an introduction to the basic concepts of software engineering including the software lifecycle, requirements analysis, design, coding, testing and documentation. Professional ethics in computer science and the social impact of computing are discussed. Additional topics may include tools for software development, software metrics, and software maintenance. The objectives of the course are met using classroom lecturers, in-class labs, and a semester-long project developed in a team setting. Prereq: CSCI-136 Computer Science II (with a grade of 'C' or better).

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. Prereq.: 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: CSCI-354 Computer Science III (with a grade of 'C' or better).

CSCI-380 – Foundations of User Experience Design – 3 Credits
User Experience (UX) designers focus on the interaction that users have with products, like websites, apps, and physical objects. They make those everyday interactions useful, enjoyable, and accessible. This course includes instruction and practice-based activities and assessments that simulate real-world UX design scenarios and are critical for success in the workplace. You'll learn how to complete the design process from beginning to end, including: Empathizing with users; Defining user pain points; Coming up with ideas for design solutions; Creating wireframes, mockups, and prototypes; Testing designs through usability studies; Iterating on designs based on feedback. Coreq: CSCI-379 Introduction to Human Computer Interaction.

CSCI-383 – Affective Computing – 3 Credits
Affective computing represents a broad, interdisciplinary research and practice area focusing on a range of topics, including computational models of emotion, cognitive-affective architectures, affective user modeling; emotion sensing and recognition; emotion expression; and the use of emotions to improve human-computer interaction across a range of contexts including health informatics and immersive environments. This course will provide an introduction to affective computing and human-computer interaction through a combination of lectures, student presentations of selected literature, projects and class discussions. The course content and format will be appropriate for computer science, cognitive science, psychology, human factors, and industrial engineering students (advanced undergraduate/graduate). Students in this course will design an experimental study to examine how emotional states like stress, anxiety, frustration, anger, and/or depression can be measured through the use of mobile applications. At the end of the course, students will be paired

• Course has not been offered in over 5 years.

with a student from the Architecture Department class called Immersive Environments to perform a study on how to design immersive environments to reduce stress, anxiety, frustration, anger, and/or depression.

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-400 – Computer & Network Forensics I – 3 Credits
Computer and Network Forensics I is the first course in the two-semester series on computer and network forensics and focuses on computer system forensics (while the second course on network forensics). It consists of three major components: assembly level reverse engineering (to understand the logic of a program given only machine code), file system forensics (to find traces of past activities on data stored in nonvolatile storage) and memory forensics (to find traces of past stalactites on data stored in volatile storage). It will also briefly discuss the law aspect of digital forensics. In order to understand the materials, students are expected to have the following prerequisites: good programming skills in both high level (preferably C/C++) and assembly level, operating systems and basic familiarity with Linux and Windows.

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. Prereq.: CSCI-201 Computer Organization I.

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. Prereq: CSCI-136 Computer Science II (with a grade of 'C' or better).

CSCI-403 – Introduction to Big Data Analysis – 3 Credits
The primary goal of big data analytics is to analyze large volume of data. The analysis of big data requires the organization, storage, and processing of data at a scalable and efficient way which goes beyond the capabilities of traditional information technologies. In this course, we will cover modern massive data analysis technologies including different platforms, models, and languages, we will also look at real world applications and how they can be implemented on big data platforms. Topics in this course include Map reduce/Hadoop, NoSQL stores, languages such as Pig Latin and JAQL, large-scale data mining and visualization. It will include programming projects that will provide students with hands-on experience on building data-intensive applications using existing big data tools and platforms.

EGPP-405 – CS Undergraduate Research – 3 Credits

• Course has not been offered in over 5 years.

This course offers undergraduate students the opportunity to engage in faculty-mentored research. Participants will work on real-world, hands-on projects that develop practical skills and deepen their understanding of computer science research.

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-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. Prereq: CSCI-354 Computer Science III (with a grade of 'C' or better), 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. Prereq: CSCI-421 Computer and Video Game Development.

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. Prereq: CSCI-136 Computer Science II (with a grade of 'C' or better).

CSCI-433 – Introduction to Database Security – 3 Credits
An overview of both the theory of and applications for providing effective security in database management systems. Topics include conceptual frameworks for discretionary and mandatory access control, data integrity, availability and performance, secure database design, data aggregation, data inference, secure concurrency control, and secure transactions processing. Models for multilevel secure databases for both relational and object-relational databases are analyzed. Assignments focus on database security concepts.

