

# Stevens Institute of Technology 2006-2007 Catalog

[Table of Contents](#)

[Calendar](#)

[Introduction](#)

[Undergraduate Programs](#)

[Graduate Programs](#)

[School of Sciences and Arts](#)

- [List of Programs](#)  
 - [Undergraduate Programs](#)  
 - [Department of Chemistry and Chemical Biology](#)

[▶](#) - [Department of Computer Science](#)

- [Department of Humanities and Social Sciences](#)

- [Department of Mathematical Sciences](#)

- [Department of Physics and Engineering Physics](#)

- [Interdisciplinary Program: Computational Science](#)

[School of Engineering](#)

[School of Technology Management](#)

[Interdisciplinary Programs](#)

[ESL and Special Courses](#)

[Physical Education, Athletics and Recreation](#)

[Research Environment](#)

[Student Services](#)

[Financing Education](#)

[Student Life](#)

[Learning About The Campus](#)

[Policies](#)

## The Arthur E. Imperatore School of Sciences and Arts



[Undergraduate Programs](#)

[Suggested Application Areas](#)

[Concentration Areas Within Computer Science](#)

[Minor in Computer Science](#)

[Interdisciplinary Program in Computational Science](#)

[Graduate Programs](#)

[Master of Science - Computer Science](#)

[Master of Science - Information Systems \(Interdisciplinary\)](#)

[Doctoral Program](#)

[Graduate Certificate Programs](#)

[Laboratories](#)

[Undergraduate Courses](#)

[Graduate Courses](#)

### Department of Computer Science

STEPHEN L. BLOOM, DIRECTOR

#### FACULTY\*

##### Professors

Stephen L. Bloom, Ph.D. (1968), Massachusetts Institute of Technology  
 Daniel Duchamp, Ph.D. (1988), Carnegie Mellon University  
 A. Satyanarayana, Ph.D. (1981), Jawaharlal Nehru Technological University

##### Associate Professors

Adriana Compagnoni, Ph.D. (1995), Katholieke University Nijmegen  
 Dominic Duggan, Ph.D. (1990), University of Maryland  
 George Kamberov, Ph.D. (1990), University of Pennsylvania  
 Aaron David Klappholz, Ph.D. (1974), University of Pennsylvania  
 David A. Naumann, Ph.D. (1992), University of Texas at Austin  
 John Oliensis, Ph.D. (1981), University of Chicago  
 Rebecca N. Wright, Ph.D. (1994), Yale University

##### Assistant Professors

Elli Angelopoulou, Ph.D. (1997), Johns Hopkins University  
 H. Quynh Dinh, Ph.D. (2002), Georgia Institute of Technology  
 Susanne Wetzels, Ph.D. (1998), Saarland University

##### Industry Research Professor

Larry Bernstein, M.S.E.E. (1962), New York University  
 Manu Malek, Ph.D. (1970), University of California Berkeley

\*The list indicates the highest earned degree, year awarded, and institution where earned.

### UNDERGRADUATE PROGRAMS

Computer science deals with the fundamental scientific laws and engineering principles which govern the design, manufacture, and use of computers. A computer scientist is involved in work ranging from mathematical studies of problem-solving procedures to

[Administrative Directory](#)

[Faculty Directory](#)

[Travel Directions](#)

[Campus Map](#)

engineering research in advanced computing systems.

The undergraduate curriculum is designed to continue the Stevens tradition of a broad-based education exceeding the requirements of the Computer Science Accreditation Board (CSAB) in both hardware and software courses, along with a strong theoretical component as a foundation for software engineering. It features basic courses in chemistry, physics, and mathematics; a sequence of courses in the humanities, as well as physical education; plus an in-depth sequence of specialized computer science courses.

The computer science courses at Stevens include the fundamentals needed by every computer scientist. As the software industry evolves, practitioners are increasingly expected to build reliable systems for mission- and life-critical applications. Such professionals distinguish themselves through mastery of scientific and mathematical foundations, mastery of software tools and methods, and experience in collaborative work on large projects. The Stevens computer science curriculum is designed to provide an outstanding education in all of these areas.

Upon completion of the program, you are able to formulate problems in algorithmic terms and solve them by means of a computer, as well as manage and reason about large software systems, including the ability to use off-the-shelf components in the construction of software applications. You will have written multi-threaded programs, including thread synchronization, and will be able to reason informally about the correctness of programs. Electives provide you with an opportunity to delve deeper into specific areas of interest, including financial systems, software engineering, intelligent design and manufacturing, mathematics and security.

Stevens computer science graduates are greatly prized by industry and find excellent positions in areas ranging from the financial industry to software and telecommunication companies, such as AT&T, IBM, Microsoft, Sun Microsystems, and Verizon. In addition, a significant number of Stevens graduates continue their education at the graduate level, including those programs offered at Stevens.

The course sequence for computer science is as follows:

### **B.S. in Computer Science - Chemistry/Biology Lab**

#### **Freshman Year**

		<b>Term I</b>		
		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 115	Calculus I	3	0	3
CS 115**	Intro. to Computer Science	3	2	4
PEP 111	Mechanics	3	0	3
CH 115/117	General Chemistry and Lab I	3	3	4
PE 200	Physical Education I	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>7</b>	<b>15</b>

**Term II**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 116	Calculus II	3	0	3
CS 284	Data Structures and Algorithms	3	1	4
MA 134	Discrete Mathematics	3	0	3
CH 281/282	Biology & Biotechnology Laboratory	3	3	4
HUM	Humanities	3	0	3
PE 200	Physical Education II	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>6</b>	<b>18</b>

**Sophomore Year****Term III**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 383	Comp. Org. & Prog.	3	0	3
CS 385	Data Structures and Algorithms II	3	1	4
CS 335	Computational Structures	3	0	3
HUM	Humanities	3	0	3
HUM	Humanities	3	0	3
PE 200	Physical Education III	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Term IV**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 488	Comp. Architecture	3	0	3
CS 434	Theory of Computation	3	0	3
	Elective*	3	0	3
MA 222	Probability & Statistics	3	0	3
HUM	Humanities	3	0	3
PE 200	Physical Education IV	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Junior Year****Term V**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 492	Operating Systems	3	0	3
CS 496	Programming Languages	3	0	3
	Elective*	3	0	3
HUM	Humanities	3	0	3

PE 200	Physical Education V	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>2</b>	<b>13</b>

**Term VI**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 442	Database Mgmt. Systems	3	1	4
CS 494	Compiler Design	3	0	3
	Elective*	3	0	3
HUM	Humanities	3	0	3
PE 200	Physical Education VI	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>3</b>	<b>14</b>

**Senior Year****Term VII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 551	Software Eng. & Pract. I	3	0	3
CS	CS Elective	3	0	3
HUM	Humanities	3	0	3
	Elective*	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

**Term VIII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 552	Software Eng. & Pract. II	3	0	3
CS	CS Elective	3	0	3
	Elective*	3	0	3
	Free Elective	3	0	3
Hu	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

**B.S. in Computer Science - Physics Lab****Freshman Year****Term I**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 115	Math. Analysis I	3	0	3
CS 115**	Intro. to Computer Science	3	2	4
PEP 111	Mechanics	3	0	3

CH 115	General Chemistry I	3	0	3
PE 200	Physical Education I	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>4</b>	<b>14</b>

**Term II**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 116	Math. Analysis II	3	0	3
CS 284	Data Structures and Algorithms	3	1	4
MA 134	Discrete Mathematics	3	0	3
PEP 112/222	Electricity and Magnetism and Laboratory	3	3	4
HUM	Humanities	3	0	3
PE 200	Physical Education II	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>6</b>	<b>18</b>

**Sophomore Year****Term III**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 383	Comp. Org. & Prog.	3	0	3
CS 385	Data Structures and Algorithms II	3	1	4
CS 335	Computational Structures	3	0	3
HUM	Humanities	3	0	3
HUM	Humanities	3	0	3
PEP 221	Computational Structures	0	3	1
PE 200	Physical Education III	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>6</b>	<b>18</b>

**Term IV**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 488	Comp. Architecture	3	0	3
CS 434	Theory of Computation	3	0	3
	Elective*	3	0	3
MA 222	Probability & Statistics	3	0	3
HUM	Humanities	3	0	3
PE 200	Physical Education IV	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Junior Year****Term V**

		Hrs. Per Wk.		
		Class	Lab	Sem.

			Cred.		
CS 492	Operating Systems	3	0	3	
CS 496	Programming Languages	3	0	3	
	Elective*	3	0	3	
HUM	Humanities	3	0	3	
PE 200	Physical Education V	0	2	1	
<b>TOTAL</b>		<b>12</b>	<b>2</b>	<b>13</b>	

**Term VI**

			Hrs. Per Wk.			Cred.		
			Class	Lab	Sem.			
CS 442	Database Mgmt. Systems	3	1	4				
CS 494	Compiler Design	3	0	3				
	Elective*	3	0	3				
HUM	Humanities	3	0	3				
PE 200	Physical Education VI	0	2	1				
<b>TOTAL</b>		<b>12</b>	<b>3</b>	<b>14</b>				

**Senior Year****Term VII**

			Hrs. Per Wk.			Cred.		
			Class	Lab	Sem.			
CS 551	Software Eng. & Pract. I	3	0	3				
CS	CS Elective	3	0	3				
HUM	Humanities	3	0	3				
	Elective*	3	0	3				
HUM	Humanities	3	0	3				
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>				

**Term VIII**

			Hrs. Per Wk.			Cred.		
			Class	Lab	Sem.			
CS 552	Software Eng. & Pract. II	3	0	3				
CS	CS Elective	3	0	3				
	Elective*	3	0	3				
	Free Elective	3	0	3				
HUM	Humanities	3	0	3				
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>				

Humanities Electives must include HSS 371.

\* Must include a total of six electives, of which one must be MGT 243, MGT 111, or BT 121; one must be a science elective; and the remaining four must be CS courses or courses from a department approved Application Area sequence.

\*\*CS 115 assumes that the student has had prior experience with

programming in C++, Java, or some other imperative programming language. Students who have little or no background in programming should take CS 105 in Term I, CS 115 in Term II, CS 284 in Term III, and CS 385 in Term IV in place of the Term IV elective.

[back to top](#)

### **Application Areas**

The Application Areas defined below allow students to devote up to four of their electives (marked by \* in the requirements) to disciplines outside computer science. Depending upon the Application Area, the science elective and/or the management elective may be chosen to support the course sequence.

You must receive prior departmental approval in order to substitute an Application Area for computer science electives. The computer science department works with other departments to develop Application Area sequences in disciplines that are related to computer science. Below are Application Areas already approved.

### **Financial Systems**

Intended for students who contemplate a career in the financial sector. Those who select this Application Area option should take MGT 243 (Macroeconomics) as their required management elective.

- MGT 244 Microeconomics
- BT 115 Financial Accounting
- BT 215 Cost Accounting
- BT 321 Finance

### **Mathematics**

This Application Area focuses on topics in mathematics that utilize computing.

- MA 221 Differential Equations
- MA 331 Intermediate Statistics
- MA 346 Numerical Methods
- MA 520 Computational Linear Algebra I

### **Computational Chemistry & Biology**

- CH 116/118 Chemistry II/Chemistry Lab II
- CH 321 Thermodynamics
- CH 381 Cell Biology
- CH 664 Computer Methods in Chemistry

It is also suggested, but not required, that CH 484 (Introduction to Molecular Genetics) be taken as the free elective.

### **Engineering**

For students with interest in hardware:

- E 245 Circuits and Systems
- CPE 358 Switching Theory and Logical Design
- CPE 390 Microprocessor Systems
- MA 221 Differential Equations

In addition, students should take PEP 112 as the science elective.

[back to top](#)

### **Concentration Areas Within Computer Science**

Here are suggested course sequences for students who are interested in specific areas within computer science. These sequences are optional - indeed, each student may choose the six

CS electives according to personal interests. A concentration does not appear on the diploma. Students should understand that concentrated electives are merely suggestions. A student may choose to take all, some, or none of the courses in a concentration.

### **Networks**

- CS 521 TCP/IP Networking
- CS 549 Distributed Systems
- CS 666 Information Networks I
- CS 668 Foundations of Cryptography
- CS 669 Network Management

### **Design of Games**

A five-course concentration in Game Design is designed to prepare the student for an entry-level position in the computer-games industry. The emphasis is on the creation of network, multi-player, and 3-D games.

- CS 437 Interactive Computer Graphics I
- CS 482 Artificial Intelligence
- CS 521 TCP/IP Networking
- CS 549 Distributed Systems
- CS 638 Interactive Computer Graphics II

It is further recommended that a game design be the subject of the student's CS 551/CS 552 software project.

### **Quantitative Software Engineering**

With two courses beyond the required CS 551 and CS 552, students will understand the fundamentals of software engineering with the unique Stevens emphasis on quantitative analysis and case histories.

CS 533 Cost Estimation and Metrics -- recommended plus one of the following:

- CS 564 Software Requirements Acquisition and Analysis
- CS 565 Software Architecture and Component-Based Design
- CS 567 Software Testing, Quality Assurance, and Maintenance
- CS 687 Engineering of Large Software Systems
- CS 689 Software Reliability Engineering

### **Minor in Computer Science**

You may qualify for a minor in computer science by taking the courses indicated below. Enrollment in a minor means you must meet the Institute's requirements for minor programs. You may not use the same courses for both a major and a minor. Only courses completed with grade of "C" or better are accepted towards a student's minor.

Requirements for a Minor in Computer Science:

	Prerequisite
CS 115 Introduction to Computer Science	
MA 134 Discrete Mathematics	
CS 383 Computer Org. & Programming	CS 284
CS 284 Data Structures and Algorithms	CS 115, Corequisite MA 334



CS 385 Data Structures and Algorithms II      CS 284

### Bachelor of Science in Cybersecurity

The Bachelor of Science program in Cybersecurity is structured to provide students with security expertise within the context of a broad education. A solid education in security requires not only a strong focus in science and computer science in particular (e.g., need for robust implementation and software validation), but must also incorporate some aspects of engineering and technology management. While cryptographers strive to develop the best security solution possible, actual implementations of theoretical concepts often fail due to technological limitations, cost restraints, and human factors that were not part of the initial design process. For a solution to gain practical relevance, the end user must be able and willing to use it. From an economical point of view a solution must provide a substantial monetary benefit to the customer. In order to allow for these complex issues to be better addressed, an education in cybersecurity must integrate science, technology, and management.

### Study Plans

In the following we present four study plans:

#### Basic Study Plan

##### Freshman Year

		Term I		
		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 115	Calculus I	3	0	3
CS 115	Introduction to Computer Science	3	2	4
CH 115	General Chemistry I	3	0	3
CH 117	General Chemistry Lab I	0	3	1
BT 131	Technogenesis: Introduction to Innovation and Creativity	3	0	3
PE 200	Physical Education I	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>7</b>	<b>15</b>

##### Term II

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 116	Calculus II	3	0	3
CS 284	Data Structures and Algorithms	3	1	4
CH 281	Biology and Biotechnology	3	0	3
CH 282	Biology Laboratory	0	3	1
MA 134	Discrete Mathematics	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education II	0	2	1

	<b>TOTAL</b>		<b>15</b>	<b>6</b>	<b>18</b>
<b>Sophomore Year</b>					
	<b>Term III</b>				
		Hrs. Per Wk.			
		Class Lab Sem.			
		Cred.			
CS 335	Computational Structures	3	0	3	
CS 383	Computer Organization and Programming	3	0	3	
CS 385	Data Structures and Algorithms II	3	1	4	
HUM	Humanities*	3	0	3	
HUM	Humanities*	3	0	3	
PE 200	Physical Education III	0	2	1	
	<b>TOTAL</b>	<b>15</b>	<b>3</b>	<b>17</b>	
	<b>Term IV</b>				
		Hrs. Per Wk.			
		Class Lab Sem.			
		Cred.			
CS 488	Computer Architecture	3	0	3	
CS	Computer Science Elective	3	0	3	
CS 392	System Programming	3	0	3	
CS 434	Theory of Computation	3	0	3	
MA 222	Probability and Statistics	3	0	3	
PE 200	Physical Education IV	0	2	1	
	<b>TOTAL</b>	<b>15</b>	<b>2</b>	<b>16</b>	
<b>Junior Year</b>					
	<b>Term V</b>				
		Hrs. Per Wk.			
		Class Lab Sem.			
		Cred.			
CS/MA 503	Discrete Mathematics for Cryptography	3	0	3	
CS 496	Programming Languages	3	0	3	
CS 492	Operating Systems	3	0	3	
CS 578	Privacy in a Networked World	3	0	3	
HUM	Humanities*	3	0	3	
PE 200	Physical Education V	0	2	1	
	<b>TOTAL</b>	<b>15</b>	<b>2</b>	<b>16</b>	
	<b>Term VI</b>				
		Hrs. Per Wk.			
		Class Lab Sem.			
		Cred.			
CS 573	Fundamentals of Cybersecurity	3	0	3	
CS 442	Database Management Systems	3	1	4	
CS 579	Foundations of Cryptography	3	0	3	

HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education VI	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Senior Year****Term VII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 551	Software Engineering and Practice I**	3	0	3
CS 576	Secure Systems	3	0	3
CS 577	Cybersecurity Laboratory	0	3	3
CS	CS Elective	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>12</b>	<b>3</b>	<b>15</b>

**Term VIII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 552	Software Engineering and Practice II**	3	0	3
CS	Security Elective	3	0	3
CS	CS Elective	3	0	3
	Elective	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

(\*) Humanities requirements: Two Group A courses at the 100 level, two Group B courses at the 100 level, and four courses at the 300/400 level that must include HSS 371.

(\*\*) Students must choose a security related project when they do their senior project as part of CS 551/552. The cybersecurity advisors are resources for project selection and project execution.

**Total Credits: 114-26-129**

**Basic Study Plan - Physics Option****Freshman Year****Term I**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 115	Calculus I	3	0	3
CS 115	Introduction to Computer. Science	3	2	4
PEP 111	Mechanics	3	0	3
BT 131	Technogenesis: Introduction to Innovation and Creativity	3	0	3
PE 200	Physical Education I	0	2	1

<b>TOTAL</b>		<b>12</b>	<b>4</b>	<b>14</b>
--------------	--	-----------	----------	-----------

**Term II**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 116	Calculus II	3	0	3
CS 284	Data Structures and Algorithms	3	1	4
PEP 112	Electricity and Magnetism	3	0	3
PEP 221	Physics Laboratory I	0	3	1
MA 134	Discrete Mathematics	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education II	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>6</b>	<b>18</b>

**Sophomore Year****Term III**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
PEP 222	Physics Laboratory II	0	3	1
CS 335	Computational Structures	3	0	3
CS 383	Computer Organization and Programming	3	0	3
CS 385	Data Structures and Algorithms II	3	1	4
HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education III	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>6</b>	<b>18</b>

**Term IV**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cre.		
CS 488	Computer Architecture	3	0	3
CS	Computer Science Elective	3	0	3
CS 392	System Programming	3	0	3
CS 434	Theory of Computation	3	0	3
MA 222	Probability and Statistics	3	0	3
PE 200	Physical Education IV	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Junior Year****Term V**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS/MA 503	Discrete Mathematics for Cryptography	3	0	3

CS 496	Programming Languages	3	0	3
CS 492	Operating Systems	3	0	3
CS 578	Privacy in a Networked World	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education V	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Term VI**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 573	Fundamentals of Cybersecurity	3	0	3
CS 442	Database Management Systems	3	1	4
CS 579	Foundations of Cryptography	3	0	3
HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education VI	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Senior Year****Term VII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 551	Software Engineering and Practice I**	3	0	3
CS 576	Secure Systems	3	0	3
CS 577	Cybersecurity Laboratory	0	3	3
CS	CS Elective	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>12</b>	<b>3</b>	<b>15</b>

**Term VIII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 552	Software Engineering and Practice II**	3	0	3
CS	Security Elective	3	0	3
CS	CS Elective	3	0	3
	Elective	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

(\*) Humanities requirements: Two Group A courses at the 100 level, two Group B courses at the 100 level, and four courses at the 300/400 level that must include HSS 371.

(\*\*) Students must choose a security related project when they do their senior project as part of CS 551/552. The cybersecurity advisors are resources for project selection and project execution.

**Total Credits: 114-26-129**

### Study Plan for Students Without Computer Science Background

#### Freshman Year

##### Term I

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 115	Calculus I	3	0	3
CS 105	Intro. to Scientific Computing	2	2	3
CH 115	General Chemistry I	3	0	3
CH 117	General Chemistry Lab I	0	3	1
BT 131	Technogenesis: Introduction to Innovation and Creativity	3	0	3
PE 200	Physical Education I	0	2	1
<b>TOTAL</b>		<b>11</b>	<b>7</b>	<b>14</b>

##### Term II

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 116	Calculus II	3	0	3
CS 115	Introduction to Computer Science	3	2	4
CH 281	Biology and Biotechnology	3	0	3
CH 282	Biology Laboratory	0	3	1
MA 134	Discrete Mathematics	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education II	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>7</b>	<b>18</b>

#### Sophomore Year

##### Term III

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 335	Computational Structures	3	0	3
CS 383	Computer Organization and Programming	3	0	3
CS 284	Data Structures and Algorithms	3	1	4
HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education III	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

##### Term IV

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		

CS 488	Computer Architecture	3	0	3
CS 392	System Programming	3	0	3
CS 385	Data Structures and Algorithms II	3	1	4
CS 434	Theory of Computation	3	0	3
MA 222	Probability and Statistics	3	0	3
PE 200	Physical Education IV	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Junior Year****Term V**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS/MA 503	Discrete Mathematics for Cryptography	3	0	3
CS 496	Programming Languages	3	0	3
CS 492	Operating Systems	3	0	3
CS 578	Privacy in a Networked World	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education V	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Term VI**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 573	Fundamentals of Cybersecurity	3	0	3
CS 442	Database Management Systems	3	1	4
CS 579	Foundations of Cryptography	3	0	3
HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education VI	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Senior Year****Term VII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 551	Software Engineering and Practice I**	3	0	3
CS 576	Secure Systems	3	0	3
CS 577	Cybersecurity Laboratory	0	3	3
CS	CS Elective	3	0	3
HUM	Humanities*	3	0	3
<b>TOTAL</b>		<b>12</b>	<b>3</b>	<b>15</b>

**Term VIII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 552	Software Engineering and Practice II**	3	0	3
CS	Security Elective	3	0	3
CS	CS Elective	3	0	3
	Elective	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

(\*) Humanities requirements: Two Group A courses at the 100 level, two Group B courses at the 100 level, and four courses at the 300/400 level that must include HSS 371.

(\*\*) Students must choose a security related project when they do their senior project as part of CS 551/552. The cybersecurity advisors are resources for project selection and project execution.

**Total Credits: 113-28-129**

### Study Plan for Students in the Stevens Scholar Program

#### Freshman Year

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
<b>Term I</b>				
MA 115	Calculus I	3	0	3
CS 115	Introduction to Computer Science	3	2	4
	Honors I			
CH 115	General Chemistry I	3	0	3
CH 117	General Chemistry Lab I	0	3	1
BT 131	Technogenesis: Introduction to Innovation and Creativity	3	0	3
PE 200	Physical Education I	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>7</b>	<b>15</b>

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
<b>Term II</b>				
MA 116	Calculus II	3	0	3
CS 115	Introduction to Computer Science	3	0	4
	Honors II			
CH 281	Biology and Biotechnology	3	0	3
CH 282	Biology Laboratory	0	3	1
MA 134	Discrete Mathematics	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education II	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>5</b>	<b>18</b>



**Sophomore  
Year****Term III**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 335	Computational Structures	3	0	3
CS 383	Computer Organization and Programming	3	0	3
	Elective	3	0	3
HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education III	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Term IV**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 488	Computer Architecture	3	0	3
CS	CS Elective	3	0	3
CS 392	System Programming	3	0	3
CS 434	Theory of Computation	3	0	3
MA 222	Probability and Statistics	3	0	3
PE 200	Physical Education IV	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Junior Year****Term V**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS/MA 503	Discrete Mathematics for Cryptography	3	0	3
CS 496	Programming Languages	3	0	3
CS 492	Operating Systems	3	0	3
CS 578	Privacy in a Networked World	3	0	3
HUM	Humanities*	3	0	3
PE 200	Physical Education V	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>2</b>	<b>16</b>

**Term VI**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 573	Fundamentals of Cybersecurity	3	0	3
CS 442	Database Management Systems	3	1	4
CS 579	Foundations of Cryptography	3	0	3
HUM	Humanities*	3	0	3
HUM	Humanities*	3	0	3

PE 200	Physical Education VI	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Senior Year****Term VII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 551	Software Engineering and Practice I**	3	0	3
CS 576	Secure Systems	3	0	3
CS 577	Cybersecurity Laboratory	0	3	3
CS	Security Elective	3	0	3
HUM	Humanities*	3	0	3
<b>TOTAL</b>		<b>12</b>	<b>3</b>	<b>15</b>

**Term VIII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
CS 552	Software Engineering and Practice II**	3	0	3
CS	Security Elective	3	0	3
CS	CS Elective	3	0	3
	Elective	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

(\*) Humanities requirements: Two Group A courses at the 100 level, two Group B courses at the 100 level, and four courses at the 300/400 level that must include HSS 371.

(\*\*) Students must choose a security related project when they do their senior project as part of CS 551/552. The cybersecurity advisors are resources for project selection and project execution. Students may also choose the physics option. For details see the basic study plan.

**Total Credits: 114-24-128**

**Curriculum Summary**

The program requires the following courses:

**Science**

CH 115 General Chemistry I  
 CH 117 General Chemistry Lab I  
 CH 281 Biology and Biotechnology  
 CH 282 Intro. Biology Lab

**Math and Statistics**

MA 115 Calculus I  
 MA 116 Calculus II  
 MA 134 Discrete Mathematics  
 MA 222 Probability and Statistics

**Core Computer Science**

CS 115 Introduction to Computer Science  
CS 284 Data Structures and Algorithms  
CS 335 Computational Structures  
CS 383 Computer Organization and Programming  
CS 385 Data Structures and Algorithms II  
CS 434 Theory of Computation  
CS 442 Database Management Systems  
CS 488 Computer Architecture  
CS 492 Operating Systems  
CS 496 Programming Languages  
CS 551 Software Engineering and Practice I  
CS 552 Software Engineering and Practice II

Students who take the Honors sequence CS 181/182 are not required to take CS 115/284/385.

### **Cybersecurity**

CS 503 Discrete Mathematics for Cryptography  
CS 573 Fundamentals of Cybersecurity  
CS 576 Secure Systems  
CS 577 Cybersecurity Laboratory  
CS 578 Privacy in a Networked World  
CS 579 Foundations of Cryptography

### **Management**

BT 131 Technogenesis: Introduction to Innovation and Creativity

### **Humanities**

Students must take at least eight humanities courses, following the undergraduate core requirements. Humanities courses must include HSS 371 (Computers and Society).

### **Physical Education**

Students must take at least six semesters of physical education.

## **Computer Science and Security Electives**

The basic study plan has four unspecified technical electives. Three of these must be computer science (CS) classes, and at least one of which must be a security elective (see the list below). The basic study plan has one free elective.

For students requiring CS 105 in Term I, the number of technical electives is three. Two of these must be computer science classes and at least one must be a security elective. There remains one free elective.

For students taking the accelerated introductory sequence CS 181/182, two technical electives must be computer science classes, and two must be security electives. This track includes a second free elective.

### **Security Electives**

The following CS and non-CS courses qualify as security electives. Note that undergraduates must meet minimum GPA requirements to enroll in 600-level courses.

CS 665 Cybersecurity Forensics  
CS 693 Cryptographic Protocols  
CS 694 E-Business Security and Information Assurance

### **CPE 592 Multimedia Network Security**

The objective of this course is to introduce current techniques in securing IP and multimedia networks. Topics under IP security will include classic cryptography, Diffie-Hellman, RSA, end-to-end authentication, Kerberos, viruses, worms, and intrusion detection. Topics from multimedia will include steganography, digital watermarking, covert channels, hacking, jamming, security features in MPEG-4, secure media streaming, wireless multimedia, copy control, and other mechanisms for secure storage and transfer of audio, image, and video data.

### **MGT 662 Legal Issues for the IT Professional**

The course is a study of every major area of law that has an impact on the world of electronic commerce. The course discusses basic commercial law as it applies to electronic commerce, jurisdictional issues, and the contracting environment for on-line activity, including UCITA, intellectual property law, domain names, the protection of databases, privacy and publicity rights, and government regulation of E-commerce, including content based restrictions, criminal law, and the prospective taxation of E-commerce. The goal of the course is to provide basic background in these issues for non-lawyers and to promote sensitivity to the technological and business scenarios in which legal issues arise, enabling better management of their technological resources and commercial opportunities. Prerequisites: MGT 679.

### **MIS 645 Cybersecurity Principles for Managers**

This comprehensive course will cover the key security concepts for managers. In the first phase, security fundamentals will be covered, with emphasis on levels of security (network, system software, middleware, applications, and business processes), authentication, authorization, access, and integrity. In the second phase, the key security technologies, such as cryptographic algorithms (symmetric and asymmetric encryption), PKI, digital certificates, and corporate security, will be discussed. The last phase of this course will discuss the management issues of security policies and security administration and describe how various security technologies and approaches can be applied to cybersecurity. Topics will include an overview Internet security, Web security, Web application security, wireless and mobile Web security, and other emerging cyber information issues. Students will a conduct security audit of Web sites and Web-based corporate applications. Prerequisites: Students should have had exposure to network architecture, data architecture, and application architecture prior to taking this course by completing the following: MGT 772, MGT 773, and MGT 776 or equivalent.

### **MIS 646 Enterprise Architectures for Information Assurance**

This course focuses on the analysis and management of Information Security Architectures. Information Security Architectures consist of organizational, process, and technology (e.g., data, applications, network, and systems) domains. The integration and effective management of such architectures is essential to effectively respond

to technical risk dynamics. The course will focus on evaluating the architectural domains and their integration. The course will rely on management research on information security, risk, IT strategic planning, and distributed computing. The student will learn the relationships between business requirements, technical requirements, and technical risk, and make appropriate choices for risk mitigation. The course will provide insights on the continuous management of the information security function in organizations. Prerequisite: MIS 645.

[back to top](#)

### **Bachelor of Science in Information Systems**

This B.S. program is designed for those students who would like to acquire the background needed to apply information technology to support the major functions of a business or public institution. The management of information resources is as important as managing the capital, labor, and raw materials of any industry. Information systems are needed to manage the collection, manipulation, storage, distribution, and utilization of an organization's information.

People who manage information systems must have both business and technical knowledge. They must understand organizational structure, objectives, and operations. They must understand systems development tools, information architecture, network configurations, databases, and systems integration.

The curriculum above combines a strong background in the fundamentals of computer science with basic business management courses.

#### **Science**

CH 115 Introduction to Chemistry or PEP 111 Mechanics  
CH 117 Chemistry laboratory or PEP 221 Physics  
Laboratory  
CH 281 Biology and Biotechnology or PEP 112 Electricity  
and Magnetism  
CH 282 Biology laboratory or PEP 222 Physics Laboratory

#### **Math and Statistics**

MA 115 Calculus I  
MA 116 Calculus II  
MA 134 Discrete Mathematics  
MA 222 Probability and Statistics  
MA 331 Intermediate Statistics

#### **Basic Computer Science**

CS 105 Introduction to Scientific Computing  
CS 115 Introduction to Computer Science  
CS 284 Introduction to Data Structures 3  
CS 385 Algorithms

#### **Professional Computer Science Core**

CS 392 Systems Programming  
CS 393 Web Programming  
CS 442 Database Management Systems (Prerequisite: CS

385)  
 CS 551/552 Senior Project I & II  
 CS 573 Fundamentals of Cybersecurity  
 CS 578 Privacy in a Networked World

### Management

MGT 111 Organizational Behavior  
 MGT 244 Microeconomics  
 BT 101 Introduction to Business Planning  
 BT 102 Diagnosing Internal Capabilities of a Company  
 BT 113 Marketing  
 BT 115 Financial Accounting  
 BT 121 IT and Communication  
 BT 414 eTechnology Infrastructure  
 BT 421 Systems Analysis and Design  
 BT 460 Information Management

### Computer Science Electives

Some new and some undergraduate revisions of graduate courses.

CS 492 Operating Systems  
 CS 533 Cost Estimation and Metrics (Prerequisite: CS 551)  
 CS 535 Financial Computing  
 CS 549 Distributed Systems (Prerequisite: CS 492)  
 CS 564 Software Requirements Acquisition and Analysis (Prerequisite: CS 551)  
 CS 565 Software Architecture and Component-Based Design (Prerequisite: CS 551)  
 CS 567 Software Testing, Quality Assurance and Maintenance (Prerequisite: CS 551)

The course sequence for the B.S. in Information Systems is as follows:

#### Freshman Year

##### Term I

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MA 115	Calculus I	3	0	3
CS 105	Introduction to Scientific Comp.	2	2	3
BT 101	Intro to Bus. Plan	3	0	3
HUM	Humanities	3	0	3
PE 115	Physical Education I	0	2	1
<b>TOTAL</b>		<b>11</b>	<b>4</b>	<b>13</b>

##### Term II

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred		
MA 116	Calculus II	3	0	3

CS 115	Intro. to Computer Science	3	2	4
BT 102	Diagnosing Internal Capabilities	3	0	3
BT 121	IT and Comm.	3	0	3
PE 200	Physical Education II	0	2	1
<b>TOTAL</b>		<b>12</b>	<b>4</b>	<b>14</b>

**Sophomore Year****Term III**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
MGT 111	Organizational Behavior	3	0	3
HUM	Humanities	3	0	3
MA 134	Discrete Mathematics	3	0	3
CS 284	Intro. to Data Structures	3	1	4
	Science I	3	0	3
PE 200	Physical Education III	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>17</b>

**Term IV**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred		
BT 115	Financial Accounting	4	0	4
BT 421	Systems Analysis and Design	3	0	3
BT 113	Marketing	3	0	3
CS 385	Algorithms	3	1	4
MA 222	Probability	3	0	3
PE 200	Physical Education IV	0	2	1
<b>TOTAL</b>		<b>16</b>	<b>3</b>	<b>18</b>

**Junior Year****Term V**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
HSS 371	Comp. and Society	3	0	3
MA 331	Intermediate Statistics	4	0	4
CS 393	Web Programming	3	0	3
MGT 244	Microeconomics	3	0	3
CS 578	Privacy in Networked World	3	0	3
PE 200	Physical Education V	0	2	1
<b>TOTAL</b>		<b>16</b>	<b>2</b>	<b>17</b>

**Term VI**

Hrs. Per Wk.

		Class	Lab	Sem.
		Cred		
HUM	Humanities	3	0	3
	Science II	3	3	4
CS 392	Systems Programming	3	3	4
CS 442	Databases	3	1	4
	Elective	3	0	3
PE 200	Physical Education VI	0	2	1
<b>TOTAL</b>		<b>15</b>	<b>9</b>	<b>19</b>

**Senior Year****Term VII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred.		
IS Elective	Elective*	3	0	3
CS 551	Senior Project I	3	0	3
BT 414	eTechnology Infrastructure	3	0	3
CS 573	Cybersecurity	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

**Term VIII**

		Hrs. Per Wk.		
		Class	Lab	Sem.
		Cred		
HUM	Humanities	3	0	3
CS 552	Senior Project II	3	0	3
IS Elective	Elective*	3	0	3
BT 460	Information Management	3	0	3
HUM	Humanities	3	0	3
<b>TOTAL</b>		<b>15</b>	<b>0</b>	<b>15</b>

Students must take eight Humanities courses, including four 100-level courses (two in Group A - Literature or Philosophy and two in Group B - History and Social Sciences), and four upper division electives (300 or 400 level). HSS 371 (Computers and Society) counts as a Humanities Elective. MGT 244 (Microeconomics) counts as either a Humanities Elective or a lower division Group B Humanities course. IS Elective is either CS or BT.

There are two options for science courses:

1. Physics option:
  - (a) Science I: PEP 111 (Mechanics)
  - (b) Science II: PEP 112/222 (Electricity and Magnetism and Physics Lab)



2. Chemistry and Biology option:
  - (a) Science I: CH 115 (Chemistry)
  - (b) Science II: CH281/282 (Bio. Tech. and lab)

**Credit Totals: 125-22-127**

### **Interdisciplinary Programs**

For students interested in interdisciplinary science and engineering, Stevens offers an undergraduate computational science program. Computational science is a new field in which techniques from mathematics and computer science are used to solve scientific and engineering problems.

[back to top](#)

### **Laboratories**

Laboratories in the Department of Computer Science are used for course-related teaching and special problems, design projects, and research. You are encouraged, and in many cases required, to gain hands-on experience with a number of different computers and computer operating systems and environments. The emphasis is to provide a broad understanding of the entire operation of the computer and its peripherals, and to teach you the use of modern tools. You are also exposed to a range of practical problems in laboratory assignments.

Research laboratories are also heavily involved in the undergraduate education for special course-related projects and for senior design. Laboratory facilities provide the latest equipment and computational facilities (hardware and software), and the department is continually involved in expanding, improving, and developing new laboratories to provide you with the latest tools. For example, the Computer Science Laboratory has a collection of SGI, Sun, and PC workstations.

### **GRADUATE PROGRAMS**

Five types of graduate degrees are offered:

1. Master of Science in Computer Science
2. Master of Science in Quantitative Software Engineering
3. Master of Science in Information Systems
4. Ph.D. in Computer Science
5. Graduate Certificate

#### **1. Master of Science - Computer Science**

The departmental requirement for admission into the master's degree program in computer science is a bachelor's degree in computer science or computer engineering with a minimum grade point average (GPA) of 3.0 on a four-point scale. Applicants whose undergraduate degree is not in computer science or computer engineering must take the GRE General Test. In addition, such applicants must have completed at least one year of calculus and at least one programming course; they may be conditionally admitted subject to the completion of some or all of the Elements of Computer

Science "ramp" courses (listed below) with a grade of B or better.

### **Elements of Computer Science**

- CS 550 Computer Organization and Programming\*
- CS 580 The Logic of Program Design\*
- CS 590 Introduction to Data Structures and Algorithms\*
- MA 502 Mathematical Foundations of Computer Science\*

\*These courses may not be applied toward a graduate degree in Computer Science.

International students must demonstrate their proficiency in English by scoring at least 550 (or 210 for the computer-based test) on the TOEFL. Applicants desiring financial assistance are required to submit all application documents to Office of Graduate Admissions before February 1 for fall admission, and before September 1 for spring admission.

All candidates for the Master of Science in Computer Science must complete the following core courses, plus six electives approved by the graduate academic advisor. Students choosing a concentration must complete the concentration courses listed below, plus elective courses (approved by their graduate academic advisor) to fulfill the requirements for the degree. Student must complete 30 credits (10 courses) of course work and earn a GPA of 3.0 or better.

### **Core Courses**

- CS 510 Theory of Programming Languages
- CS 514 Computer Architecture
- CS 520 Introduction to Operating Systems
- CS 600 Analysis of Algorithms

### **Computer Science Concentrations**

A student may choose one of the concentrations listed below or select electives from other areas of specialization with the approval of the academic advisor.

#### **CyberSecurity Concentration**

- CS 503 Discrete Mathematics for Cryptography
- CS 573 Fundamentals of CyberSecurity
- CS 666 Information Networks I
- CS 668 Foundations of Cryptography
- CS 693 Cryptographic Protocols
- CS 694 E-Business Security and Information Assurance

Upon successful completion, the student will meet the requirements for a Master of Science in Computer Science (with a concentration in CyberSecurity indicated on the Transcript) and a Graduate Certificate in CyberSecurity.

#### **Quantitative Software Engineering Concentration**

- CS 533 Cost Estimation and Metrics
- CS 540 Fundamentals of Quantitative Software Engineering

and any two of the following courses:

- CS 552 Software Engineering and Practice II

CS 564 Software Requirements Acquisition and Analysis  
CS 565 Software Architecture and Component-Based Design  
CS 567 Software Testing, Quality Assurance, and Maintenance  
CS 687 Engineering of Large Software Systems  
CS 689 Software Reliability Engineering

Upon successful completion, the student will meet the requirements for a Master of Science in Computer Science (with a concentration in Quantitative Software Engineering indicated on the Transcript) and a Graduate Certificate in Quantitative Software Engineering.

[back to top](#)

## **2. Master of Science - Quantitative Software Engineering**

The M.S. program in Quantitative Software Engineering emphasizes both practical software technology and the practical skills needed to apply software technology to the realization of high-quality software products on time and within budget. It is case-history and project oriented, and teaches industry-standard practices and tools, including the use of software process metrics. It is intended for those with more practical goals, as practicing software developers, as software managers or as software entrepreneurs. Students interested in developing new technologies or in technological or theoretical research should consider the M.S. in CS program.

### **Admission Requirements**

Students must have an undergraduate degree with a GPA of 3.0 or better in Computer Science or Computer Engineering, or alternatively, an undergraduate degree in another field and on-the-job experience in software development. A solid working knowledge of a programming language is mandatory; C++ is the preferred language. Letters of recommendation should be specific as to the applicant's achievements rather than generally laudatory. Students with no software experience may be admitted subject to taking up to four of the Elements of Computer Science courses found earlier in the section (also referred to as 'ramp' courses.) They may also be required to take a course in C++ for credit.

### **Degree Requirements**

CS 533 Cost Estimation and Metrics  
CS 540 Fundamentals of Quantitative Software Engineering  
CS 564 Software Requirements Acquisition and Analysis  
CS 565 Software Architecture and Component-Based Design  
CS 567 Software Testing, Quality Assurance, and Maintenance  
CS 568 Software Project I  
CS 569 Software Project II  
Elective I  
Elective II  
Elective III

The three electives, chosen with the approval of the advisor, may be in software technology or in management. Recommended software technology topics include database management systems,

distributed systems, system performance analysis, etc. Recommended management topics include project management, group dynamics, etc. Students should choose electives for which they have the stated prerequisites, some of which may not be among the required QSE courses. The 10 courses must be completed with a GPA of 3.0 or better.

[back to top](#)

### **3. Master of Science - Information Systems (Interdisciplinary) with Computer Science, E-Commerce, Information Security, Integrated Information Architecture, or Quantitative Software Engineering Concentrations**

This program is designed to meet the increasing need for information technology professionals with both managerial and technical skills. It is an interdisciplinary program involving the Wesley J. Howe School of Technology Management and the Computer Science department. For additional information about the Master of Science - Information Systems program, please refer to the **Wesley J. Howe School of Technology Management** section of the catalog.

In addition to strong practical, real-world IT and management skills, graduates of the program leave with improved communication, interpersonal, and team skills. The MSIS is a professional degree that integrates information and organizational cultures, with emphasis on IT professionals who can contribute to the business.

Degree requirements include 12 graduate courses in each of the interdisciplinary MSIS tracks (36 credits) with a minimum GPA of 3.0. All require a bachelor's degree in Information Systems, Management, or Computer Science. Work experience is considered when reviewing educational background. Students considering doctoral study are encouraged to complete a master's thesis as part of their degree.

#### **Core Courses**

- MGT 609 Introduction to Project Management
- MGT 623 Financial Management
- MGT 680 Organizational Behavior and Theory
- MIS 620 Analysis and Development of Information Systems
- MIS 730 Integrating IS Technologies
- MIS 750 Management of IT Organizations
- MIS 760 Information Technology Strategy

#### **Computer Science Track Concentration Courses**

- CS 533 Cost Estimation and Metrics
- CS 540 Fundamentals of Quantitative Software Engineering
- CS 561 Database Management Systems I
- CS 666 Information Networks I

The typical admission profile includes information systems technical career advancement, 3+ years information technology experience and a bachelor's in information systems or computer science. A strong mathematical and technical background is recommended.

#### **E-Commerce Technical Track Concentration Courses**

Core courses for this concentration are found in the **Wesley J. Howe School of Technology Management** section of the catalog.

Select 3 from:

- CS 533 Cost Estimation and Metrics
- CS 537 Interactive Computer Graphics I
- CS 561 Database Management Systems I
- CS 619 E-Commerce Technologies
- CS 636 Integrated Services - Multimedia

#### **Information Security Concentration Courses**

- MIS 645 CyberSecurity Principles for Managers
- MIS 646 Enterprise Architecture for Information Security
- CS 573 Fundamentals of CyberSecurity
- CS 694 E-Business Security and Information Assurance

Select one from:

- MIS 620 Analysis and Development of Information Systems
- MIS 630 Data and Knowledge Management
- MIS 640 Network Management

#### **Integrated Information Architecture Track Concentration Courses**

- CS 561 Database Management Systems I
- NIS 560 Introduction to Networked Information Systems
- NIS 611 Digital Communications Engineering
- MIS 630 Data and Knowledge Management

And, select one from the following:

- CPE 592 Multimedia Network Security
- CPE 654 Design and Analysis of Network Systems
- CS 636 Integrated Services - Multimedia
- CS 666 Information Networks I

#### **Quantitative Software Engineering Track Concentration Courses**

- CS 533 Cost Estimation and Metrics
- CS 540 Fundamentals of Quantitative Software Engineering
- CS 564 Software Requirements Acquisition and Analysis
- CS 565 Software Architecture and Component-Based Design
- MIS 630 Data and Knowledge Management

Typical admission profile includes application systems analysis or testing career advancement, 3+ years information technology experience, and a bachelor's degree in information systems or computer science. A strong mathematics and technical background is recommended.

[back to top](#)

#### **4. Doctoral Program**

Admission requirements to the Ph.D. program are naturally more stringent than those for the lesser degrees. To be admitted to the doctoral program, a student must have a master's degree in either computer science or computer engineering and a minimum GPA of 3.5 on a 4.0 scale. More attention is paid to the student's background and potential to perform independent research. All applications are considered individually. Exceptional students may be accepted after receiving the bachelor's degree. All Ph.D. applicants are required to take the GRE general exam and are strongly encouraged to take the CS subject examination. International students, except those from English-speaking countries, must submit a minimum TOEFL score of 550 (written) or 210 (electronic).

Applicants are required to submit all application documents to Office of Graduate Admissions before February 1 for fall admission and before September 1 for spring admission.

The Ph.D. degree requires 90 credits. A maximum of 30 credits can be applied toward the 90-credit requirement of the Ph.D. degree from a previous master's degree or from any other graduate courses, subject to the approval of the department. All Ph.D. candidates must take at least 30 credits of thesis work and at least 21 credits of course work at Stevens beyond the master's degree.

All Ph.D. candidates must pass the Ph.D. qualifying examination, which has two phases: a written examination and a project. Students may take the written examination only twice. Failure to pass the written examination in the second attempt will result in dismissal from the Ph.D. program. In addition, there is a required proposal defense and dissertation defense. For a detailed description of the requirements, request a copy of the doctoral requirements from the Department of Computer Science and the Graduate Student Handbook from the Office of Graduate Admissions.

After the student has successfully completed the qualifying and preliminary examinations, s/he must choose an advisor. The advisor must be a full-time Computer Science faculty member. However, an outside co-advisor may be chosen subject to the approval of the Ph.D. program committee and the department director.

Once a suitable topic has been found and agreed upon with the advisor, the student must prepare a thesis proposal. The proposal must indicate the direction the thesis will take and procedures that will be used to initiate the research. Ordinarily, some preliminary results are included in the proposal. In addition, the proposal must indicate that the student is familiar with the research literature in his/her area. To this end, the proposal must include the results of a thorough literature search. A committee of at least three faculty members must accept the written dissertation proposal. The committee chairperson is the dissertation advisor. The other two members should be Computer Science department faculty. After the written proposal has been accepted, the examination committee conducts an oral defense. At this defense, the student presents his/her proposal.

All Ph.D. candidates who are working on a dissertation must have a doctoral dissertation advisory committee, chaired by the dissertation advisor, consisting of at least four members. The dissertation advisor and at least two other members must be full-time faculty of the Computer Science department. The committee may include members from other institutions. In addition, a regular faculty member with another department at Stevens must also be a member of the committee. The committee must approve the completed dissertation unanimously. After the dissertation has been approved, it must be defended in a seminar open to the public.

[back to top](#)

## **5. Graduate Certificate Programs**

The Computer Science department offers graduate certificate programs to students meeting the regular admission requirements for the master's program. Each certificate program is self-contained and highly focused, comprising 12 or more credits. Most of the

courses may be used toward the master's degree, as well as for the certificate. Those who enroll in the Database Systems program are expected to have a working knowledge of C++; students who do not have such knowledge will be admitted, but are required to take CS 590 as a prerequisite.

**Computer Graphics**

- CS 537 Interactive Computer Graphics I
- CS 558 Computer Vision
- CS 600 Analysis of Algorithms
- CS 638 Interactive Computer Graphics II

**Cybersecurity**

- CS 503 Discrete Mathematics for Cryptography
- CS 573 Fundamentals of Cybersecurity
- CS 668 Foundations of Cryptography
- CS 693 Cryptographic Protocols
- CS 694 E-Business Security and Information Assurance

**Database Systems**

- CS 561 Database Management Systems I
- CS 562 Database Management Systems II
- CS 563 Object-Oriented Database Management Systems
- CS 674 Theory of Object-Oriented Software Design

**Design of Trustworthy Software**

- CS 533 Cost Estimation and Metrics
- CS 567 Software Testing, Quality Assurance, and Maintenance
- CS 643 Formal Verification of Software
- CS 666 Information Networks I
- CS 689 Software Reliability Engineering
- CS 694 E-Business Security and Information Assurance

**Distributed Systems**

- CS 520 Introduction to Operating Systems
- CS 521 TCP/IP Networking
- CS 549 Distributed Systems
- CS 666 Information Networks I

**Quantitative Software Engineering**

- CS 533 Cost Estimation and Metrics
- CS 540 Fundamentals of Quantitative Software Engineering

and any two of the following courses:

- CS 552 Software Engineering and Practice II
- CS 564 Software Requirements Acquisition and Analysis
- CS 565 Software Architecture and Component-Based Design
- CS 567 Software Testing, Quality Assurance, and Maintenance
- CS 687 Engineering of Large Software Systems
- CS 689 Software Reliability Engineering

**Security Management and Forensics**

- CS 573 Fundamentals of Cybersecurity
- CS 695 Cybersecurity Forensics
- CS/TM 694 E-Business Security and Information Assurance
- MGT 710 Risk Management: Methods and Applications

**Software Design**

- CS 565 Software Architecture and Component-Based Design
- CS 567 Software Testing, Quality Assurance, and Maintenance
- CS 600 Analysis of Algorithms

CS 765 Selected Topics in Computer Science C++

### **Theoretical Computer Science**

CS 600 Analysis of Algorithms  
CS 601 Algorithmic Complexity  
CS 630 Automata and Formal Languages  
CS 634 Decidability and Computability

### **Elements of Computer Science**

CS 550 Computer Organization and Programming\*  
CS 580 The Logic of Program Design\*  
CS 590 Introduction to Data Structures and Algorithms\*  
MA 502 Mathematical Foundations of Computer Science\*

\*These courses may not be applied toward a graduate degree in computer science.

[back to top](#)

## **LABORATORIES**

Laboratories in the Department of Computer Science are used for course-related teaching and special problems, design projects, and research. Students are encouraged, and in many cases required, to gain hands-on experience with a number of different computers and computer operating systems and environments. The emphasis is to provide a broad understanding of the entire operation of the computer and its peripherals, and to instruct students in the use of modern tools. Students are exposed to a range of practical problems in laboratory assignments.

Research laboratories are also heavily involved in both undergraduate and graduate education with special projects, and dissertation projects. All research laboratories serve a dual-use function. Undergraduate students use these facilities through special course-related projects and for senior design. Graduate students pursue course-related projects and thesis research.

The laboratory facilities provide the latest equipment and computational facilities (hardware and software). The department is continually involved in expanding, improving, and developing new laboratories to provide the students with the latest tools. Both undergraduate and graduate curriculums are continually reviewed and updated to insure that we provide the current methodologies and technologies and to maintain the standards that are incorporated in Stevens' goals.

### **Computer Science Laboratory**

The Computer Science Laboratory offers students access to some 25 workstations running NetBSD, a free, secure, and highly portable UNIX-like open source operating system. These workstations are available to the entire Stevens community for education, research, and even recreation. In addition, many courses make use of the lab for programming purposes. The Computer Science department also maintains a state-of-the-art clustered High Performance Computing Facility (HPCF) suitable for research in areas of computer science that may require substantial computational effort. HPCF consists of 30 computing nodes, each with two 1GHz processors and 2GB of memory, communicating with each other over a gigabit switched network. The nodes are connected to a dual-processor I/O server



with two 1.5GHz processors and 3GB of memory, providing access to approximately 1.2TB of fast redundant fault-tolerant disk storage.

### **Computer Vision Laboratory - Dr. Elli Angelopoulou**

The primary objective of the research performed in the Computer Vision Laboratory is to apply rigorous physical and mathematical principles towards image interpretation. The work performed in the lab is multidisciplinary, combining many diverse academic disciplines, including physics, mathematics, engineering, and, above all, computer science. Some of the major thrusts in the lab include photometry, 3-D-shape reconstruction, shape analysis, object recognition, and multispectral imaging.

The Computer Vision Laboratory offers students hands-on experience with image capturing and processing equipment. There is a dedicated workstation that is mainly used for the capture of still images and movies. The environmental conditions in the lab are strictly controlled. If needed, the lab can become a dark room. An optic table allows for the precise positioning of equipment. A collection of optical components allows for experimentation with enhanced image capture. The lab has its own server and UNIX workstations for storing, processing, and analyzing images.

One of the industrial sponsors of the lab is Siemens Corporate Research, whose Imaging and Visualization Department has already donated a wide collection of optical equipment. There is also ongoing collaboration with the General Robotics Automation and Sensory Perception (GRASP) Laboratory at the University of Pennsylvania.

### **Laboratory for Secure Systems - Dr. Adriana Compagnoni, Dr. Dominic Duggan, Dr. David Naumann, Dr. Susanne Wetzels, Dr. Rebecca Wright**

The Lab's mission is to pioneer new technologies for high-assurance and secure systems and prototype tools that can provide guarantees that a system will not exhibit unpredictable behavior in a hostile environment. Our objective is to consolidate and organize research and tool-building efforts already under way at Stevens. The Lab is funded by grants from the New Jersey Commission on Science and Technology, the National Science Foundation, and the Stevens Institute of Technology Technogenesis Fund. The facilities of the Lab include several desktop machines, PDAs with wireless Ethernet, and Bluetooth devices for experimentation. The Lab is affiliated with the New Jersey Institute for Trustworthy Enterprise Software.

Part of the research work is focused on building better trust models for components. Some of this work is using static analysis techniques to check access control and information flow properties for untrusted components. There is also work on pushing type safety from high-level languages down to the assembly language level, and, in the process, checking properties of heap space usage. Other work has been performed on type systems for dynamic linking and "hot" updates of program libraries at run-time.

Another thrust of the work in the Lab has been in network security, particularly for wireless networks. Work continues on attacks that can be mounted on ad-hoc wireless networks, and in the design of new authentication and key establishment protocols that can be used to improve the security of wireless communication in general. Recent work has also looked at type-based approaches to

cryptography, to specify and ensure trustworthiness guarantees for communication channels.

### **Software Engineering Laboratory - Mr. Larry Bernstein, Dr. David Klappholz**

The Laboratory for Quantitative Software Engineering, supported by grant money from the New Jersey Commission on Science and Technology, has six workstations, of which five are dual bootable to either Windows 2000 or SuSe Linux. All six workstations have AMD Athlon(tm) processors with 256MB RAM. The workstations are connectable as both Ethernet and wireless LANs.

The lab's use is two-fold; first, it is used by those students in the required two-semester Senior Design sequence whose projects are more profitably implemented on networked workstations than on personal laptops, with which all Stevens students are equipped. A special feature of the Senior Design course is that it is taught using a novel pedagogic methodology entitled Live-Thru Case Histories. Further development of, and the study of the efficacy of, the Live-Through Case History Method are being studied under grants from the New Jersey Commission on Science and Technology and the National Science Foundation, in part with the aid of custom software being developed in the lab.

Second, as affiliates of both Prof. Barry Boehm's University of Southern California Center for Software Engineering and of the DOD-sponsored CEBASE (Center for Experimentally-Based Software Engineering) led by Prof. Boehm and Prof. Victor Basili of the University of Maryland, the Software Engineering Lab's faculty are using the lab for experimentation in software engineering technologies and methodologies. Subjects of these studies include high-reliability software, methods for avoiding the need to perform full-scale defect detection and elimination, and modern agile software development practices pair programming, refactoring, etc.

### **Visualization Laboratory (VLab) - Dr. George Kamberov, Dr. H. Quynh Dinh**

The research in the VLab is in the general areas of visualization, computer graphics, and computer vision, with applications in medical imaging and diagnostics, cell biology, scientific computing, robotics, and computational finance. Current research projects include the development of new geometric methods and efficient computational algorithms for representation, recognition, and visualization of surface shapes and shape deformations, and for pre-compression data reduction in visual data communications.

The VLab is part of the mV2 (multimedia vision and visualization) group, and has close ties with the Vision Lab at Stevens.

### **New Jersey Center for Software Engineering**

Members of the Computer Science Department hold a large grant from the New Jersey Commission on Science and Technology (NJCS&T), focused on research in software engineering aspects of networks and distributed programming. This grant is held jointly with New Jersey Institute of Technology (NJIT) and Rutgers University, New Brunswick.

Based at Stevens, the New Jersey Center for Software Engineering (NJCSE) was founded in mid-2000 as the corporate outreach (technology transfer) arm of this research activity. Academic institutions affiliated with NJCSE are Stevens Institute of

Technology, New Jersey Institute of Technology, Rutgers University in New Brunswick, and Monmouth University.

NJCSE activities for 2004 included regular technical meetings with Stevens, Rutgers, and NJIT researchers and industry representatives. Technology transfer to strengthen the New Jersey software industry is a major goal of NJCSE. NJCSE offers companies state-of-the-art technical programs, periodic reports on the state-of-the-art in software engineering, and early access to some of the best graduating computer science students in New Jersey.

[back to top](#)

## **UNDERGRADUATE COURSES**

### **CS 105 Introduction to Scientific Computing**

**(2-2-3)**

This is a first course in computer programming for students with no prior experience. Students will learn the core process of programming: given a problem statement, how does one design an algorithm to solve that particular problem and then implement the algorithm in a computer program? The course will also introduce elementary programming concepts like basic control concepts (such as conditional statements and loops) and a few essential data types (e.g., integers and doubles). Exposure to programming will be through a self-contained user-friendly programming environment, widely used by the scientific and engineering communities, such as Matlab. The course will cover problems from all fields of science, engineering, and business.

### **CS 115 Introduction to Computer Science**

**(3-2-4)**

This is an introductory programming course using the Java language. The topics include: basic facts about object-oriented programming and Java through inheritance and exceptions; recursion; UML diagrams and how to read class diagrams; ethics in computer science; and some basic understanding about computer systems: the compile/link/interpret/execute cycle and data representation. Prerequisite: CS 105 or substantial high school exposure to C++, Java, or some other major imperative programming language.

### **CS 181 Introduction to Computer Science Honors I**

**(3-2-4)**

Getting acquainted with C++: data types, input and output, functions, writing simple C++ programs, flow control, Boolean expressions, decision statements, if/then, and switch/case. Loop operations, while, do/while, and for loops. Arrays and pointers. Defining structs and classes, constructors and destructors, and operator overloading using an example String class. Templates. Abstract data types: vectors, lists, stacks, queues, and priority trees with applications. Trees and simple sorting with searching algorithms. By invitation only. Students who complete this class are exempt from CS 115 and CS 284.

### **CS 182 Introduction to Computer Science Honors II**

**(4-0-4)**

An overview of the C++ language. Algorithm design and computational complexity. Abstract data types. Review of fundamental data structures: vectors, lists, stacks, and queues.

Trees and binary trees. Heaps and priority queues with applications. Dictionaries with applications. Implementation of dictionaries using binary search trees, AVL trees, and red-black trees. Sorting algorithms. Graphs and networks with applications. Graph implementations and depth-first and breadth-first searching with applications. Other graph algorithms. Students who complete this class are exempt from CS 385. Prerequisites: CS 181 and permission of the instructor.

### **CS 284 Data Structures and Algorithms (3-1-4)**

This is a course on standard data structures, including sorting and searching and using the Java language. The topics include: stages of software development; testing; UML diagrams; elementary data structures (lists, stacks, queues, and maps); use of elementary data structures in application frameworks (e.g., JFC); searching; sorting; and introduction to asymptotic complexity analysis. Prerequisite: CS 115 or CS 181. Corequisite: MA 134.

### **CS 335 Computational Structures (3-0-3)**

This course uses functional programming to study discrete mathematics, building on MA 134. It begins with an introduction to a small subset of a functional language used in the course to explore examples and implement some basic algorithms. Operations on relations are explored by the use of lists to represent finite sets, functions and relations, as well as operations of composition, transitive closure, direct image, etc. Some algebraic structures, such as monoids and semirings, are introduced; lists and relations are used as primary examples of monoids; and trees are introduced as inductive data types. Other topics: structural recursion and induction; abstract syntax as data type; interpreters for simple languages (monoid expressions, Boolean expressions, integer expressions, and relational expressions) without variable binding; inductively defined relations; transition rules for configurations of simple machine; partial orders and lattices; simple abstract interpretations of expressions; and equational logic. Prerequisite: MA 134.

### **CS 381 Switching Theory and Logical Design (3-0-3)**

Digital systems; number systems and codes; Boolean algebra; application of Boolean algebra to switching circuits; minimization of Boolean functions using algebraic, Karnaugh map, and tabular methods; design of combinational circuits; programmable logic devices; sequential circuit components; and design and analysis of synchronous and asynchronous sequential circuits. Cross-listed with CPE 358. Prerequisite: CS 115 or CS 181.

### **CS 383 Computer Organization and Programming (3-0-3)**

Basic structure of the stored program computer, addressing methods and program sequencing, instruction sets and their implementation, the CPU and microprogrammed control, input/output organization, peripherals and interfacing, and main memory. Detailed study of a small machine. Assignments are devoted to assembly language programming. Prerequisite: CS 181 or CS 284.

### **CS 385 Algorithms**

**(3-1-4)**

This is a course on more complex data structures, and algorithm design and analysis, using the C language. Topics include: advanced and/or balanced search trees; hashing; further asymptotic complexity analysis; standard algorithm design techniques; graph algorithms; complex sort algorithms; and other "classic" algorithms that serve as examples of design techniques. Prerequisite: CS 284.

**CS 392 Systems Programming****(3-0-3)**

Introduction to systems programming in C on UNIX. Students will be introduced to tools for compilation, dynamic linking, debugging, editing, automatic rebuilding, and version control. Some aspects of the UNIX system call interface will be studied, drawn from this list: process creation, signals, terminal I/O, file I/O, inter-process communication, threads, network protocol stacks, programming with sockets, and introduction to RPC. Style issues to be covered include: naming, layout, commenting, portability, design for robustness and debugability, and language pitfalls. X programming and GUI design will be covered, if time allows. Prerequisite: CS 385 or CS 182.

**CS 393 Web Programming****(3-0-3)**

Introduction to Internet programming. Covers how the Internet works, the basics of HTML, and how to create advanced web sites using PHP, MySQL, and JavaScript. The course also places importance on presentations and team work. Prerequisite: CS 385 or CS 182.

**CS 434 Theory of Computation****(3-0-3)**

Introduction to the mathematical theory of computation. A number of models of computation are considered, as well as their relation to various problem classes (e.g. solvable problems and polynomial time solvable problems). Topics relevant to computer science, such as the theory of formal languages and low complexity classes, are particularly emphasized. Prerequisites: CS 335 and MA 134.

**CS 437 Interactive Computer Graphics****(3-0-3)**

A comprehensive introduction to the field of Computer Graphics. Students study the conceptual framework for interactive computer graphics: transformations, viewing, shading, clipping, rasterization, curves and surfaces, and selected topics. OpenGL is used as an application-programming interface. Prerequisite: CS 385 or CS 182.

**CS 442 Database Management Systems****(3-1-4)**

Introduction to the design of relational databases and the use of standard relational query languages. Topics include: relational schemas; keys and foreign key references; relational algebra (as an introduction to SQL); SQL; Entity-Relationship (ER) database design; translating from ER models to relational schemas and from relational schemas to ER models; functional dependencies; and normalization. Prerequisite: CS 385 or CS 182.

**CS 482 Artificial Intelligence****(3-0-3)**

An introduction to the large and diverse field of artificial intelligence.

Topics include: problem solving by search and constraint satisfaction; alpha-beta search for two-player games; and logic and knowledge representation, planning, learning, decision theory, statistical learning, and computer vision. Prerequisite: CS 385 or 182.

### **CS 487 Digital System Design**

**(3-0-3)**

Digital design concepts, building blocks for digital systems, and digital system organization. Design of combinational logic circuits and arithmetic functions. Approaches to sequential circuit design, separation of data and control, and the algorithmic state machine. System controller design: system controllers using MSI/LSI circuits. Clocks and timing. Asynchronous systems design. Metastability and reliability. Cross-listed with CPE 487. Prerequisite: CS 381.

### **CS 488 Computer Architecture**

**(3-0-3)**

An introduction to the functional level structure of modern pipelined processors and the empirical and analytic evaluation of their performance. Topics include: empirical and analytic techniques for measuring performance (use of various means, Amdahl's Law, and benchmarks); tradeoff analysis; principles of instruction set design and evaluation (memory addressing, operations, types and sizes of operands, instruction set encoding, CISC vs. RISC, and related compilation issues); pipelining (basics, data hazards, and control hazards); and memory systems. Prerequisite: CS 383. Corequisite: MA 222.

### **CS 492 Operating Systems**

**(3-0-3)**

The use and internals of modern operating systems. Lectures focus on internals whereas programming assignments focus on use of the operating system interface. Major topics include: the process concept; concurrency and how to program with threads; memory management techniques, including virtual memory and shared libraries; file system data structures; and I/O. Prerequisites: CS 383 and either CS 385 or CS 182.

### **CS 494 Compiler Design**

**(3-0-3)**

Design and implementation of compilers and principles of languages translation. Each student implements a complete compiler for a small but substantial language. The stages of a compiler. Boot-strapping a compiler. Lexical analysis, regular expressions, and finite state machines. Syntactic analysis, context free grammars, and parsers. Semantic analysis, type checking, and symbol tables. Syntax-directed translation. Data flow analysis and peephole optimization. Code generation. Prerequisite: CS 385 or CS 182.

### **CS 496 Principles of Programming Languages**

**(3-0-3)**

An introduction to programming language design and implementation, with an emphasis on the abstractions provided by programming languages. Assignments include implementing one or more interpreters for simple procedural languages, in a functional language such as Scheme or ML. Recursive types and recursive functions; structural induction. Abstract data types. Abstract syntax. Implementing languages with interpreters. Static vs. dynamic

scoping, closures, state. Exceptions. Types: type-checking, type inference, static vs. dynamic typing. Object-oriented languages: classes and interfaces, inheritance and subtyping. Polymorphism and genericity. Prerequisites: CS 335, CS 434, and either CS 385 or CS 182.

### **CS 498 Senior Research I (0-8-3)**

Individual research project under the guidance of a faculty member of the department, whose prior approval is required. Either a written report in acceptable journal format or the completion of a senior thesis, as well as an oral presentation, is required at the end of the project. Senior students only. CS 498 and CS 499 cannot be taken simultaneously.

### **CS 499 Senior Research II (0-8-3)**

Individual research project under the guidance of a faculty member of the department, whose prior approval is required. Either a written report in acceptable journal format or the completion of a senior thesis, as well as an oral presentation, is required at the end of the project. Senior students only. CS 498 and CS 499 cannot be taken simultaneously.

[back to top](#)

## **GRADUATE COURSES**

All Graduate courses are 3 credits except where noted.

Undergraduate students may take any 500-level course for which they satisfy the prerequisites. Higher-numbered graduate courses may be taken only by undergraduates who satisfy the prerequisites, have a 3.0 GPA, and receive permission of the instructor. Note that prerequisites for graduate courses are stated in terms of other graduate courses, and that equivalences exist between certain undergraduate and graduate courses:

CS 510	Principles of Programming Languages	=	CS 496	Principles of Programming Languages
CS 514	Computer Architecture	=	CS 488	Computer Architecture
CS 516	Compiler Design	=	CS 494	Compiler Design
CS 520	Introduction to Operating Systems	=	CS 492	Introduction to Operating Systems
CS 537	Interactive Computer Graphics	=	CS 437	Interactive Computer Graphics
CS 550	Computer Organization and Programming	=	CS 383	Computer Organization and Programming
CS 561	Database Management Systems	=	CS 442	Database Management Systems
CS 570	Introduction to Programming	=	CS 115	Introduction to Computer Science

CS 580	Data Structures	=	CS 284	Data Structures and Algorithms
CS 590	Algorithms	=	CS 385	Data Structures and Algorithms II
MA 502	Mathematical Foundations of Computer Science	=	MA 134	Discrete Mathematics

In fulfilling their study plans, undergraduates should always take the undergraduate course whenever a choice exists.

### **CS 501 Introduction to JAVA Programming**

An introduction to the Java programming language for those students who have little or no programming background. It is intended as an elective for the Master of Science in Information Systems to be taken near the end of the program. Basic topics considered will be programs and program structure in general and Java syntax, data types, flow of control, classes, methods and objects, arrays, exception handling, and recursion. In addition, the use of Java in enterprise-wide computing and distributed systems will be introduced by considering APIs in general, and the ones specific to JDBC and the Java security features in particular. Not for credit for Computer Science department majors.

### **CS 503 Discrete Mathematics for Cryptography**

Topics include basic discrete probability, including urn models and random mappings; a brief introduction to information theory; elements of number theory, including the prime number theorem, the Euler phi function, the Euclidean algorithm, and the Chinese remainder theorem; and elements of abstract algebra and finite fields including basic fundamentals of groups, rings, polynomial rings, vector spaces, and finite fields. Carries credit toward the Applied Mathematics degree only when followed by CS 668. Recommended for high-level undergraduate students. Prerequisite: MA 502. Cross-listed with MA 503.

### **CS 505 Probability and Stochastic Processes I**

Axioms of probability. Discrete and continuous random vectors. Functions of random variables. Expectations, moments, characteristic functions, and moment-generating functions. Inequalities, convergence concepts, and limit theorems. Central limit theorem. Characterization of simple stochastic processes: wide-sense stationarity and ergodicity.

### **CS 510 Principles of Programming Languages**

Principles of functional, imperative, and object-oriented programming languages; elements of language theory; the typed- $\lambda$  calculus, functional languages, and stack implementation of recursion; imperative languages, block structure, and more on stack allocation model; user-defined types, heap storage model; and object-oriented languages, data abstraction, genericity, polymorphism, and inheritance. Case studies may include Algol, Pascal, Ada, LISP, Scheme, Smalltalk, Java, and C++. Prerequisites: Undergraduates: CS 335, CS 434, and either CS 385 or CS 182; Graduates: MA 502 and CS 590.

### **CS 511 Concurrent Programming**

The study of concurrency as it appears at all levels and in different types of computing systems. Topics include: models of concurrency;



languages for expressing concurrency; formal systems for reasoning about concurrency; the challenges of concurrent programming; race conditions; deadlock; livelock and nondeterministic behavior; prototypical synchronization problems, such as readers-writers and dining philosophers; mechanisms for solution of these problems, such as semaphores, monitors, and conditional critical regions; important libraries for concurrent programming; message passing, both synchronous and asynchronous; and applications of multithreaded concurrent programming and parallel algorithms. Substantial programming required. Prerequisites: CS 492, CS 520, or equivalent.

### **CS 514 Computer Architecture**

Measures of cost, performance, and speedup; instruction set design; processor design; hard-wired and microprogrammed control; memory hierarchies; pipelining; input/output systems; and additional topics as time permits. The emphasis in this course is on quantitative analysis of design alternatives. Prerequisite: CS 550 or equivalent. Cross-listed with CPE 514 and NIS 514.

### **CS 516 Compiler Design**

Lexical analysis; syntax analysis; symbol table construction; semantic analysis; syntax-directed translation; dataflow analysis; liveness analysis; and register allocation. The emphasis in this course is on the integration of the various parts of a compiler. Each student writes a complete compiler for a small, but substantial, language. Prerequisite: CS 510 or equivalent.

### **CS 520 Introduction to Operating Systems**

Overview of operating systems. CPU scheduling, process concept, suspend and resume, interrupt processing, and multiprogramming. Concurrent processes, the critical section problem, mutual exclusion, semaphores, process coordination, and communication and synchronization. CPU scheduling algorithms. Deadlocks, conditions for deadlock, deadlock detection, and avoidance. Memory management, memory management schemes-single contiguous, static and dynamic partitions, relocatable partitions, paging, demand paging, and page replacement strategies. Secondary storage management, disk scheduling concepts, and algorithms. Students will be given programming assignments on a regular basis. This course is not centered on any particular operating system or hardware; instead, it covers fundamental concepts that are applicable to a variety of systems. Prerequisites: CS 590 and CS 550 or their equivalents.

### **CS 521 TCP/IP Networking**

Introduction to IP networking. Examination of all layers of the OSI stack. Detailed examination of the IP, ICMP, UDP, and TCP protocols. Basic concepts of network design: end-to-end principle, routing, encapsulation, flow control, congestion control, and security. Detailed coverage of TCP. Some treatment of important Internet applications and services. Emphasis on network layer and above. Assignments focus on protocols and software. Prerequisite: CS 492 or CS 520.

### **CS 527 Logical Design of Digital Systems I**

Design concepts for combinational and sequential (synchronous and asynchronous) logic systems. The design processes are described algorithmically and are applied to complex function design at the

gate and register level. The designs are also implemented using software development tools-logic compilers for programmable logic devices and gate arrays. Cross-listed with CPE 643.

### **CS 528 Logical Design of Digital Systems II**

The design of complex digital logic systems using processor architectures. The architectures are implemented for reduced instruction set computers (RISC) and extended to complex instruction set computers (CISC). The emphasis in the course is the design of high-speed digital systems, and includes processors, sequencer/controllers, memory systems, and input/output. Prerequisites: CS 514, CS 527.

### **CS 533 Cost Estimation and Metrics**

The course deals with the management of software projects through the use of objective metrics, which help developers and managers to understand the scope of the work to be accomplished, the risks which will occur, the tasks to be performed, the resources and effort to be expended, and the schedule to be observed. It provides the student with a thorough introduction to facility with, and understanding of, such industry-standard software sizing metrics as Function, Feature and Object Points, and their relationship to the lines-of-code metric. It provides the student with a thorough introduction to, and understanding of, such industry-standard software estimation tools such as COCOMO II and KnowledgePlan. Prerequisites: CS 540 or CS 551.

### **CS 535 Financial Computing**

This is a course in modeling the values of assets and financial derivatives and the software implementation of these models for pricing, simulations, and scenario analysis. The course includes an introduction to markets and financial derivatives, and a development of the necessary tools from the theories of stochastic processes and parabolic differential equations. An integral part of the course is the use of financial information sources and software packages available on the Internet for modeling and analysis. Prerequisites: Acquaintance with Multivariable Calculus and programming in C++ and/or Java.

### **CS 537 Interactive Computer Graphics**

This is an introductory-level course to computer graphics. No previous knowledge on the subject is assumed. The objective of the course is to provide a comprehensive introduction to the field of computer graphics, focusing on the underlying theory, and thus providing strong foundations for both designers and users of graphical systems. The course will study the conceptual framework for interactive computer graphics, introduce the use of OpenGL as an application programming interface (API), and cover algorithmic and computer architecture issues. Cross-listed with CPE 537. Prerequisite: Graduates: CS 590; Undergraduates: CS 385 or CS 182.

### **CS 540 Fundamentals of Quantitative Software Engineering**

This course introduces the subject of software engineering, also known as software development process or software development best practice, from a quantitative, analytic-, and metrics-based point of view. Topics include introductions to: software life-cycle process models from the heaviest weight, used on very large projects, to the lightest weight, e.g. extreme programming; industry-standard

software engineering tools; teamwork; project planning and management; and object-oriented analysis and design. The course is case-history and project oriented. Prerequisites: Admission to the M.S. in CS program and completion of any required ramp courses, OR admission to the M.S. in QSE program. Undergraduates may take this course if they have senior status, or have obtained the written permission of the instructor. CS 540 and CS 551 may not both be taken. Cross-listed with CPE 540.

### **CS 549 Distributed Systems**

Developing robust applications in distributed environments. Coursework includes developing a fault-tolerant distributed application. RPC and RMI; Web Services; application servers (e.g., JEE and Websphere). Transactions: concurrency control and recovery, distributed transactions, nested transactions, and business transactions. Models of distributed systems, impossibility results, and Byzantine failures. Protocol design and examples (2PC and 3PC). Distributed snapshots. Logical time and vector clocks. Replication for fault tolerance: primary-backup and state machine approaches, quorum consensus, and process groups. Peer-to-peer networks. Prerequisite: CS 492 or CS 520.

### **CS 550 Computer Organization and Programming**

This course provides an intensive introduction to material on computer organization and assembly language programming required for entrance into the graduate program in Computer Science or Computer Engineering. The topics covered are: structure of stored program computers; linking and loading; assembly language programming, with an emphasis on translation of high-level language constructs; data representation and arithmetic algorithms; basics of logic design; and processor design: data path, hardwired control, and microprogrammed control. Students will be given assembly language programming assignments on a regular basis. No graduate credit for students in Computer Science or Computer Engineering. Cross-listed with CPE 550.

### **CS 551 Software Engineering and Practice I**

Software design and development theory of software design, with emphasis on large systems. Models of the software process: specifications development, designing, coding, and testing. Program abstraction with functional abstraction and with abstract data types. Top-down and bottom-up development methods. Common software architecture models. Specification validation, design verification, testing strategies, and test coverage issues. CS 551 and CS 540 may not both be taken. Prerequisite: CS 385 or CS 182.

### **CS 552 Software Engineering and Practice II**

Covers the practical aspects of the software process. Cost and schedule estimation; management systems; technical vs. administrative responsibilities; documentation; libraries; configuration management; automated software tools; and documentation. The class is partitioned into software teams, each of which develop and manage a typical large-scale software development process. Projects are drawn from industry and government. Prerequisite: CS 551.

### **CS 558 Computer Vision**

An introduction to the field of Computer Vision, focusing on the underlying algorithmic, geometric, and optic issues. The course

starts with a brief overview of basic image processing topics (convolution, smoothing, and edge detection). It then proceeds on various image analysis topics: binary images, moments-based shape analysis, Hough transform, image formation, depth and shape recovery, photometry, motion, classification, and special topics. Prerequisite: CS 385, CS 182, or CS 590; Corequisites: MA 112 and MA 115 or equivalent. Cross-listed with CPE 558.

### **CS 561 Database Management Systems**

Introduction to the use of relational database systems; the relational model; the entity-relationship model; translation of entity-relationship diagrams into relational schemes; relational algebra; SQL; and normalization of relational schemes. Students who have had a previous course in database systems must obtain permission of the instructor to enroll in this course. Prerequisite: CS 590 or equivalent.

### **CS 562 Database Management Systems II**

A continuation of CS 561. Review of the relational model, relational algebra, SQL, and normalization; storage organization; indexes; B-trees and B+-trees; query optimization; concurrency control; and recovery control. Prerequisite: CS 561 or equivalent.

### **CS 563 Object-Oriented Database Management Systems**

Introduction to fundamental principles of object-oriented database management systems. Topics include: review of the relational database model and its deficiencies; concepts underlying object-oriented methodology; application of object-orientation to database management systems; transaction processing; object-oriented databases under client/server architectures. Prerequisite: CS 561.

### **CS 564 Software Requirements Acquisition and Analysis**

Requirements Acquisition is one of the least understood and hardest phases in the development of software products, especially because requirements are often unclear in the minds of many or most stakeholders. This course deals with the identification of stakeholders, the elicitation and verification, with their participation, of the requirements for a new or to-be-extended software product. It deals further with the analysis and modeling of requirements, the first steps in the direction of software design. Finally, it deals with the quality assurance aspects of the software requirements phase of the software development process. This course is case-history and project-oriented, and uses industry-standard software tools. Prerequisite: CS 540 or CS 551.

### **CS 565 Software Architecture and Component-Based Design**

This course deals with the high-level (architectural) and low-level issues involved in the design of software systems/products. At the high level it deals with such issues as component-based design, cohesion, interconnection complexity, and methods for minimizing the latter; it also deals with the use of middleware, performance analysis and simulation, and the use of COTS components. At the low level, it deals with object-oriented design, design patterns, and code refactoring. Finally, it deals with validation and verification of both architecture and code designs. This course is case-history and project-oriented. Prerequisite: CS 540 or CS 551.

### **CS 567 Software Testing, Quality Assurance, and Maintenance**

This course provides in-depth coverage of software testing, configuration management, quality assurance, and maintenance, both in terms of defect removal and enhancement. It deals with the performance of these activities in a variety of life-cycle models, from spiral-model or Rational Unified Process development, through lighter weight varieties of incremental and iterative models down, to extreme programming. It is a project-oriented course. Prerequisite: CS 540 or CS 551.

### **CS 568-CS 569 Software Project I-II**

These two courses, which must be taken in sequence, constitute the capstone of the M.S. program in Quantitative Software Engineering. Students will use knowledge and skills gained in earlier courses to build real software products, in teams, for real stakeholders. Prerequisites: For CS 568: CS 540, CS 533, CS 564, CS 565, CS 567, and the ability to program in C++, which may be obtained by taking CS 580 as an elective. (C++ is the basic programming language which will be assumed for all projects, except in cases in which it is technically inappropriate.) Only in extraordinary cases will a student without all prerequisites be allowed to take this course, and then only with the written permission of the instructor. Prerequisite for CS 569: CS 568.

### **CS 570 Introduction to Programming in C++**

Standard data types and programming constructs of C++. Students will be given weekly programming assignments. Prerequisite: CS 385 or CS 590.

### **CS 571 Java**

The course consists of an in-depth discussion of Java language and programming techniques. Comparison of Java to other languages, such as C/C++, is made throughout the course to emphasize various shortcomings of the language and their implications on design paradigms. Some aspects of GUI libraries, multithreading support, and Java native interface are also discussed. Not for undergraduate credit in Computer Science, Computer Engineering, and Information Systems degree programs.

### **CS 573 Fundamentals of CyberSecurity**

This course studies the mathematical models for computer security (Bell-LaPadula, Clark-Wilson, Biba, and Gligor models). It analyzes and compares, with respect to formal and pragmatic criteria, the properties of various models for hardware, software, and database security. Topics also include: formal specification and verification of security properties, operating system security, trust management, multi-level security, security labeling, security auditing and intrusion detection, security policy, safeguards and countermeasures, risk mitigation, covert channels, identification and authentication, password schemes, access control lists, and data fusion techniques. The course includes a project. Prerequisite: CS 392 or CS 520.

### **CS 5xx Object-Oriented Analysis and Design**

Theory of object-oriented design, classes, interfaces, inheritance hierarchy, and correctness; abstract data types, encapsulation, formal specification with preconditions, postconditions and invariants, and proofs of correctness; object-oriented software, objects and classes, genericity, inheritance, polymorphism, and overloading; single and multiple inheritance, programming by contract, subclassing as subcontract, specification, and verification;

programming language examples include C++, Java, Smalltalk, and Eiffel.

### **CS 576 Secure Systems**

Attacks on computer systems have become part of everyday life. It is the goal of this class to teach a basic understanding of the possible security failures, as well as the protection mechanism. The class will cover an introduction to network and host security concepts and mechanisms; basic cryptographic algorithms and protocols; authentication and authorization protocols; access control models; common network (wired and wireless) attacks; typical protection approaches, including firewalls and intrusion detection systems; and operating systems and application vulnerabilities, exploits, and countermeasures. The class is designed for undergraduate, master's, and Ph.D. students. Those who take the class are expected to be able to program in C, have some basic knowledge of assembly language, and be familiar with network programming, as well as Unix-like operating systems. Prerequisites: CS 392, CS 494 or CS 516, CS 492 or CS 520, CS 579. Corequisite: CS 577.

### **CS 577 Cybersecurity Lab**

Theoretical foundations in cryptographic algorithms, cryptographic protocols, access control models, formal methods, security policy, etc. provide the necessary background to understand the real-world implications of cryptography and network security. This laboratory course is designed to provide students with a hands-on experience based on the theoretical knowledge they have acquired by taking other security-oriented courses. This hands-on experience is of great importance for future jobs in industry. The course will accomplish its goals through a number of in-lab programming exercises. Topics covered include: basic cryptographic algorithms and protocols; authentication and authorization protocols; access control models; common network (wired and wireless) attacks; typical protection approaches including firewalls and intrusion detection systems; and operating systems and application vulnerabilities, exploits, and countermeasures. The class will involve a fair amount of programming. Prerequisites: CS 392, CS 494 or CS 516, CS 492 or CS 520, CS 579. Corequisite: CS 576.

### **CS 578 Privacy in a Networked World**

Increasing use of computers and networks in business, government, recreation, and almost all aspects of daily life has led to a proliferation of online sensitive data: data that, if used improperly, can harm the data subjects. As a result, concern about the ownership, control, privacy, and accuracy of these data has become a top priority. This course focuses on both the technical challenges of handling sensitive data and the policy and legal issues facing data subjects, data owners, and data users. This course is suitable for advanced undergraduate computer science majors, graduate students in computer science, and students in technology management or other majors with some computer science background. Course readings draw on a variety of sources, including both technical materials and the popular press. The course includes a privacy-related project. Cross-listed with MIS 654.

### **CS 580 Data Structures**

This is a course on standard data structures, including sorting and searching and using the Java language. The topics include: stages of software development; testing; UML diagrams; elementary data

structures (lists, stacks, queues, and maps); use of elementary data structures in application frameworks (e.g., JFC); searching; sorting; and an introduction to asymptotic complexity analysis. Corequisite: MA 502. Cross-listed with CPE 580.

### **CS 590 Algorithms (3-0-3)**

This is a course on more complex data structures, and algorithm design and analysis, using the C language. Topics include: advanced and/or balanced search trees; hashing; further asymptotic complexity analysis; standard algorithm design techniques; graph algorithms; complex sort algorithms; and other "classic" algorithms that serve as examples of design techniques. Prerequisites: CS 580 and MA 502. Cross-listed with CPE 590.

### **CS 600 Analysis of Algorithms**

The complexity and correctness of algorithms: big oh, big omega, and big theta notations, recurrence relations and their solutions. Worst, average, and amortized analysis of algorithms with examples. Basic and advanced data structures for searching, sorting, compression, and graph algorithms. Students will be given programming assignments on a regular basis. Prerequisite: CS 590. Cross-listed with CPE 600.

### **CS 601 Algorithmic Complexity**

Analysis of algorithms: resource-bounded computation and time and space complexity. Various models of computation will be studied. Complexity classes and reducibilities, hardness, and completeness. Randomized algorithms and approximation algorithms. Prerequisite: CS 600.

### **CS 613 Knowledge Discovery and Data Mining**

This course introduces fundamental and practical tools, techniques, and algorithms for Knowledge Discovery and Data Mining (KD&DM). It provides a balanced approach between methods and practice. On the methodological side, it covers several techniques for transforming corporate data into business intelligence. These include: On-line Analytical Processing (OLAP) Systems, Artificial Neural Networks (ANN), Rule-Based Systems (RBS), Fuzzy Logic (FL), Machine Learning (ML), Classification Trees (C4.5 Algorithm), and Classification and Regression Trees (CART Algorithm). To illustrate the practical significance of the various techniques, half of the course is devoted to case studies. The case studies, drawn from real-world applications, demonstrate application of techniques to real world problems. A list of products, software, and vendors will be provided. Prerequisite: CS 505.

### **CS 617 Elements of Compiler Design II**

Optimization: dataflow analysis, copy propagation, dead code elimination, common subexpression elimination, code hoisting, elimination of redundant induction variables, compiling functional and object-oriented languages; control-flow analysis, and static single-assignment form intermediate languages. Additional topics at discretion of the instructor. Prerequisite: CS 516.

### **CS 619 E-Commerce Technologies**

The course provides an understanding of electronic commerce and related architectures, protocols, and technologies. It describes the e-commerce concept, objectives, and market drivers, as well as its

requirements and underpinning techniques and technologies, including the Internet, WWW, multimedia, intelligent agents, client-server, and data mining. Security in e-commerce is addressed, including types of security attacks, security mechanisms, Virtual Private Networks (VPNs), firewalls, Intranets, and extranets. Implementation issues in e-commerce, including the design and management of its infrastructure and applications (ERP, CRM, and SCM), are discussed. M-commerce is addressed; electronic payment systems with their associated protocols are described, and various B2C and B2B applications are presented. Also, policy and regulatory issues in e-commerce are discussed. Cross-listed with TM 619, CPE 619 and NIS 619. Prerequisite: CS 666, CPE 678, TM 610, or MGT 776.

### **CS 625 Foundations of Distributed Computing**

Design and analysis of distributed algorithms, and impossibility results showing when some problems are unsolvable. Models of synchronous and asynchronous distributed computing. Fault models, including crash failures and malicious failures, and communication models, including message passing and shared memory systems. Distributed algorithms and impossibility results for problems, such as consensus, Byzantine agreement, clock synchronization, mutual exclusion, and secure multiparty computation. Prerequisite: CS 600

### **CS 630 Automata and Formal Languages**

Analysis of finite automata and regular sets. Formal languages and grammars, and Chomsky-hierarchy. Context-free languages and PDAs. Applications to parsing. Prerequisite: MA 502 or equivalent.

### **CS 634 Decidability and Computability**

Computable functions and Turing machines. Primitive recursive functions, recursive functions, loop-programs, and while-programs. Decidability, solvability, and unsolvable problems. High complexity classes and reducibilities, hardness, and completeness. The arithmetical hierarchy and definability. The connection to complexity theory is emphasized throughout the course. Prerequisite: MA 502 or equivalent.

### **CS 636 Integrated Services - Multimedia**

Types of multimedia information: voice, data video facsimile, graphics, and their characterization; modeling techniques to represent multimedia information; analysis and comparative performances of different models; detection techniques for multimedia signals; specification of multimedia representation based on service requirements; and evaluation of different multimedia representations to satisfy user applications and for generating test scenarios for standardization. Prerequisite: EE 605, NIS 605, or CS 505. Cross-listed with CPE 636 and NIS 636.

### **CS 638 Interactive Computer Graphics II**

Mathematical foundations and algorithms for advanced computer graphics. Topics include 3-D modeling, texture mapping, curves and surfaces, physics-based modeling, and visualization. Special attention will be paid to surfaces and shapes. The class will consist of lectures and discussion on research papers assigned for reading. In class, we will study the theoretical foundations and algorithmic issues. In programming assignments, we will use OpenGL as the particular API for writing graphics programs. C/C++ programming skills are essential for this course. Prerequisites: CS 437, CS 537, or



equivalent. Cross-listed with CPE 638.

### **CS 639 Real-Time Rendering, Gaming, and Simulations Program**

The course is an introduction to the techniques for designing and building computer games and real-time graphics-oriented simulations. The topics include 3-D game engine architecture, design, and implementation; simulation, modeling, and object control; character behavior and behavior-based animation; human-computer interaction, and event-driven simulations. Prerequisites: CS 482, and CS 437 or CS 537.

### **CS 643 Formal Verification of Software**

Formal systems for specification and verification of software; review of the first-order predicate calculus; abstract data types, formal specification, preconditions, postconditions, invariants, predicate transformers, proofs of correctness, and partial and total correctness; correctness for assignments, alternatives, iterations, and procedure calls. Prerequisite: CS 600.

### **CS 651 Introduction to Network and Graph Theory**

Introduction to the theory and applications of networks and graphs. Topics include paths, connectivity, trees, cycles, planarity, network flows, matchings, colorings, and some extremal problems. Prerequisite: MA 502 or equivalent.

### **CS 655 Queuing Systems with Computer Applications I**

Queuing models will be developed and applied to current problems in telecommunication networks and performance analysis of computer systems. Topics include elementary queuing theory, birth-death processes, open and closed networks of queues, priority queues, conservation laws, models for time-shared computer systems, and computer communication networks. