COMPUTER SCIENCE

Office: Department of Computer Science, Suite 379
Mail Code: Ritchie School of Engineering & Computer Science
2155 E. Wesley Avenue, Denver, CO 80208
Phone: 303-871-3192
Email: info@cs.du.edu
Web Site: http://www.du.edu/rsecs/departments/cs (http://www.du.edu/rsecs/departments/cs/)

Computer Science affects every aspect of society and the world: in the apps we use on devices and on the web, in artificial intelligence that is adding sophisticated intelligence and reasoning to applications, in data science that is helping us understand the troves of data being generated in our word, in our entertainment through games, videos and movies, in the way we advance science and mathematics and design products in engineering, and in the way we vote, bank, and run businesses. The Department of Computer Science offers five undergraduate programs with two concentrations and and teaches students a solid theoretical foundation of computer science while allowing students to specialize in various areas through electives. Graduates of the program have found employment in all areas of the computer software industry and in the application of computer science in a wide variety of companies and fields. Our graduates work at companies including IBM, Microsoft, Google, Amazon, Oracle, LinkedIn, Meta, Twitter, Raytheon, and Lockheed Martin. Graduates of these programs also are prepared to pursue graduate school in computer science or other related fields. The department offers majors in computer science, game development and applied computing, with concentrations in cybersecurity and artificial intelligence.

Computer science encompasses the theory and techniques by which information is encoded, stored, communicated, transformed and analyzed. It is concerned with the theory of algorithms (that is, effective procedures or programs), with the structure of languages for the expression of algorithms and with the design of algorithms for the solution of practical problems. A central concern in the discipline is problem solving and understanding how to think abstractly for solving large problems and to be able to understand the cost of computation in terms of time and space in solving such problems. Given the incredible amounts of data produced today, an increasing focus in computer science is towards using data to develop algorithms in machine learning and artificial intelligence to analyze and solve new, previously difficult problems.
Computer science students have the ability to take courses in the department in a broad spectrum of topics, including, but not limited to, programming, networking, cybersecurity, privacy, human-computer interaction, game development and design, robotics and human-robot interactions, algorithms, web development, data science, and artificial intelligence.

The Department of Computer Science at the University of Denver offers a bachelor of science in computer science and has strong mathematical foundations that include completing a math cognate, which may be used towards completing a math minor. In addition, the department offers two concentrations with the bachelor of science in computer science in artificial intelligence and cybersecurity.

Computer Science

Bachelor of Science in Computer Science Major Requirements

(183 credits required for the degree (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofscience/))

The Bachelor of Science in Computer Science prepares students for jobs in highly technical fields or related disciplines where computing is used and needed. Students learn how to problem solve, develop software, analyze the space and time performance of data structures and algorithms, work with and comprehend low-level systems, and apply modern programming techniques. Electives in the major allow them to expand their knowledge of computer science in areas they're interested in pursuing but ensure students receive some breadth of knowledge in the discipline.

59 credits of COMP courses are required by the Bachelor of Science in Computer Science requires that are broadly classified as **Core** courses and **Breadth** courses. The list below lists all the **Core** classes required, totaling 35 credits. Students are also required to satisfy 24 credits of 3000-level COMP electives through the **Breadth** requirement: 4 credits must be taken from courses that are classified into the following 4 categories: 1) ethics, diversity, equity, justice, inclusivity; 2) systems; 3) theory; 4) programming languages. This totals 16 credits and the remaining 8 credits *may* have one of these classifications.

Students must also satisfy a Math cognate consisting of 12 Math credits at Calculus I (MATH 1951) or higher. These Math courses may be used to satisfy a Math minor. Please refer to the Department of Mathematics (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/mathematics/) for details about the Math minor.

Code	Title	Credits
Core Courses		
COMP 1201 & COMP 1202 & COMP 1203	Introduction to Computer Science I and Introduction to Computer Science II and Introduction to Computer Science III (Taken in parallel with COMP 1351, COMP 1352, and COMP 1353) [1]	6

COMP 1351 & COMP 1352 & COMP 1353	Introduction to Programming I and Introduction to Programming II and Introduction to Data Structures & Algorithms I	9
COMP 2300	Discrete Structures in Computer Science [2]	4
COMP 2370	Introduction to Data Structures & Algorithms II	4
COMP 2361	Systems I	4
COMP 2362	Systems II	4
COMP 2381	Object-Oriented Software Development	4
Breadth Courses [3, 4]		
Complete 24 credits of 3000-level computer science courses, where at least one course must satisfy each of the categories: ethics/DEI, systems, theory, and programming languages.		24
Total Credits		**59**

1 These courses are typically taken in parallel with Introduction to Programming I (COMP 1351), Introduction to Programming II (COMP 1352), and Introduction to Data Structures & Algorithms I (COMP 1353).

2 If used towards the Mathematics minor, 4 additional COMP credits must be taken as this course cannot count both towards the Mathematics minor and the total COMP credits required by the degree.

3 Internship/Co-Op in Computing (COMP 3904) may not be used to satisfy 3000-level computer science breadth credits.

4 Electives in the Programming Language category: Programming Languages (COMP3351), Compiler Design (COMP3352), Compiler Construction (COMP3353)
Electives in the Systems category: Operating Systems (COMP3361), Networking (COMP3621), Game Networking (COMP3681), Topics in Systems (COMP3704), Computer Security (COMP3721), Network Security (COMP3722), Ethical Hacking (COMP3723), Computer Forensics (COMP3731)
Electives in the Theory category: DataStructures&Algorithms (COMP3371), Advanced Algorithms (COMP3372), Introduction to AI (COMP3501), Computational Geometry (COMP3591)
Electives in the Ethics/DEI category: Human Computer Interaction (COMP3100), Human Data Security and Privacy (COMP3732),
Other special topics may potentially count in a category. Please see your advisor for more information on special topic classes.

Additional Requirements

Students must also satisfy the Math cognate requirements:

Code	Title	Credits
Math Cognate		
12 additional credits of Math electives at MATH 1951 or higher. [1]		12

1 Courses satisfying the Math cognate may also be used to satisfy the Mathematics minor.

Concentration in Artificial Intelligence

Artificial Intelligence (AI) is a broad sub-field of Computer Science including search, planning, natural language processing, computer vision, speech recognition, optimization, knowledge representation, machine learning, and neural networks to name a few of the major areas in AI. This AI Concentration takes students from the basic mathematical foundations to theoretical computer science underpinnings of artificial intelligence to practical applications using AI, enabling students to understand, design and develop modern AI for solving real-world problems.

The Artificial Intelligence Concentration is a concentration that can be taken with the Bachelor of Science degrees offered by the Computer Science Department. Students enrolled in a Bachelor of Science in Computer Science or Game Development will use the required classes as part of their Math Cognate and the AI required courses as electives in their degrees.

The Artificial Intelligence Concentration has the following courses:

Code	Title	Credits
Math Required Courses (2 courses)		
MATH 2060	Elements of Linear Algebra	
MATH 3080	Introduction to Probability	
Computer Science Required Courses (2 courses):		
COMP 3501	Introduction to Artificial Intelligence	4
COMP 3703	Topics-Artificial Intelligence (Foundations of Machine Learning)	4
Computer Science Electives (3 from the following) [1]		**12**
COMP 3432	Machine Learning	
ENCE 3620	Computer Vision	

COMP 3510	Software for AI Robotics
COMP 3703	Topics-Artificial Intelligence (Mind-Reading Machines)
COMP 3703	Topics-Artificial Intelligence (Large Language Models)
COMP 3703	Topics-Artificial Intelligence (Natural Language Processing)
Total Credits	**20**

Concentration in Cybersecurity

The cybersecurity concentration is a specific track as part of the Bachelor of Science in Computer Science Degree, applicable for students who have an interest in computer security (cybersecurity) and/or would like to have experience in the field. A concentration indicates a study focus that is reflected on transcripts and the degree itself (similar to a minor, but not significantly different enough from the major itself to be an actual minor).

Students can attain the cybersecurity concentration by completing at least 20 credits of the listed cybersecurity specific courses offered by the computer science department (or those approved by the major advisor, cybersecurity program director, or department chair).

The cybersecurity concentration is made up of the following required courses. They can be completed in any order, though normal prerequisite requirements apply.

Code	Title	Credits
Required Courses		
COMP 3621	Computer Networking	4
COMP 3721	Computer Security	4
COMP 3722	Network Security	4
COMP 3723	Ethical Hacking	4
One of the Following:		**4**
COMP 3384	Secure Software Engineering	
COMP 3731	Digital Forensics	
COMP 3732	Human-Centered Data Security and Privacy	
Total Credits		**20**

Bachelor of Arts in Computer Science Major Requirements

The Bachelor of Arts in Computer Science prepares a student for a job in a highly technical field or related discipline where computing is needed. Students learn how to problem solve, develop software, analyze the space and time performance of data structures and algorithms, work with and comprehend low-level systems, and apply modern programming techniques. Electives in the major allow them to expand their knowledge of computer science in areas they're interested in pursuing but ensure students receive some breadth of knowledge in the discipline. This program is similar to the Bachelor of Science in Computer Science, except that it does not require the Math cognate.

59 credits of COMP courses are required in the Bachelor of Arts in Computer Science that are broadly classified as **Core** courses and **Breadth** courses. The table below lists all the **Core** classes required, totaling 35 credits. Students are also required to satisfy 24 credits of 3000-level COMP electives through the **Breadth** requirement: 4 credits must be taken from courses that are classified into the following 4 categories: 1) ethics, diversity, equity, justice, inclusivity; 2) systems; 3) theory; 4) programming languages. This totals 16 credits and the remaining 8 credits *may* have one of these classifications.

Code	Title	Credits
Core Courses		
COMP 1201 & COMP 1202 & COMP 1203	Introduction to Computer Science I and Introduction to Computer Science II and Introduction to Computer Science III	6
COMP 1351 & COMP 1352 & COMP 1353	Introduction to Programming I and Introduction to Programming II and Introduction to Data Structures & Algorithms I	9
COMP 2300	Discrete Structures in Computer Science	4
COMP 2370	Introduction to Data Structures & Algorithms II	4
COMP 2361	Systems I	4
COMP 2362	Systems II	4
COMP 2381	Object-Oriented Software Development	4
Breadth Courses [2, 3]		

Complete 24 credits of 3000-level computer science courses, where at least one course must satisfy each of the categories: ethics/DEI, systems, theory, and programming languages. [1,2]	24
Total Credits	**59**

1. Internship/Co-Op in Computing (COMP 3904) may not be used to satisfy 3000-level computer science elective credits.
2. Computer Science courses may be designated as covering topics in theory, systems, programming languages, and ethics/DEI (diversity, equity, inclusivity). One course must be taken from each of these 4 categories, and the other courses may be from any (or no) category.
3. Electives in the Programming Language category: Programming Languages (COMP3351), Compiler Design (COMP3352), Compiler Construction (COMP3353)
   Electives in the Systems category: Operating Systems (COMP3361), Networking (COMP3621), Game Networking (COMP3381), Topics in Systems (COMP3704), Computer Security (COMP3721), Network Security (COMP3722), Ethical Hacking (COMP3723), Computer Forensics (COMP3731)
   Electives in the Theory category: DataStructures&Algorithms (COMP3371), Advanced Algorithms (COMP3372), Introduction to AI (COMP3501), Computational Geometry (COMP3591), Topics in AI (COMP3703)
   Electives in the Ethics/DEI category: Human Computer Interaction (COMP3100), Human Data Security and Privacy (COMP3732)
   Other special topics may potentially count in a category. Please see your advisor for more information on special topic classes.

Minor in Computer Science Requirements

A minor in Computer Science requires a total of 21 credits. The goal of the minor is to provide students with a foundation in computer programming and a basic understanding of the discipline, while allowing some flexibility in elective courses to compliment their interests. Note that the Minor in Computer Science may also be used to satisfy the Computer Science course requirements of the Computational Physics Concentration (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/physicsandastronomy/#programofstudytext). The following courses are required for the Minor in Computer Science:

Code	Title	Credits
Required Courses		
COMP 1201	Introduction to Computer Science I	2
COMP 1202	Introduction to Computer Science II	2
COMP 1351	Introduction to Programming I	3
COMP 1352	Introduction to Programming II	3
COMP 1353	Introduction to Data Structures & Algorithms I	3
Electives		
8 credits of 2000-level or higher COMP courses [1]		8
Total Credits		**21**

1. COMP 3904 Internship/Co-Op in Computing cannot be used to satisfy the elective requirements of the minor.

Applied Computing

The Bachelor of Arts in Applied Computing (BA in AC) provides a quality education for a serious computer user. It complements the department's Bachelor of Science in Computer Science by providing a program that combines collaboration with other departments and an applications-oriented emphasis. A prospective BA in AC major must satisfy all the requirements for a Bachelor of Arts (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofarts/) degree as outlined in the University Undergraduate Bulletin.

The BA in AC is a suitable degree for many collaborative programs within the University and would be ideal for students in graphic arts and electronic publishing or would provide an appropriate foundation for a student who wishes to pursue a career in the field of educational technology. A graduate with a BA in AC would be very attractive in the data processing unit of large financial, banking, or insurance institutions, as a network or system administrator or as a World Wide Web designer/programmer. Holders of the BA in AC degree would also be well suited to continue in any number of specialized Master's or certificate programs in fields as diverse as video an graphics production, fashion design, telecommunications, instructional technology, or management information systems.

Bachelor of Arts in Applied Computing Major Requirements

(183 credits required for the degree (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofarts/))

The Bachelor of Arts in Applied Computing prepares students for a job in a technical field or a discipline where computing is used as a tool for problem solving. Students learn how to problem solve, develop software, and apply computer science techniques to any discipline. The flexibility in this program allows students to more easily double-major in another area of their interest to understand better how to apply computing principles to that field.

This major requires 47 credits of computer science and other approved computer applications and mathematics courses. Students are required to take 19 credits in core computer science courses and 4 credits in courses that contain topics in ethics, diversity, equity, inclusivity, and social justice.

Majors pick a computer science stream to focus on and will complete 12 credits in that stream. In addition, majors will select a cognate in an allied field where they will complete 12 credits of computational-related courses (note, these cannot be used for the major also).

Per University requirements (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofarts/), no more than 60 credit hours in any one department can be applied towards a Bachelor of Arts (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofarts/) degree. Required courses are as follows:

Code	Title	Credits
Core Courses		
COMP 1201 & COMP 1202 & COMP 1203	Introduction to Computer Science I and Introduction to Computer Science II and Introduction to Computer Science III	6
COMP 1351 & COMP 1352 & COMP 1353	Introduction to Programming I and Introduction to Programming II and Introduction to Data Structures & Algorithms I	9
COMP 2381	Object-Oriented Software Development	4
Ethics, Diversity, Equity, Justice, Inclusivity Elective		
4 credit COMP elective designated as ethics, diversity, equity, justice, inclusivity		4
Stream Courses		
12 credits of chosen computer science stream		12
Cognate Courses		
12 credits of chosen cognate courses or COMP electives		12
Total Credits		**47**

Computer Science Streams for the Bachelor of Arts in Applied Computing

Students must complete 12 credits in one of the following streams. Computing streams not listed may be approved by the department and advisor, but must consist of COMP courses. The following streams are pre-approved but may not list all possible courses that apply towards a stream, and in particular, computer science topics courses may be used to fulfill the requirements of a stream (e.g., Topics in Computer Science (COMP 3705)).

- **Data:** Database Organization & Management I (COMP 3421), NoSQL Databases (COMP 3424), Data Visualization (COMP 3433)
- **Web Development:** World Wide Web Programming (COMP 3410), Web Programming II (COMP 3411), one or more of { Web Projects: Web Development III (COMP 3412), Database Organization & Management I (COMP 3421), NoSQL Databases (COMP 3424) }
- **Systems:** Systems I (COMP 2361), Systems II (COMP 2362), one or more of { Operating Systems I (COMP 3361), Computer Networking (COMP 3621), Computer Security (COMP 3721), Network Security (COMP 3722), COMP 3723 Ethical Hacking, COMP 3731 Computer Forensics }

Cognate in Allied Fields

Students must also complete 12 credits that are computational and/or data oriented in nature in one of the approved allied fields. These should be chosen with and approved by their advisor and can include continued coursework in Computer Science. Courses used to satisfy this requirement, may not also be used to satisfy other major or minor requirements. The allied disciplines include, but are not limited to:

- Emergent Digital Practices (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/emergentdigitalpractices/)
- Geographic Information Science (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/geographyandtheenvironment/#programofstudytext)
- Mathematics (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/mathematics/)
- Accounting (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/accounting/), Finance (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/finance/), Business Information and Analytics (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/businessintelligenceandanalytics/), or Innovation and Entrepreneurship (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/innovationandentrepreneurship/)
- Media, Film, and Journalism Studies (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/mediafilmampjournalismstudies/)
- Social Sciences
- Natural Sciences
- Arts, Humanities, and Music (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/mediafilmampjournalismstudies/)

- Computer Science (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/computerscience/)

Game Development

The game development programs are a joint effort by the University of Denver Computer Science, Emergent Digital Practices and Art programs. Specific degrees offered are

- Bachelor of Arts with a major in Game Development, and
- Bachelor of Science with a major in Game Development.

Graduates of these programs not only have solid computer science, graphics and game-programming skills, but also a strong foundation in art and/or the critical, technical and design aspects of digital media. To obtain this foundation, the major requires work in allied fields. Allied areas include Studio Art and Emergent Digital Practices. The depth of allied knowledge is dependent on the degree (BA or BS) chosen.

Bachelor of Arts in Game Development Major Requirements

(183 credits required for the degree (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofarts/))

The Bachelor of Arts in Game Development is a degree which prepares students to be capable of creating artistic content and designing game play for games while having a strong technical background, enabling them to bridge the gap between artist, programmer and designer. The BA in Game Development requires a Minor in Emergent Digital Practices (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/emergentdigitalpractices/). A graduate of this program will be able to study and work as a developer, game designer and an artist. The BA requires more courses in the allied art fields than the BS and is balanced by having fewer required Math and Computer Science courses.

59 credits, including the following:

Code	Title	Credits
Required Courses		
COMP 1201 & COMP 1202 & COMP 1203	Introduction to Computer Science I and Introduction to Computer Science II and Introduction to Computer Science III [1]	6
COMP 1351 & COMP 1352 & COMP 1353	Introduction to Programming I and Introduction to Programming II and Introduction to Data Structures & Algorithms I	9
COMP 2300	Discrete Structures in Computer Science	4
COMP 2370	Introduction to Data Structures & Algorithms II	4
COMP 2381	Object-Oriented Software Development	4
COMP 2821	Introductory Game Design	4
COMP 3820 & COMP 3821	Introduction to Game Programming and Game Programming I	8
COMP 3831 & COMP 3832 & COMP 3833	Game Prototyping and Game Capstone I and Game Capstone II	12
ARTS 1250	Drawing	4
EDPX 3600	3D Modeling	4
Total Credits		**59**

[1] These courses are usually taken in sequence with Introduction to Programming I (COMP 1351), Introduction to Programming II (COMP 1352), and Introduction to Data Structures & Algorithms I (COMP 1353).

Bachelor of Science in Game Development Major Requirements

(183 credits required for the degree (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofscience/))

The Bachelor of Science in Game Development is a combination of a Computer Science degree, specific courses in Game Development, and requires a cognate in Mathematics, a second minor of your choice, and a cognate of five approved classes from Art (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/schoolofartandarthistory/), Emergent Digital Practices (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/emergentdigitalpractices/), Creative

Writing (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/english/), or Music (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/music/). In addition, both degrees require satisfying the University Common Curriculum requirements for the BS degree.

This program provides a strong computer science and technical background, preparing students for all aspects of game programming, while providing them with a foundation in art, emergent digital practices, creative writing, or music. Thus, graduates of this program are able to help in the programming and development of games, while understanding and being able to communicate effectively with the artists, designers, writers and musicians who are part of any game development project. The BS requires more mathematics and more required COMP classes than the BA and is balanced with fewer classes in the allied fields.

Courses in this program are broadly classified as **Core**, **Breadth**, and **Cognate** courses. **Core** and **Breadth** courses total 63 credits in Computer Science. Cognate Courses are divided between the Math Cognate at 12 credits and the Art Cognate at 20 credits. Both cognates can be used, and usually are, to satisfy the courses and credits of their respective minors.

Code	Title	Credits
Core Courses		
COMP 1201 & COMP 1202 & COMP 1203	Introduction to Computer Science I and Introduction to Computer Science II and Introduction to Computer Science III [1]	6
COMP 1351 & COMP 1352 & COMP 1353	Introduction to Programming I and Introduction to Programming II and Introduction to Data Structures & Algorithms I	9
COMP 2300	Discrete Structures in Computer Science	4
COMP 2370	Introduction to Data Structures & Algorithms II	4
COMP 2361 & COMP 2362	Systems I and Systems II	8
COMP 2381	Object-Oriented Software Development	4
COMP 2821	Introductory Game Design	4
COMP 3820 & COMP 3821 & COMP 3822	Introduction to Game Programming and Game Programming I and Game Programming II	12
COMP 3831 & COMP 3832 & COMP 3833	Game Prototyping and Game Capstone I and Game Capstone II	12
Total Credits		**63**

1 These courses are usually taken in sequence with Introduction to Programming I (COMP 1351), Introduction to Programming II (COMP 1352), and COMP 1353 Introduction to Data Structures & Algorithms I.

Additional Requirements

Students must complete both a Math and Art cognate:

Code	Title	Credits
Math Cognate		
12 additional credits of Math electives at MATH 1951 or higher. [1]		12
Art Cognate		
20 credits from art, emergent digital practices, creative writing, or music. [2]		20

1 These cognate credits may also be used to satisfy a Math minor.
2 These cognate courses may be used to satisfy credits in a minor in one of these fields.

Bachelor of Science and Bachelor of Arts in Computer Science

The following is a typical plan of study for a BS or BA in Computer Science major. BA students do not need to complete the Math cognate requirement and can replace these with Common Curriculum or other major/minor courses as needed. Flexibility in the junior and senior years allows students to study abroad in any quarter fairly easily.

First Year

Fall	Credits	Winter	Credits	Spring	Credits
FSEM 1111	4	WRIT 1122	4	WRIT 1133	4
COMP 1201	2	COMP 1202	2	COMP 1203	2
COMP 1351	3	COMP 1352	3	COMP 1353	3
Math cognate course	4	Math cognate course	4	Math cognate course	4
Common Curriculum Course [1]	4	Common Curriculum Course	4	Common Curriculum Course	4
	17		**17**		**17**

Second Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 2300	4	COMP 2370	4	COMP 2361	4
COMP 2381	4	Common Curriculum Course	4	COMP 3000-level elective	4
Common Curriculum Course	4	Common Curriculum Course	4	1st minor course[2]	4
Common Curriculum Course	4	Common Curriculum Course	4	Common Curriculum Course	4
	16		**16**		**16**

Third Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 2362	4	COMP 3000-level elective	4	COMP 3000-level elective	4
COMP 3000-level elective	4	1st minor course	4	1st minor course	4
1st minor course	4	2nd minor course	4	2nd minor course	4
2nd minor course[2]	4	Common Curriculum Course	4	Common Curriculum Course	4
	16		**16**		**16**

Fourth Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 3000-level elective	4	COMP 3000-level elective	4	2nd minor course	4
1st minor course	4	2nd minor course	4	Elective	4
2nd minor course	4	Elective	4	Elective	4
	12		**12**		**12**

Total Credits: 183

Bachelor of Arts in Applied Computing

The following is a typical plan of study for BA in Applied Computing majors. Note that flexibility in the schedule allows students to study abroad and easily double-major in another major or even double minor.

First Year

Fall	Credits	Winter	Credits	Spring	Credits
FSEM 1111	4	WRIT 1122	4	WRIT 1133	4
COMP 1201	2	COMP 1202	2	COMP 1203	2
COMP 1351	3	COMP 1352	3	COMP 1353	3
Common Curriculum Course[1]	4	Common Curriculum Course	4	Common Curriculum Course	4
Common Curriculum Course	4	Common Curriculum Course	4	Common Curriculum Course	4
	17		**17**		**17**

Second Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 2381	4	COMP Stream course	4	COMP Stream course	4
Common Curriculum Course	4	Common Curriculum Course	4	COMP Stream course	4
Minor Course	4	Minor Course	4	Common Curriculum Course	4
Elective	4	Elective	4	Minor Course	4
	16		**16**		**16**

Third Year

Fall	Credits	Winter	Credits	Spring	Credits
Cognate course	4	Cognate course	4	Cognate course	4
Common Curriculum Course	4	Common Curriculum Course	4	Minor Course	4
Minor Course	4	Minor Course	4	Elective	4
Elective	4	Elective	4	Elective	4
	16		**16**		**16**

Fourth Year

Fall	Credits	Winter	Credits	Spring	Credits
Elective	4	Elective	4	Elective	4
Elective	4	Elective	4	Elective	4
Elective	4	Elective	4	Elective	4
	12		**12**		**12**

Total Credits: 183

Bachelor of Science in Game Development

The following is a typical plan of study for BS in Game Development majors. Generally students who study abroad do this in their junior year of the major. Note that courses listed as Minor are typically combined with the required Art Cognate.

First Year

Fall	Credits	Winter	Credits	Spring	Credits
FSEM 1111	4	WRIT 1122	4	WRIT 1133	4
COMP 1201	2	COMP 1202	2	COMP 1203	2
COMP 1351	3	COMP 1352	3	COMP 1353	3
Math cognate	4	Math cognate	4	Math cognate	4
Common Curriculum Course [1]	4	Common Curriculum Course	4	Common Curriculum Course	4
	17		**17**		**17**

Second Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 2300	4	COMP 2370	4	COMP 2361	4
COMP 2381	4	COMP 3820	4	COMP 3821	4
COMP 2821	4	Common Curriculum Course	4	Minor Course	4
Common Curriculum Course	4	Minor course	4	Common Curriculum Course	4
	16		**16**		**16**

Third Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 2362	4	COMP 3822	4	Art Cognate Course	4
Art Cognate Course[3]	4	Art Cognate course	4	Minor Course	4
Common Curriculum Course	4	Minor Course	4	Common Curriculum Course	4
Common Curriculum Course	4	Minor Course	4	Common Curriculum Course	4
	16		**16**		**16**

Fourth Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 3831	4	COMP 3832	4	COMP 3833	4
Art Cognate Course	4	Art Cognate Course	4	Minor Course	4
Common Curriculum Course	4	Minor Course	4	Minor Course	4
	12		**12**		**12**

Total Credits: 183

Bachelor of Arts in Game Development

The following is a typical plan of study for BA in Game Development majors. Students have the flexibility to study abroad typically in their junior year and have significant flexibility to either double major in Emergent Digital Practices (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/emergentdigitalpractices/) (instead of just the minor) or take a second minor.

First Year

Fall	Credits	Winter	Credits	Spring	Credits
FSEM 1111	4	WRIT 1122	4	WRIT 1133	4
COMP 1201	2	COMP 1202	2	COMP 1203	2
COMP 1351	3	COMP 1352	3	COMP 1353	3
Common Curriculum Course[1]	4	EDPX Minor Course[4]	4	EDPX Minor Course	4
Common Curriculum Course	4	Common Curriculum Course	4	Common Curriculum Course	4
	17		**17**		**17**

Second Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 2300	4	COMP 2370	4	COMP 3821	4
COMP 2381	4	COMP 3820	4	EDPX Minor Course	4
ARTS 1250	4	Common Curriculum Course	4	Common Curriculum Course	4
COMP 2821	4	Common Curriculum Course	4	Common Curriculum Course	4
	16		**16**		**16**

Third Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP Elective	4	COMP Elective	4	EDPX 3600	4
EDPX Minor Elective	4	EDPX Minor Elective	4	EDPX Minor Elective	4
Common Curriculum Course	4	Common Curriculum Course	4	Elective	4
Common Curriculum Course	4	Elective	4	Elective	4
	16		**16**		**16**

Fourth Year

Fall	Credits	Winter	Credits	Spring	Credits
COMP 3831	4	COMP 3832	4	COMP 3833	4
Elective	4	Elective	4	Elective	4
Elective	4	Elective	4	Elective	4
	12		**12**		**12**

Total Credits: 183

1 Please visit Common Curriculum Courses (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/degreesanddegreerequirements/) for details.

2 Please visit the Bachelor of Science (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/bachelorofscience/) for details.

3 Please visit the Bachelor of Science in Game Development (http://bulletin.du.edu/undergraduate/undergraduateprograms/traditionalbachelorsprogram/degreesanddegreerequirements/) for details.

4 Please visit the Emergent Digital Practices Minor (http://bulletin.du.edu/undergraduate/majorsminorscoursedescriptions/traditionalbachelorsprogrammajorandminors/emergentdigitalpractices/#programofstudytext) for details.

Requirements for Distinction in the Major in Computer Science, Game Development, or Applied Computing

Distinction in any of the majors granted through the Department of Computer Science require the following:

- Minimum 3.3 cumulative GPA in major courses
- Research project including thesis and presentation

Research Project

The undergraduate research project is an advanced project conducted with a faculty advisor in the Computer Science Department that a student may spend multiple quarters working on. The recommendation is that students find a faculty advisor in their junior year so that they may begin planning out the project which could be carried out over the summer and in their senior year. While a student may use work from a Independent Research (COMP 3995) or Independent Study (COMP 3991) as the basis for their distinction project, the scope of the distinction project must be larger than what was accomplished in those courses. The additional work required to complete the distinction project cannot have credits awarded for it; however, it may be supported by a PinS grant (https://career.du.edu/resources/partners-in-scholarship-pins/) so long as the scope is sufficient. The determination of what additional work is sufficient to complete the distinction project is left to the faculty advisor overseeing the project.

Thesis

In addition to the work carried out to complete the project, students are required to write an undergraduate thesis describing the related work, methodology used, and their results. While it is not a requirement that the undergraduate thesis is published, it is expected that the work is of publishable quality.

The thesis paper should follow the University of Denver Thesis and Dissertation Formatting Guidelines (https://www.du.edu/graduate-education/current-student-resources/dissertation-thesis-information/) specified by the Office of Graduate Education.

Presentation or Defense

Once the thesis is complete, students must present their work publicly. Students must complete at least one of the following public presentations, chosen in collaboration with their advisor:

1. **Formal thesis defense.** If a student chooses to defend their work, they must give a presentation in front of a committee, which consists of the faculty advisor and at least one other faculty member of the Computer Science Department. The student shall choose the committee with the input of the faculty advisor. The committee shall then decide whether the student will pass the defense.

2. **Presentation at Computer Science Research Day,** an annual event held by the DU CS department, typically held in mid-late May. The option to present as part of Computer Science Research Day is subject to meeting any scheduling deadlines set by that year's Research Day committee and the distinction advisor.
3. **Presentation at Student Signature Work Showcase (https://academicaffairs.du.edu/student-research/annual-research-showcase/),** a campus-wide annual event open to students across DU, typically held in mid-late May. The option to present as part of the Student Signature Work Showcase is subject to meeting any scheduling deadlines set by that year's Signature Work Showcase Committee and the distinction advisor.
4. **Another public presentation** in a format and venue agreed upon with the faculty advisor. For instance, this could include submitting and presenting a paper at an academic conference, presenting at a community workshop that the work is relevant to, or other similar formats. As above, this option is subject to meeting any scheduling deadlines set by the venue and the distinction advisor.

Limitations

The undergraduate distinction research project may not be used to fulfill the requirements of any other University of Denver degree or award, nor may any other University of Denver degree or award be used to fulfill the requirements of the undergraduate distinction research project.

Computer Science

COMP 1101 Analytical Inquiry I (4 Credits)
Students explore the use of mathematics and computer programming in creating animations. Students create animations on their laptop computers using animation software. This course counts toward the Analytical Inquiry: The Natural and Physical World requirement.

COMP 1201 Introduction to Computer Science I (2 Credits)
This course introduces the discipline of computer science and how it applies the natural and physical world and society. Topics include the history of computing, computer hardware components, the internet, ethics, and uses computation as a means to analyze, process, model, and understand our world. This course counts toward the Analytical Inquiry: The Natural and Physical World requirement. Ideally taken concurrently with COMP 1351.

COMP 1202 Introduction to Computer Science II (2 Credits)
This course continues the introduction of the discipline of computer science by exploring major areas within it. Topics covered include examples from data structures, algorithms, databases, programming languages, parallel computing, artificial intelligence, robotics, cyber-security, data science, gaming, and ethics. This course counts toward the Analytical Inquiry: The Natural and Physical World requirement. Prerequisite: COMP 1201.

COMP 1203 Introduction to Computer Science III (2 Credits)
This course introduces testing and software development in computer science. Topics covered include using debuggers, version control systems, unit testing and general testing, Unified Modeling Language (UML), computing ethics, and software development in a team setting. Prerequisite: COMP 1202.

COMP 1351 Introduction to Programming I (3 Credits)
This course is an introduction to fundamental aspects of computer programming. Topics covered include variables, conditional statements, iteration, functions, basic data structures, objects, file input/output and interactions. Satisfies 3 credits of Analytical Inquiry: Natural and Physical World.

COMP 1352 Introduction to Programming II (3 Credits)
This course continues to introduce more advanced programming topics using the Python programming language. Topics include classes, types, inheritance, methods/functions, testing, graphical-user interfaces, threads, data manipulation, functional programming, and recursion. This course counts toward the Analytical Inquiry: The Natural and Physical World requirement. Prerequisite: COMP 1351.

COMP 1353 Introduction to Data Structures & Algorithms I (3 Credits)
This course introduces data structures used in computation, including their behavior, usage, implementation, and the analysis of their space usage. In addition, the algorithms used for access, manipulation, and updating the data structures is covered. Data structures and algorithms addressed include contiguous and linked lists, stacks, queues, hash tables, heaps, trees, self-balancing trees, graphs, and graph traversal. Introductory runtime analysis is used to prove time and space requirements for data structures and their performance while being used. Prerequisite: COMP 1352.

COMP 1601 Computer Science Pathways (1 Credit)
This course is designed to help first year computer science and game development students succeed in a very challenging major. Topics and activities may include academic success strategies; personal inventory exercises; interviewing computer science alumni; exploring ethical concerns within the profession; seminars by industry and academic experts; establishing the relationships between computing and other disciplines; critical and creative thinking activities; disseminating information on the dual degree programs, the honors program requirements, the honor code, and computer science department program structures; and readings from and discussions about computing related articles and publications.

COMP 1671 Introduction to Computer Science I (4 Credits)
Characteristics of modern computers and their applications; analysis and solution of problems; structure programming techniques; introduction to classes, abstract data types and object-oriented programming. This course counts toward the Analytical Inquiry: The Natural and Physical World requirement. Prerequisite: high school algebra.

COMP 1672 Introduction to Computer Science II (4 Credits)
Advanced programming techniques; arrays, recursion, dynamic data structures, algorithm abstraction, object-oriented programming including inheritance and virtual functions. Prerequisite: COMP 1671.

COMP 1991 Independent Study (1 Credit)

COMP 2300 Discrete Structures in Computer Science (4 Credits)
Number systems and basic number theory, propositional and predicate logic, proof techniques, mathematical induction, sets, counting and discrete probability, case studies with applications from computer science, such as data representation, algorithm analysis and correctness, and system design. Prerequisites: COMP 1672 or COMP 1352.

COMP 2355 Intro to Systems Programming (4 Credits)
The prerequisites for this class are a good understanding of imperative and object-oriented programming in Java. The prerequisites for this class include a good understanding of basic programming constructs, such as branches (if, switch), loops (for, while, do), exceptions (throw, catch), functions, objects, classes, packages, primitive types (int, float, boolean), arrays, arithmetic expressions and boolean operations. Computer organization is a parallel prerequisite; if possible, students should register for both this course and COMP 2691. You must have a good understanding of basic data structures such as arrays, lists, sets, trees, graphs and hash-tables. This is a class on systems programming with focus on the C programming language and UNIX APIs. There will be programming assignments designed to make you use various Debian GNU/Linux system APIs. Programming assignments involve writing code in C or C++. Prerequisite: COMP 2673.

COMP 2361 Systems I (4 Credits)
This course introduces low-level computer systems and programming. Topics covered include Linux, the C programming language, pointers, dynamic memory management, number systems, instruction set architectures, debugging, and caching. Prerequisites: COMP 1353.

COMP 2362 Systems II (4 Credits)
This course introduces computer operating systems and parallel programming. Topics covered include processes, process forks, threads, race conditions, synchronization, scheduling, memory systems, resource sharing, and sockets. Prerequisite: COMP 2361.

COMP 2370 Introduction to Data Structures & Algorithms II (4 Credits)
This course introduces the performance analysis of algorithms, including proof techniques; data structures and their physical storage representation, including space and performance analysis; recursive techniques; stacks, queues, lists, trees, sets, graphs; sorting and searching algorithms. Prerequisites: MATH 2200 or COMP 2300; COMP 1353 or COMP 2673.

COMP 2381 Object-Oriented Software Development (4 Credits)
Some problems are most naturally modeled by a hierarchy of objects and the relationships between those objects. This course introduces object-oriented design and development as a problem solving technique. Topics covered include the Java programming language, including classes, methods, fields, inheritance, interfaces, polymorphism, generics, static typing, design patterns, and the Java Collections Framework. Prerequisite: COMP 1353.

COMP 2673 Introduction to Computer Science III (4 Credits)
An introduction to several advanced topics in computer science. Topics vary from year to year and may include any of the following: theory of computing, cryptography, databases, computer graphics, graph theory, game theory, fractals, mathematical programming, wavelets, file compression, computational biology, genetic algorithms, neural networks, simulation and queuing theory, randomized algorithms, parallel computing, complexity analysis, numerical methods. Prerequisite: COMP 1672 or COMP 1771.

COMP 2691 Introduction to Computer Organization (4 Credits)
This course covers basic topics in Computer Organization and is a required course in the BS in Computer Science, BS in Game Development, and BS in Computer Engineering degrees. Topics include: instruction set architectures, integer and floating point arithmetic, processors, memory systems, and topics in storage and Input/Output. Prerequisite: COMP 1672.

COMP 2701 Topics in Computer Science (1-5 Credits)

COMP 2821 Introductory Game Design (4 Credits)
Learn fundamental game design practices through the creation of paper and physical game prototypes using a play-centric design process. Topics include the formal elements, dramatic elements, and system dynamics of games, with an emphasis on playtesting, game analysis, and group game projects. Program Prerequisites: Restricted to Game Development and EDPX majors or instructor approval. Course prerequisites: COMP 1352 or EDPX 2100.

COMP 2901 Computing and Society (4 Credits)
This course is designed to explore the social implications of computing practices, organization and experience. These topics and other issues are correlated with examples from the older and modern history of technology and science. Some formal experience with computing is assumed, but students who have a good familiarity with ordinary computing practice should be ready. Students are also expected to contribute their expertise in one or more of the areas of their special interest. Cross listed with DMST 3901.

COMP 3000 Seminar: The Real World (1 Credit)
Series of lectures by alumni and others on surviving culture shock when leaving the University and entering the job world. Open to all students regardless of major. Cross listed with MATH 3000.

COMP 3009 Essential Math for Data Science and AI (4 Credits)
The primary objective of the course is to empower students with the mathematical tools most used in data science. Students will develop an understanding of fundamental topics such as matrix operations, eigenvectors, and singular value decomposition (SVD), along with key calculus principles, including derivatives and integrals, with a focus on their applications in data science. The course also explores elements of multivariate calculus and optimization techniques to solve real-world problems, as well as probability concepts relevant to modeling uncertainty and variability in data. Additionally, students will gain exposure to implementing mathematical models using Python libraries such as NumPy, Pandas, and Matplotlib to analyze and manipulate data. By synthesizing these mathematical principles, students will be prepared for more advanced topics in data science and machine learning. Graduate student will be expected to upload a course presentation/project assignment to their individual Github portfolio.

COMP 3100 Human-Computer Interaction (4 Credits)
Introduces students in computer science and other disciplines to principles of and research methods in human-computer interaction (HCI). HCI is an interdisciplinary area concerned with the study of interaction between humans and interactive computing systems. Research in HCI looks at cognitive and social phenomena surrounding human use of computers with the goal of understanding their impact and creating guidelines for the design and evaluation of software, interfaces, physical products, and services in industry. Prerequisite: COMP 1353, but students from all disciplines are welcome. Cross-listed with COMP 4100.

COMP 3200 Discrete Structures (4 Credits)
Discrete mathematical structures and non-numerical algorithms; graph theory, elements of probability, propositional calculus, Boolean algebras; emphasis on applications to computer science. Prerequisites: COMP 2300 and COMP 1353 for undergraduates and COMP 3004 for graduates.

COMP 3334 Parallel and Distributed Computing for Data Science and AI (4 Credits)
Current techniques for effective use of parallel processing and large-scale distributed systems for data science. Programming assignments will give students experience in the use of these techniques. Specific topics will vary from year to year to incorporate recent developments. This course is not to be used for the MS Computer Science. Graduate students are required to contribute at least one course assignment to their personal Github portfolio. Prerequisite: COMP 2370.

COMP 3351 Programming Languages (4 Credits)
Learn the fundamentals of programming languages through functional programming through an in-depth understanding of syntax and semantics around program structures and how programming languages are parsed and interpreted. Understand recursion as a fundamental problem-solving paradigm and the important role that higher order types and kinds play in eliminating errors and simplifying software development. Satisfies the Programming Language elective requirement. Prerequisites: COMP 2362 and COMP 2370.

COMP 3352 Elements of Compiler Design (4 Credits)
Techniques required to design and implement a compiler; topics include lexical analysis, grammars and parsers, type-checking, storage allocation and code generation. Prerequisite: COMP 3351.

COMP 3353 Compiler Construction (4 Credits)
Design and implementation of a major piece of software relevant to compilers. Prerequisite: COMP 3352.

COMP 3361 Operating Systems I (4 Credits)
Operating systems functions and concepts; processes, process communication, synchronization; processor allocation, memory management in multiprogramming, time sharing systems. Prerequisites: for undergraduates: (COMP 2355 and COMP 2691) or COMP 2361; COMP 2370; for graduate students: COMP 3003, 3004, and 3005.

COMP 3371 Data Structures & Algorithms (4 Credits)
Design and analysis of algorithms and data structures; asymptotic complexity, recurrence relations, lower bounds; algorithm design techniques such as incremental, divide-and-conquer, dynamic programming, iterative improvement, greedy algorithms; randomized data structures and algorithms. Prerequisites: COMP 2370 or equivalent and COMP 3200.

COMP 3372 Advanced Algorithms (4 Credits)
Advanced techniques for the design and analysis of algorithms and data structures; amortized complexity, self-adjusting data structures; randomized , online, and string algorithms; NP-completeness, approximation and exact exponential algorithms; flow networks.

COMP 3381 Software Engineering I (4 Credits)
An introduction to software engineering. Topics include software processes, requirements, design, development, validation and verification and project management. Cross-listed with COMP 4381. Prerequisites: COMP 3351; COMP 3361 or COMP 2362; or instructor permission.

COMP 3382 Software Engineering II (4 Credits)
Continuation of COMP 3381. Topics include component-based software engineering, model-driven architecture, and service-oriented architecture. Prerequisite: COMP 3381.

COMP 3384 Secure Software Engineering (4 Credits)
This course is concerned with systematic approaches for the design and implementation of secure software. While topics such as cryptography, networking, network protocols and large scale software development are touched upon, this is not a course on those topics. Instead, this course is on identification of potential threats and vulnerabilities early in the design cycle. The emphasis in this course is on methodologies and paradigms for identifying and avoiding security vulnerabilities, formally establishing the absence of vulnerabilities, and ways to avoid security holes in new software. There are programming assignments designed to make students practice and experience secure software design and development. Prerequisites: COMP 2362 or COMP 3361.

COMP 3410 Web Programming I (4 Credits)
This course will give an overview of machine learning techniques, their strengths and weaknesses, and the problems they are designed to solve. This will include the broad differences between supervised, unsupervised and reinforcement learning and associated learning problems such as classification and regression. Techniques covered, at the discretion of the instructor, may include approaches such as linear and logistic regression, neural networks, support vector machines, kNN, decision trees, random forests, Naive Bayes, EM, k-Means, and PCA. After taking the course, students will have a working knowledge of these approaches and experience applying them to learning problems. Enforced Prerequisites: COMP 1353, MATH 1952, and MATH 2060.

COMP 3411 Web Programming II (4 Credits)
In this course you will learn how to develop a full-stack web application that is capable of serving dynamic content from a database. Furthermore, you will learn the core design concepts and principles that will enable you to develop scalable and easy to maintain webapplications - a set of skills that will serve you well in both your personal and professional projects in the future. Prerequisite: COMP 3410.

COMP 3412 Web Projects: Web Development III (4 Credits)
In this course you will learn how to develop, as a group, a full-stack web application that is capable of serving dynamic content from a database. We will use the MongoDB, ExpressJS, Angular, and Node.js (MEAN) software stack to work on a real-life problem presented to us by an external product owner. In the class we will use the Scrum framework for Agile development to work, as a software team, through several sprints of development. You will be peer reviewing each other throughout the course, and the product owner will also be reviewing your product through end-of-sprint demos as features are completed. The goal for this class is for it to be a fun, collaborative, and educational environment that demonstrates what it is like to work as a real software team. Prerequisite: COMP 3411.

COMP 3421 Database Organization & Management I (4 Credits)
An introductory class in database management systems covering both relational and non-relational databases with an emphasis on relational. Topics include database design, ER modeling, relational algebra, SQL, scripting, and embedded SQL. Each student will design, load, query and update a nontrivial database using a relational database management system (RDBMS). In addition, an introduction to a NoSQL database will be included. Graduate students will read one or two relevant technical papers and write a summary report. Prerequisites: for undergraduates: COMP 1353 or COMP 2673; for graduates: COMP 3005 or COMP3006 (MS Data Science).

COMP 3424 NoSQL Databases (4 Credits)
In this course, students learn what NoSQL databases are, learn to identify the differences between them, and gain a fundamental understanding between SQL, relational databases, and NoSQL databases. Students further explore which type of NoSQL database is the correct one given a use-cases, examining types, methods of communicating with it, contrasts to other NoSQL databases, performance and scalability. Prerequisites: for undergraduates, COMP 2355 or COMP 2361; for graduates: COMP 3005.

COMP 3431 Data Mining (4 Credits)
Data Mining is the process of extracting useful information implicitly hidden in large databases. Various techniques from statistics and artificial intelligence are used here to discover hidden patterns in massive collections of data. This course is an introduction to these techniques and their underlying mathematical principles. Topics covered include: basic data analysis, frequent pattern mining, clustering, classification, and model assessment. Prerequisites: COMP 2370.

COMP 3432 Machine Learning (4 Credits)
This course will give an overview of machine learning techniques, their strengths and weaknesses, and the problems they are designed to solve. This will include the broad differences between supervised, unsupervised and reinforcement learning and associated learning problems such as classification and regression. Techniques covered, at the discretion of the instructor, may include approaches such as linear and logistic regression, neural networks, support vector machines, kNN, decision trees, random forests, Naive Bayes, EM, k-Means, and PCA. After taking the course, students will have a working knowledge of these approaches and experience applying them to learning problems. Enforced Prerequisites: COMP 1353, MATH 1952, and MATH 2060.

COMP 3433 Data Visualization (4 Credits)
This course explores visualization techniques and theory. The course covers how to use visualization tools to effectively present data as part of quantitative statements within a publication/report and as an interactive system. Both design principles (color, layout, scale, and psychology of vision) as well as technical visualization tools/languages will be covered. Prerequisites: COMP 1353 and Python Programming.

COMP 3441 Introduction to Probability and Statistics for Data Science & AI (4 Credits)
The course introduces fundamentals of probability for data science. Students survey data visualization methods and summary statistics, develop models for data, and apply statistical techniques to assess the validity of the models. The techniques will include parametric and nonparametric methods for parameter estimation and hypothesis testing for a single sample mean and two sample means, for proportions, and for simple linear regression. Students will acquire sound theoretical footing for the methods where practical, and will apply them to real-world data, primarily using R. Prerequisite: COMP 1353, MATH 1952, MATH 2060.

COMP 3442 Advanced Probability and Statistics for Data Science (4 Credits)
This course builds on material in Probability and Statistics 1. Students will carry out model fitting and diagnostics for multiple regression, ANOVA, ANCOVA, and generalized linear models. Dimension reductions techniques such as PCA and Lasso are introduced, as are techniques for handling dependent data. The course introduces the principles of resampling and Bayesian Analysis. Students will acquire sound theoretical footing for the methods where practical, and will apply them to real-world data, primarily using R. Prerequisite: COMP 3441.

COMP 3450 Machine Learning Operations (4 Credits)
This course introduces the engineering and deployment of machine learning systems. Students will learn essential technical and organizational skills needed to develop and deploy ML solutions, progressing from fundamental concepts through to practical implementation. Key topics include ML project lifecycle management, containerization, web application development, data management strategies, and cloud deployment. Focus will be given to the application of industry-standard tools and best practices for building production-ready machine learning systems. Prerequisites: COMP 3421 and COMP 3432.

COMP 3455 Shell Scripting and System Tools (4 Credits)
This course covers navigating and utilizing tools in a UNIX environment, including use of common command line utilities, Bash and Python shell scripting, source control via Git, pipes and I/O redirection, networking in Python and OS multi-processing/multi-threading. More emphasis will be placed on using these tools than on how those tools work. Students should have experience with Python prior to taking this course.

COMP 3456 Deep Learning for Sequence Data (4 Credits)
This course introduces modern approaches to analyzing sequential data, with a focus on deep learning methods for time series and natural language processing. Students will develop practical skills in building and implementing neural network models for sequential data analysis, while gaining a thorough understanding of the underlying concepts. Through hands-on projects and real-world applications, students will learn to process, analyze, and generate predictions from various types of sequential data. The course emphasizes practical implementation and modern deep learning frameworks, bridging the gap between traditional statistical methods and contemporary neural network approaches. Prerequisite: COMP 3432.

COMP 3490 Undestanding AI (4 Credits)
Understanding Artificial Intelligence provides a comprehensive introduction to artificial intelligence (AI) for students of all backgrounds. This course explores the diverse approaches to AI, including rule-based systems, expert systems, knowledge representation, search algorithms, machine learning, natural language processing, robotics, and human-AI interaction. Students will gain hands-on experience with AI tools and applications while critically examining their impact on society. The course emphasizes the capabilities, limitations, and ethical considerations of AI, preparing students to engage thoughtfully with AI technologies in various fields. Prerequisites: Non-majors only, python programming is helpful but not required.

COMP 3491 Applications of Generative AI (4 Credits)
Applications of Generative Artificial Intelligence is a hands-on course explores the transformative power of generative artificial intelligence (AI) across various industries. Students will engage directly with generative AI tools and techniques, including text generation, image synthesis, music composition, code generation, and more. Through practical exercises, projects, and case studies, students will develop a deep understanding of how generative AI can enhance creativity, automate workflows, and drive innovation in diverse fields. This course is designed for students who have completed COMP 3490 and want to apply AI in practical, industry-relevant contexts. Prerequisite: COMP 3490 and non-majors.

COMP 3495 AI Design Project (4 Credits)
The Artificial Intelligence Design Project serves as the culminating experience for students who have completed COMP 3490 and COMP 3491. This course provides an opportunity for students to apply their AI knowledge and skills to a self-directed project in a domain of their choice. Over ten weeks, students will identify a problem, design an AI-driven solution, and develop a functional prototype or proof-of-concept. Emphasizing creativity, problem-solving, and ethical considerations, this capstone-style course enables students to make a tangible impact in their field. The course includes milestone check-ins, peer reviews, and a final presentation to showcase their work.

COMP 3501 Introduction to Artificial Intelligence (4 Credits)
Introduces a variety of Artificial Intelligence concepts and techniques, relevant to a broad range of applications. Students survey multiple techniques including search, knowledge representation and reasoning, probabilistic inference, machine learning, and natural language processing. Examines concepts of constraint programming, evolutionary computation and non-standard computation. Prerequisites: COMP 2370.

COMP 3525 Human Centered AI - Mind Reading Machines (4 Credits)
This course explores a key intersection of Artificial Intelligence (AI) and Human-Computer Interaction (HCI). How can AI enhance our interactions with computers by catering to our individual needs and differences? Can we teach computers to understand people's thoughts and feelings to improve overall interaction? Humans often adjust their communication when they notice the person they're talking to is frustrated—why can't computers do the same? In this interdisciplinary, research-focused course, you will read, present, and discuss seminal papers at the intersection of AI and HCI, including (but not limited to) affective computing, physiological computing, augmented cognition, and multimodal interaction. ## Learning Objectives (General Goal) Demonstrate a broad understanding of the intersections between Human-Computer Interaction and Artificial Intelligence, with an emphasis on user modeling. (LO1) Recognize how theories from fields outside of Computer Science can inform and enhance Computer Science practices. (LO2) Develop knowledge of state-of-the-art user modeling techniques, including practical applications and associated challenges. (LO3) Develop critical thinking and scientific communication skills related to current AI topics, including science writing and scientific discourse.

COMP 3531 Deep Learning: Model Design and Application (4 Credits)
This course addresses the foundational concepts and components of Artificial Neural Networks (ANN), highlighting their capabilities, strengths, and weaknesses as a machine learning algorithm. Students taking this course will develop ANN models from scratch in Python as a basis for understanding their design as well as the underlying mechanics and calculations that shape their behavior. Key topics such as forward-backward propagation, loss function characteristics and optimization will be considered in relation to model design and computational efficiency as well as to problems such as exploding and vanishing gradients. Training strategies (e.g., dropout, initialization, batch normalization) will further enable students to assess trade-offs in model bias & variance. Coupled with hands-on assignments, these building blocks provide the knowledge and skills required to effectively design and implement ANN models that are ethically and technically sound. As well as foreground important architectures such as Convolutional ANNs, Recurrent ANNs, LSTMS, and Transformers as well as their applicability to modern problems. Student learning and proficiency will be assessed based on a combination of quizzes, coding assignments, exams, and a culminating project. Prerequisite: COMP 3432.

COMP 3581 Algorithms for Data Science & AI (4 Credits)
This course introduces the design and analysis of algorithms within the context of data science and artificial intelligence. Topics include; asymptotic complexity and algorithm design techniques such as incremental, divide and conquer, dynamic programming, randomization, greedy algorithms, and advanced sorting techniques. Examples to illustrate techniques are drawn from multi-dimensional clustering (k-means and probabilistic), regression, decision trees, order statistics, data mining using apriori algorithms, and algorithms for generating combinatorial objects. Prerequisite: COMP 2370.

COMP 3591 Computational Geometry (4 Credits)
This class deals with the design and implementation of efficient algorithms for problems defined over geometric objects, such as points, lines, polygons, surfaces, etc. The methods and algorithms covered find applications in many areas, including computer graphics (e.g., hidden surface removal), computer-aided design and manufacturing (e.g., 3D printing), machine learning (e.g., supervised and unsupervised classification), geographic information systems (e.g. terrain visibility), robotics (e.g., motion planning), data mining (e.g., dimensionality reduction), and computer vision (3D reconstruction), to name a few. Fundamental geometric problems such as partitioning, proximity, intersection, convexity, visibility, point location, and motion planning are focused on. Efficient data structures and algorithms for their solutions and design techniques germane to the field, such as divide-and-conquer, plane sweep, randomization, duality, etc. are discussed in detail. Practical methods for the robust implementation of geometric algorithms are also covered. Prerequisites: COMP 2300 and COMP 2370.

COMP 3621 Computer Networking (4 Credits)
An introduction to computer networks with an emphasis on Internet protocols. Topics include: internet design, application layer protocols such as SMTP and HTTP, session layer protocols including TCP and UDP, the internet protocol (IP), link layer technology such as Ethernet, and security issues related to networking. Programming experience of client/server architectures using sockets and TCP/UDP through projects is emphasized. Prerequisites: for undergraduates: (COMP 2355 or COMP 2361) and COMP 2370; for graduates COMP 3004 and COMP 3006. Cross listed with COMP 4621.

COMP 3681 Networking for Games (4 Credits)
Implementing the networking code for multiplayer games is a complex task that requires an understanding of performance, security, game design, and advanced programming concepts. In this course, students are introduced to the networking stack and how this is connected to the Internet, learn how to write protocols for games, and implement several large games using a game engine that demonstrate the kind of networking and protocols required by different genres of games. In addition, tools are introduced that help understand and debug networking code, simplify the creation of protocols, and make the development of networking code easier.

COMP 3701 Topics in Computer Graphics (4 Credits)

COMP 3702 Topics in Database (4 Credits)

COMP 3703 Topics-Artificial Intelligence (4 Credits)

COMP 3704 Advanced Topics: Systems (4 Credits)

COMP 3705 Topics in Computer Science (1-4 Credits)

COMP 3711 Special Topics in Data Science & AI (4 Credits)
Special Topics in Data Science & AI highlights selected methods & applications in machine learning that are not specifically addressed within the curriculum. Areas of study may include, but are not limited to, natural language processing, recommender systems, data science for social justice, machine learning operations, generative AI and more. Prerequisites: COMP 3432 and COMP 3441.

COMP 3721 Computer Security (4 Credits)
This course gives students an overview of computer security along with some cryptography. Some network security concepts are also included. Other concepts include coverage of risks and vulnerabilities, policy formation, controls and protection methods, role-based access controls, database security, authentication technologies, host-based and network-based security issues. Prerequisite: COMP 2362 or COMP 3361. Cross listed with COMP 4721.

COMP 3722 Network Security (4 Credits)
Network Security covers tools and techniques employed to protect data during transmission. It spans a broad range of topics including authentication systems, cryptography, key distribution, firewalls, secure protocols and standards, and overlaps with system security concepts as well. This course will provide an introduction to these topics, and supplement them with hands-on experience. Prerequisites: COMP 3721 or permission of instructor.

COMP 3723 Ethical Hacking (4 Credits)
Ethical hacking is the process of probing computer systems for vulnerabilities and exposing their presence through proof-of-concept attacks. The results of such probes are then utilized in making the system more secure. This course will cover the basics of vulnerability research, foot printing targets, discovering systems and configurations on a network, sniffing protocols, firewall hacking, password attacks, privilege escalation, rootkits, social engineering attacks, web attacks, and wireless attacks, among others. Prerequisites: COMP 1203 or COMP 2673 (CS Intro sequence).

COMP 3731 Computer Forensics (4 Credits)
Computer Forensics involves the examination of information contained in digital media with the aim of recovering and analyzing latent evidence. This course will provide students an understanding of the basic concepts in preservation, identification, extraction and validation of forensic evidence in a computer system. The course covers many systems level concepts such as disk partitions, file systems, system artifacts in multiple operating systems, file formats, email transfers, and network layers, among others. Students work extensively on raw images of memory and disks, and in the process, build components commonly seen as features of commercial forensics tools (e.g. file system carver, memory analyzer, file carver, and steganalysis). Prerequisites: COMP 3361; COMP 2355 or 2361 for undergraduates; COMP 3006 for graduates.

COMP 3801 Introduction Computer Graphics (4 Credits)
Fundamentals of 3D rendering including the mathematics behind coordinate systems, projections, clipping, hidden surface removal, shadows, lighting models, shading models, and mapping techniques. Significant use of 3D APIs through shader programming is covered along with the basics of 3D model representation and animations. Satisfies "Advanced Programming" requirements for graduate students. Prerequisites: COMP 2370, MATH 1952 or 1962.

COMP 3820 Introduction to Game Programming (4 Credits)
Learn the fundamentals of game programming by creating 2D and 3D games using a modern game engine. Topics include working with 2D/3D art assets, character controllers, physics, camera systems, 2D and 3D animation integration, lighting, audio, user interfaces, artificial intelligence, and level design. The emphasis in this course is on game programming and implementing game mechanics in fully-working games. Prerequisites: COMP 2381 and COMP 2821.

COMP 3821 Game Programming I (4 Credits)
A continuation of Introduction to Game Programming, this course introduces advanced topics that are essential as future game programmers. Students have the opportunity to learn game engine architecture, 2D and 3D linear algebra for graphics, sprites, 2D and 3D animations, input handling, finite state machines, particle systems, user interfaces, game audio, and artificial intelligence for games. Prerequisites: COMP 2370 and COMP 3820.

COMP 3822 Game Programming II (4 Credits)
In this course, students learn how to work with a 3D game engine and build 3D games. Topics include algorithms, mathematics for 3D game engines, scene management, animations, 3D shaders, particle systems, physics for games, UIs, terrain systems, and working with higher-level scripting languages on top of the low-level implementation language. Prerequisites: COMP 3821. Suggested corequisite or prerequisite: COMP 3801.

COMP 3831 Game Prototyping (4 Credits)
Introduces game prototyping, where game concepts are created and developed into working prototypes using student-chosen game development tools. Engages in critical awareness of game creation practices through discussions, critiques, demos, and player testing. Students have the opportunity to explore their creativity, to expand their knowledge of game design, and to build a broad portfolio of ideas and working projects that demonstrate both their design and technical skillsets. Required for Game Development majors, though all majors are welcome. Prerequisites: COMP 2821.

COMP 3832 Game Capstone I (4 Credits)
Students design, build, critique, and playtest their game prototypes from Game Prototyping. Both art and programming are developed by the student teams with the instructor acting as a project manager to ensure that goals are met through the 10-week development process through various milestones. In addition to building the game, students alter their game design document to add new features, making corrections to prior design issues, and focus on making the game playable and "fun." Prerequisite: COMP 3821 and COMP 3831.

COMP 3833 Game Capstone II (4 Credits)
Students design, build, critique and playtest their working game from Game Capstone I. Both art and programming are developed by the student teams with the instructor acting as a project manager to ensure that goals are met through the 10-week development process through various milestones. In addition to building the game, students modify their design document and implement changes in their game, create new concept art for the features, build an introduction level into their game, test the game with "Play testers", and focus on creating a game that is "fun" to play. By the end of the quarter, their game is ready for distribution on an appropriate platform. Prerequisite: COMP 3832.

COMP 3904 Internship/Co-Op in Computing (0-10 Credits)
Experiential learning through employment with a company to work in computer science. Students are expected to find the internship/co-op and connect their hiring manager with an advisor in the department to get approval. Requirements for approval include a job description that shows what will be accomplished throughout the quarter and a final report from the hiring manager to report on their performance so that a grade may be assigned. Prerequisites: COMP 2370 and approval of department. MS Data Science students: practical experience working under the supervision of data science employer and a data science faculty advisor. Learning outcomes will be assessed via student work such as summary reports, literature reviews, presentations, code notebooks and/or repositories. Prerequisites: permission of Data Science Faculty Director.

COMP 3991 Independent Study (1-10 Credits)
Cannot be arranged for any course that appears in the regular course schedule for that particular year.

COMP 3995 Independent Research (1-10 Credits)

Game Studies

GAME 2010 Foundations of Gaming (1-2 Credits)
This course introduces students to foundational concepts in gaming from an interdisciplinary perspective. Students will explore what a game is and how games make interdisciplinary connections. Students will be invited to bring their passion, values, and lived experience to think critically through the challenges and opportunities in the rapidly-evolving field of gaming. Restricted to Gaming LLC Students.

GAME 2011 Applications in Gaming (1-2 Credits)
Games are as old as human culture, and their ongoing evolution makes them increasingly relevant to both human connection and the public good today. This course provides students with opportunities to explore how games connect with cultures and histories across the globe. In considering applications of gaming, students will engage with principles of ethics, inclusion, and equity in game development and gaming culture. Restricted to Gaming LLC Students. Prerequisite: Foundations of Gaming.

GAME 2012 Future Directions in Gaming (1-2 Credits)
This course challenges students to look to the future of gaming for the public good, addressing key questions such as: How do games intersect with large challenges and opportunities facing the world? How do we engage with the designers and players across games to advance the public good? Students will critically examine how game design and play intersect with concepts of the public good. Students will explore their agency in designing and using gaming as a strategy for public good discovery; and make connections with local industry professionals. Restricted to Gaming LLC Students. Prerequisite: Applications of Gaming.