CSCI-449 – Introduction to Computer Networks (formerly CSCI-450) – 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: CSCI-354 Computer Science III (with a grade of 'C' or better), CSCI-201 Computer Organization I, MATH 181 Discrete Structures.

• Course has not been offered in over 5 years.

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. Prereq: CSCI-449 Intro to Computer Networks.

CSCI-452 – Network and Web Programming – 3 Credits
This class will introduce (1) network programming, writing applications that communicate with each other via TCP/UDP sockets, and (2) Web programming, writing applications that are accessible through a Web Browser over the Hypertext Transport Protocol (HTTP). For web programming, both client side and server-side programming will be covered. The LAMP stack (Linux, Apache, MySQL, and PHP) will be used for server side, Javascript and CSS will be covered for client side. HTML will be taught before studying programming. As prerequisites, programming skills are needed. Light database experience is preferred but not required.

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. Prereq: Intro to Cybersecurity I.

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. Prereq: CSCI-213 Linux Lab.

CSCI-461 – Network and Web Programming II – 3 Credits
This project-based course is the second course in a two-course web development sequence within the Computer Science program. This project-based course emphasizes the design and implementation of dynamic, data-driven web applications using modern full-stack technologies. Students build interactive front-end applications with React while integrating external APIs, web services, and database-backed systems to create scalable, real-world solutions. The course focuses on client–server architecture, RESTful APIs, asynchronous programming, state management, authentication workflows, and deployment pipelines. Students gain hands-on experience consuming and manipulating live data, managing application state

• Course has not been offered in over 5 years.

across components, and deploying production-ready applications to cloud platforms. Emphasis is placed on professional software engineering practices, including version control, debugging, documentation, and collaborative development. By the end of the semester, students will produce a portfolio-quality capstone project that demonstrates technical proficiency in full-stack development, effective use of web services, and the ability to design and deploy applications that meet real user needs. This course prepares students for internships and entry-level roles in software engineering, web development, and product engineering. Prerequisites: Sophomore level required.

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. Prereq: CSCI-354: Computer Science III (with a grade of 'C' or better).

CSCI-471 – Applied Data Structures – 3 Credits
This course will teach students the skills needed for technical coding interviews at companies like Google. It will focus on understanding how to choose optimal algorithms and data structures for different problems, how to apply them, and how to explain their reasoning. Topics covered will include hash tables, recursion, linked lists, trees, and graphs. CSCI-136 Computer Science II (with a grade of 'C' or better).

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. Prereq: CSCI-375 Software Engineering.

CSCI-473 – Applied Data Science – 3 Credits
This first half of the course introduces students to the application of probability and statistics for data science. This course covers the fundamentals of probability theory and statistical inference used in data science; Probabilistic models, random variables, useful distributions, expectations, law of large numbers, central limit theorem; Statistical inference; point and confidence interval estimation, hypothesis tests, linear regression. This second half of the class is about learning from data, in order to gain useful predictions and insights. Separating signal from noise presents many computational and inferential challenges, which we approach from a perspective at the interface of computer science and statistics. Through real-world examples of wide interest, we introduce methods for five key facets of an investigation: data munging /scraping /sampling /cleaning in order to get an informative, manageable data set; data storage and management in order to be able to access data - especially big data - quickly and reliably during - subsequent analysis; exploratory data analysis to generate hypotheses and intuition about the data; prediction based on statistical tools such as regression, classification, and clustering; communication of results through visualization, stories, and interpretable summaries.

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. Prereq: CSCI-470: Fundamentals of Algorithms, MATH 189: Probability and Statistics.

• Course has not been offered in over 5 years.

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. Prereq: CSCI-470: Fundamentals of Algorithms.

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. Pre-req: CSCI-350 Structures of Programming Languages.

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. Prereq.: Operations Research.

CSCI-479 – Cybersecurity Network CPS/IoT – 3 Credits
This course is designed to introduce emerging topics related to cybersecurity challenges and practical cyber-defense/countermeasures in networked Cyber-Physical Systems (CPS) and Internet-of-Things (IoT). The course will cover fundamental concepts, technologies, theoretical understanding and practical basis for cybersecurity of networked CPS/IoT.

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. Prereq: Junior Standing.

CSCI-481 – Product Management – 3 Credits
This course introduces students to the role and responsibilities of a product manager (PM) and the skills required to lead successful product development. Students will gain hands-on experience in core product management competencies, including customer discovery, product pitching, requirements definition, user interface design, and data-driven decision-making using metrics. The course also emphasizes essential management skills such as influencing without authority, negotiation, and navigating difficult conversations. In addition, students will build confidence in product management interviews through targeted preparation

• Course has not been offered in over 5 years.

and practice. Throughout the course, students will apply their learning by developing a comprehensive, end-to-end product plan that integrates strategy, design, and execution.

CSCI-482 – Natural Language Processing – 3 Credits
This is an introductory course to the field of Natural Language Processing (NLP). This course offers an introduction to NLP for computer science majors (graduate and undergraduate students) and graduate students in biological or biomedical sciences, focusing on the computational techniques used to analyze and generate human language. In addition to NLP techniques, this course integrates cognitive neuroscience principles, exploring the brain's role in language and visual processing, learning, and memory plasticity. Neuroimaging techniques and the ethical implications of AI in cognitive neuroscience will also be examined.

CSCI-490 – Software Developer Studio – 3 Credits
Software Developer Studio 3 Credits This course is an integrative experience that brings together all components of the undergraduate computer science curriculum in an applied, hands-on real-world setting. The course is a bridge between the academic experience and the professional workplace. The objectives of the Capstone course are to give the student the experience of 1) being involved in building a non-trivial, real-world, software development project; 2) working with other developers in a team environment; 3) building a web application and 4) integrating knowledge of topics such as programming, data structures, algorithms, database, and software engineering within a team- project framework. Students will present their final product at the end of the semester. Prereq: CSCI-136 Computer Science II.

CSCI-491 – Senior Project I – 3 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. Prereq: Senior Standing.

CSCI-492 – Senior Project II – 3 Credits
Senior Project II 3 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 department's Systems Development Laboratory. The student is required to demonstrate the system solution to the faculty and the student body of the department. Prereq: Senior Project I.

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 actually 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 classroom 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 actually 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. Instructor Permission.

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 in order 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. Instructor Permission.

CSCI-495 – CS Practicum Experience – 3 Credits
Students will be grouped into teams of 3-4 members and will join a corporate project team. They will participate in all corporate project team meetings and activities for two days each week. Each student team will be assigned a specific project, and each member will set individual, measurable goals to track throughout the semester. The team will present their work to the corporate project team at least once during the course**.** Instructor Permission.

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. Prereq: Junior Standing, MEEG, EECE, CSCI major.

CSCI-991 – Co-Op Education• – 1 Credits
Description unavailable.

CSCI-999 – Co-Op Education• – 12 Credits
Description unavailable.

• Course has not been offered in over 5 years.

Science Requirements

Three science courses that develop and apply the scientific method must be taken with the following restrictions:

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

Science courses include:

- BIOL 101 Biology I•
- BIOL 102 Biology II•
- CHEM 003 Chemistry I, and CHEM 005 Chemistry Lab
- CHEM 004 Chemistry II, and CHEM 006 Chemistry Lab II
- PHYS 013 Physics for Engineers I, and PHYS 023 Physics for Engineers Lab I
- PHYS 014 Physics for Engineers II, and PHYS 024 Physics for Engineers Lab II
- PHYS 001. General Physics I•
- PHYS 002. General Physics II•

NOTE: CHEM003/004, CHEM 005/006, and PHYS 001, PHYS 002 will each add an additional 1 credit increase in your total credits.

Technical Electives

The objective of this requirement is to expand the student's contact with advanced topics in the major field geared toward their own interest. Students can explore any of the following courses as technical electives.

- Any CSCI course numbered 300 to 499 (excluding required courses), exception given to CSCI-200.
- Any CSCI Graduate courses open to seniors.
- Computer Engineering EECE 212 Fundamentals of Digital Systems
- Computer Engineering EECE 406 Advanced Digital Systems
- Computer Engineering EECE 417 Computer Systems Architect
- Computer Engineering EECE 416 Microcomputer Design.
- Any Mathematics courses not required, and having a Calculus I prerequisite.
- INFO 395 Information Assurance
- INFO 391 Intro. to Information Security
- INFO 393 Network/Internet Security Management
- EGPP 493 Bison Startup: Technology Entrepreneurship and Lean Startups
- EGPP 494 Bison Accelerate: Launch and Iterate
- EGPP 405 CS Undergraduate Research
- EGPP-421 Intro. to AI: Tools and Techniques
- EGPP-422 Ethical and Responsible AI
- CHEM 184 Methods in Computational Chemistry
- PHYS 154 Computational Physics I.

• Science course includes both Lecture and Lab.

- PHYS 155 Computational Physics II
- Any course approved by the Department Chair.

Students who want to explore technical elective courses based on a prescribed concentration pathway, should review the Suggested Concentration Pathways in the **APPENDIX I** and reach out to a Concentration Advisor for any questions or assistance.

Non-Technical (Social/Humanities) Electives

Three courses (each 3 credits) must be taken from the Social/Humanities Electives. Among these courses must be:

- One from the University's African American Cluster (i.e. any course Underlined below), or any course from AFRO, or AFST.
- 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 by the student's academic advisor and approved by the CS Program Director or Department Chairperson.

.

ARTH 161 Art Appreciation CLAS 014 Intro to Humanities I CLAS 015 Intro to Humanities II CLAS 016 Ideas in Antiquity CLAS 050 Ancient Egypt and Near East CLAS 051 Greece and Rome CLAS 101 Greek Literature CLAS 102 Roman Literature CLAS 103 Classical Art CLAS 108 Greek Drama CLAS 109 Classical Mythology CLAS 111 Satire and Comedy in the Ancient World CLAS 109 Classical Mythology CLAS 111 Satire and Comedy in the Ancient World CLAS 113 Women in the Ancient World CLAS 114 Love in Antiquity ENGL 014 Intro to Humanities I ENGL 015 Intro to Humanities II ENGL 054 Afro-Am Lit to 1940 ENGL 055 Afro-Am Lit since 1940 ENGL 056 Modern Caribbean ENGL 180 Third World Lit Myth and Archetype FREN 014 Intro to Humanities FREN 015 Intro to Humanities II FREN 100 Francophone Literature in English	FREN 106 Africa Cinema FREN 107 Women in Literature GERM 014 Intro to Humanities I GERM 015 Intro to Humanities II GERM 100 Individual & Society GERM 101 Literature of Love GERM 107 Women in Literature GERM 109 Slavic Mythology GERM 111 Classic Films in English HUMA 107 Women in Literature MUSC 100 Introduction to Music RUSS 014 Intro to Humanities RUSS 100 Russian Short Stories in English RUSS 101 Literature in Revolution RUSS 103 Love and Hate in Literature RUSS 109 Slavic Mythology SPAN 014 Intro to Humanities I SPAN 015 Intro to Humanities II SPAN 100 Hispanic Literature in English SPAN 107 Women in Literature THFD 010 Introduction to the Theatre
AFRO 005 Afro-American Studies I AFRO 006 Afro-American Studies II	HIST 001 Intro to Study of Civilization HIST 005 Intro to Black Diaspora I

AFST 106 Intro to African Studies CLAS 104 Greek Civilization CLAS 105 Roman Civilization CLAS 112 Ancient Law and Politics CLAS 115 Slavery in the Greco-Roman World GERM 145 Cultural Life of Germany I GERM 146 Cultural Life of Germany II	HIST 006 Intro to Black Diaspora II HIST 101 World Geography HIST 102 Economic Geography RUSS 145 Cultural Life of Russia RUSS 150 Dissident Soviet Writers
AFRO 131 Black Philosophy I AFRO 133 19th Century Black Social-Political AFRO 191 Thought Contemporary Slavery AFST 101 Intro to Contemporary Africa ANTH 120 Intro to Biological Anthropology CLAS 110 Blacks in Antiquity ECON 001 Principles of Economics I ECON 002 Principles of Economics II ECON 199 Intro to Urban Economics POLS 003 Intro to Comparative Politics POLS 005 Intro to African Politics POLS 011 State and Local Government POLS 143 Black Electoral Politics POLS 006 Pan-Africanism FASH 102 Perspectives on African American Dress ARTH 193 Black Body Dress and Culture	HIST 030 Intro to African History I HIST 031 Intro to African History II HIST 040 Intro the Hist of Latin Am and the Caribbean I HIST 041 Caribbean II HIST 050 Intro to European History I HIST 051 Intro to European History II HIST 054 Intro to England HIST 060 Intro to East Asian Civilization I HIST 061 Intro to East Asian Civilization II HIST 003 Europe and the Wider World I HIST 004 Europe and the Wider World II HIST 009 US History to 1877 HIST 010 US History since 1877 PSYC 050 Introduction to Psychology POLS 001 Intro to Political Science MUTP 100 Blacks in the Arts SOCI 001 Intro to Sociology

PROGRAM POLICIES

Program 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; In some classes Generative AI 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.

Intellectual Property (IP) Policy

The Department of Electrical Engineering and Computer Science encourages innovation, creativity, entrepreneurship, and research. Students frequently create software, algorithms, applications, datasets, reports, designs, inventions, and other intellectual property (IP) as part of coursework, research, and extracurricular activities.

Student Ownership of Intellectual Property

In general, students who are not employees of the University retain ownership of intellectual property they create as part of their academic coursework, independent projects, personal research, or entrepreneurial activities.

Examples include:

- Programming assignments and course projects
- Capstone projects
- Mobile and web applications
- Software tools and utilities
- Games and simulations
- Independent research conducted without sponsored funding
- Personal entrepreneurial ventures

University Interest in Intellectual Property

The University may claim ownership or certain rights to intellectual property created by students when one or more of the following conditions apply:

1. The intellectual property is developed through a sponsored research project funded by a federal agency, private foundation, corporation, or other external sponsor.
2. The intellectual property is developed using substantial University resources beyond those normally available to students for instructional purposes.
3. The intellectual property is developed in university research facilities outside the scope of the student's enrolled academic program.
4. The student is employed by the University and creates the intellectual property within the scope of their employment responsibilities.
5. The intellectual property is subject to contractual obligations, grant requirements, sponsored research agreements, licensing agreements, or other University commitments.

Research Projects and Sponsored Activities

Students participating in faculty-led research projects, grant-funded activities, research assistantships, innovation programs, incubators, or externally sponsored projects may be required to sign additional agreements governing intellectual property rights, confidentiality, publication, data access, and commercialization.

Students should consult with the faculty principal investigator and the appropriate University office before publicly disclosing, publishing, distributing, or commercializing intellectual property developed through such activities.

Course Projects

Unless otherwise specified in writing, intellectual property created solely for a course remains the property of the student or student team that created it.

Faculty members may retain limited rights to use student work for instructional, assessment, accreditation, recruitment, or program evaluation purposes, consistent with university policies and applicable law.

Team Projects

Students working on team projects are encouraged to discuss ownership, authorship, licensing, and commercialization expectations at the beginning of the project. In the absence of a written agreement, intellectual property created jointly by multiple students may be jointly owned by the contributing creators.

Questions Regarding Intellectual Property

Questions regarding ownership, licensing, patents, copyrights, software distribution, commercialization, or technology transfer should be directed to the Department Chair and the University's designated office responsible for intellectual property and technology commercialization. In all cases, University policy shall govern and take precedence over departmental guidelines.

Classroom Conduct and Attire Policy

Students are expected to maintain a professional and respectful learning environment. Clothing displaying lewd, obscene, or offensive language or imagery is not permitted in class. Head coverings such as hats, do-rags, and similar items may not be worn in class unless required for religious, cultural, or medical reasons. Cell phones and other personal electronic devices should be silenced and stored during class and may only be used when authorized by the instructor for instructional purposes or approved accommodations. Devices may be used as audio recorders only with the instructor's permission.

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 department understands it, the "contract" is null and void if the student drops out of school for a semester or more.)

Disciplines (and department 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. Please utilize Check List on page 12 to track your progress through the CS curriculum.

STAFF

Jada Williams, Program Coordinator	Jada.Williams2@howard.edu	1016 Downing Hall
Angela McRae Administrative Assistant III	Angela.Mcrae@howard.edu	1016 Downing Hall

CONTACT US

Phone: (202) 806-6585
Web: https://cea.howard.edu/academics/departments/electrical-engineering-and-computer-science/undergraduate/bscs

Advising Portal: https://howardu.sharepoint.com/sites/CSAdvising

Surface Mail:
Department of Electrical Engineering and Computer Science
College of Engineering and Architecture
Howard University
Room 1016 Downing Hall
2300 Sixth Street NW
Washington, DC 20059

APPENDIX I: Suggested Concentration Pathways

The Department of Electrical Engineering and Computer Science does not offer formal concentrations that are recorded on a student's transcript or degree. However, students who wish to develop expertise in a particular area of computer science may choose to follow one of several suggested concentration pathways. Each pathway consists of a recommended set of technical electives and science courses designed to provide additional depth in a specific area of interest. These pathways serve as academic advising guides and do not constitute official degree concentrations.

The available concentration pathways faculty advisors are listed below:

A. Computer Science – Dr. Burge and Dr. Keeling
B. Human-Centered Computing – Dr. Washington
C. Computational Biology – Dr. Liu
D. Computational Chemistry – Dr. Liu
E. Gaming – Dr. Shurn
F. Computational Mathematics – Dr. Andy
G. Computer Networks – Dr. Li
H. Computational Physics – Dr. Andy
I. Cybersecurity – Dr. Blackstone

Students are not required to select a concentration pathway. By default, all students follow the Computer Science Pathway (A), which provides the greatest flexibility in selecting technical and science electives. Students interested in pursuing one of the specialized pathways are encouraged to consult with the corresponding pathway advisor regarding course availability and selection, and academic planning.

SCIENCE REQUIREMENTS

Students must complete three science courses that develop and apply the scientific method. The science courses selected should align with the concentration pathway being pursued and must satisfy the following requirements:

Science Course Recommendations by Concentration Pathway

A. Computer Science
   - Any approved science course may satisfy Science A.
   - Any approved science sequence different from Science A may satisfy Science B (1) and Science B (2).
B. Human-Centered Computing
   - Same as Concentration Pathway – A. Computer Science
C. Computational Biology
   - Any approved science course outside the Department of Biology may satisfy Science A.
   - BIOL 101 Biology I - (Science B (1))
   - BIOL 102 Biology II - (Science B (2))
D. Computational Chemistry
   - Any approved science course outside the Department of Chemistry may satisfy Science A.
   - CHEM 003 Chemistry I and CHEM 005 Chemistry Lab I - (Science B (1))

- CHEM 004 Chemistry II and CHEM 006 Chemistry Lab II - (Science B (2))

E. Gaming
- Same as Concentration Pathway – A. Computer Science

F. Computational Mathematics
- Same as Concentration Pathway – A. Computer Science

G. Computer Networks
- Same as Concentration Pathway – A. Computer Science

H. Computational Physics
- Any approved science course outside the Department of Physics and Atmospheric Science may satisfy Science A.
- PHYS 013 Physics for Engineers I and PHYS 023 Physics for Engineers Lab I - (Science B (1))
- PHYS 014 Physics for Engineers II and PHYS 024 Physics for Engineers Lab II - (Science B (2))

I. Cybersecurity
- Any Same as Concentration Pathway – A. Computer Science

TECHNICAL ELECTIVES

The objective of the technical elective requirement is to provide students with advanced study in areas that complement their academic and professional interests. Students pursuing a concentration pathway are encouraged to select technical electives aligned with their chosen pathway. Students are not required to pursue a concentration pathway and may instead select technical electives from the broader Computer Science Pathway.

The Concentration Pathway – A. Computer Science provides the greatest flexibility in technical elective selection.

A. Computer Science
- Any CSCI course numbered 300–499 (excluding required courses), with the exception of CSCI 200.
- Any CSCI graduate course open to seniors.
- EECE 212 Fundamentals of Digital Systems.
- EECE 406 Advanced Digital Systems.
- EECE 417 Computer Systems Architecture.
- EECE 416 Microcomputer Design.
- Any mathematics course not otherwise required and having Calculus I as a prerequisite.
- INFO 395 Information Assurance.
- INFO 393 Network/Internet Security Management.
- EGPP 493 Bison Startup: Technology Entrepreneurship and Lean Startups.
- EGPP 494 Bison Accelerate: Launch and Iterate.
- EGPP 405 Undergraduate Research.
- EGPP-421 Intro. to AI: Tools and Techniques
- EGPP-422 Ethical and Responsible AI
- CHEM 184 Methods in Computational Chemistry
- PHYS 154 Computational Physics I.
- PHYS 155 Computational Physics II
- Any course approved by the Department Chair.

B. Human-Centered Computing

- CSCI 379 Introduction to Human Computer Interaction.
- CSCI 380 Foundations of User Experience.
- CSCI 381 Affective Computing.
- Any technical elective approved for Concentration Pathway – A. Computer Science

C. Computational Biology

- BIOL 200 Genetics•.
- BIOL 320 Molecular Biology•.
- CSCI 474 Computational Biology.
- Any technical elective approved for Concentration Pathway – A. Computer Science

D. Computational Chemistry

- CHEM 001 General and Applied Chemistry•.
- CHEM 184 Methods in Computational Chemistry.
- MATH 159 Differential Equations or MATH 164 Introduction to Numerical Analysis.
- Any technical elective approved for Concentration Pathway – A. Computer Science

E. Gaming

- CSCI 421 Computer and Video Game Development.
- CSCI 422 Game Engine Programming.
- Any technical elective approved for Concentration Pathway – A. Computer Science

F. Computational Mathematics

- MATH 159 Differential Equations.
- MATH 164 Introduction to Numerical Analysis.
- MATH 189 Probability and Statistics
- Any technical elective approved for Concentration Pathway – A. Computer Science

G. Computer Networks

- CSCI 450 Introduction to Computer Networks.
- CSCI 451 Applied Wireless Networks.
- CSCI 452 Internet/Web Programming.
- CSCI 479 Cybersecurity Networks, CPS, and IoT.
- INFO 393 Network/Internet Security Management.
- Any technical elective approved for Concentration Pathway – A. Computer Science

H. Computational Physics

- PHYS 154 Computational Physics I.
- PHYS 155 Computational Physics II.
- MATH 159 Differential Equations or MATH 164 Introduction to Numerical Analysis.
- Any technical elective approved for Concentration Pathway – A. Computer Science

I. Cybersecurity

- INFO 395 Information Assurance.
- CSCI 454 Introduction to Cybersecurity II.
- INFO 393 Network/Internet Security Management.
- CSCI 479 Cybersecurity Networks, CPS, and IoT.
- Any technical elective approved for the Computer Science Pathway.

• Course must be registered with an override, as it is not listed on Approved List of Technical Electives

APPENDIX II: Registration

Howard University utilizes the BisonHub student information system to support registration and academic records. Students should access BisonHub through AccessHU using their Howard University credentials.

https://accesshu.howard.edu/

Students should follow the four-step registration process outlined below to prepare for advising and register for courses each semester.

STEP 1: Complete the Registration Checklist
Students must complete the Registration Checklist prior to meeting with their academic advisor for registration approval.

Registration Checklist Link:
Students must be logged into their Howard University account (Bison account) to access the Registration Checklist. The checklist may be accessed using the link or QR code provided below.

https://bit.ly/CS-CheckList-2022-2026

You will be forced to copy the CheckList to create your own. Once you create your checklist, save it to your Google Drive for future use. You will be updating your CheckList every semester. The CheckList is used to help you manage and document meeting the graduation requirements for the program for the Catalog year specified.

Students will be prompted to create a personal copy of the checklist in a shared folder (i.e. OneDrive or Google Drive) before completing it. The checklist serves as a planning and advising tool and should be updated each semester.

Students should update their Registration Checklist with the GRADES (i.e., “A,” “B,” “C,” “D,” “F,” and “T” for Transfer or Advanced Standing) and the semester/year completed for all courses taken during the previous semester. This ensures that the checklist remains accurate and up to date for advising.

For new first-time-in-college students, this may include adding "T" grades for approved AP, IB, CAPE, dual enrollment, transfer, or other advanced standing credits that have been accepted by the University.

Students are expected to maintain an updated version of the checklist and provide it to their advisor during advising appointments.

STEP 2: The Registration Process

Students should use the checklist to verify completed coursework and identify remaining degree requirements outlined in the CS Student Handbook for the applicable catalog year. It is the student's responsibility to ensure that all prerequisites, co-requisites, and graduation requirements are satisfied.

The graduation requirements include:

- Computer Science core requirements
- Mathematics and Science requirements
- Available Technical Electives
- Available Non-Technical Electives

Students should also use the checklist to develop a proposed schedule for the upcoming semester prior to meeting with their advisor. This can be done by entering the DATE, the course number (CRS#) (in science and elective courses), and GRADE -- select IP (In Process) into your CheckList.

STEP 3: Meet with your Academic Advisor

Advising is managed by the CEA Office of Student Success (OSS). The CS Program advisors are:

Mr. Ladan W. Johnson Room LKD 1114 Ladan_johnson1@howard.edu Last Names: **A - K**	**Ms. Bobbi Harewood** Room LKD 1012B Bobbi_harewood@howard.edu Last Names **L - Z**

It is strongly recommended that students schedule and attend an advising appointment with their assigned academic advisor prior to registering for courses in BisonHub.

Each semester, the Office of Student Services (OSS) coordinates a Registration Campaign to support students during the registration process. During advising appointments held as part of the campaign, academic advisors will:

- Review the student's academic progress and degree requirements
- Discuss proposed course selections for the upcoming semester
- Verify completion of prerequisites and co-requisites
- Ensure the student is making satisfactory progress toward graduation
- Provide guidance on academic planning, course sequencing, and academic success strategies

Students are expected to arrive prepared for advising appointments with an updated Registration Checklist and a proposed schedule of courses for the upcoming semester.

In addition to the official registration advising period, students are encouraged to meet with their academic advisor throughout the academic year as needed. Advising appointments outside of the Registration Campaign may be scheduled at a mutually convenient time for both the student and the advisor.

If you need to speak to someone regarding **career guidance**, or **Science and Technical Electives based on a Concentration Pathway**, you should schedule an appointment to meet with a Concentration Pathway Faculty Advisor located in APPENDIX I.

STEP 4 (If needed): Handling Registration Errors

Students may occasionally encounter registration error messages when attempting to enroll in courses through BisonHub. The Registration Error Form is used by students in the College of Engineering and Architecture (CEA) to request review and possible overrides for specific registration restrictions.

The form may be obtained from your academic advisor, and can be used to request overrides for issues including, but not limited to:

- Time conflicts
- Level restrictions
- Field of study or major restrictions
- Special approval requirements
- Prerequisite or co-requisite requirements
- Test score requirements

Requests for time conflict overrides require approval from the instructor(s) of the affected course(s), typically documented through a signature or written authorization on the form.

Submission of the form does not guarantee approval. All requests are subject to departmental and university policies, course availability, and final approval by the appropriate academic unit.

APPENDIX III: Variances for Coursework

Variance for Coursework

The curriculum and graduation requirements outlined in this handbook are designed to ensure that all students achieve the Student Outcomes of the degree program. Under limited circumstances, a student may request a variance to a specific course requirement when unique academic circumstances warrant consideration of an alternative course or academic experience.

A variance request is not automatically approved and must demonstrate that the proposed alternative provides substantially equivalent academic content and supports the educational objectives of the degree program.

Approval Process

All variance requests must be initiated by the student's academic advisor and submitted using the Department's **Variance in Graduation Requirements Form**. The completed form must be reviewed and approved through the following process:

1. **Student Request** – The student discusses the request with their academic advisor and provides any supporting documentation using the **Variance in Graduation Requirements Form**.
2. **Advisor Review and Recommendation** – The advisor evaluates the request and determines whether the proposed alternative is academically appropriate.
3. **Program Director or Department Chair Review** – Following advisor approval, the request is forwarded to the Program Director or Department Chair for secondary review and final approval.

No variance is considered approved until all required signatures have been obtained.

Required Documentation

The **student must provide** documentation using the **Variance in Graduation Requirements Form** that includes:

- The student's name and student identification number.
- A description of the standard graduation requirement.
- The course number, title, and catalog description of the required course(s), when applicable.
- A description of the proposed alternative course, academic experience, transfer credit, internship, research experience, or other substitute requirement.
- The course number, title, and catalog description of the proposed alternative, when applicable.
- Signatures from the student.

After receipt of the Variance Form, the **advisor must provide**:

- A written recommendation explaining the academic rationale for the variance request.
- Documentation demonstrating how the proposed alternative satisfies the learning objectives and educational intent of the original requirement.
- Signatures of approval.

After receipt of a signed Variance Form, the **Program Director of Department Chair must provide**:

- A written recommendation explaining the academic rationale for the variance request.
- Signatures of approval.

Circumstances for Consideration

Examples of circumstances that may warrant consideration of a variance include:

- Course discontinuation or significant curriculum changes.
- Transfer credit evaluations.
- Study abroad experiences.
- Equivalent coursework completed at another accredited institution.
- Other exceptional circumstances approved by the Department.

Record Retention

Once approved, please update the CheckList with the Required Course and Replacement Course information. Approved variances will be maintained in the student's departmental advising folder in OneDrive and will be used during graduation certification (i.e. degree audits) and accreditation reviews. Students should retain copies of all approved variance documentation for their records.

The Department reserves the right to deny requests that do not adequately demonstrate academic equivalency or that would compromise the integrity of the degree program.

Variance in Graduation Requirements Form
Computer Science Program
College of Engineering and Architecture
Howard University

Student Name: _____
Student ID Number: _____

Description of Standard Graduation Requirement
Provide the course number, title, catalog description, and graduation requirement being satisfied.

Description of Requested Alternative
Provide the proposed substitute course, transfer course, internship, research experience, or other alternative and explain how it satisfies the original requirement.

Student Signature: _____ Date: _____

Advisor Recommendation
Provide the academic rationale for the variance request and explain why the proposed alternative is considered equivalent to the standard requirement.

Advisor Signature: _____ Date: _____

Program Director/Chair Recommendation

Program Director/Chair Signature: _____ Date: _____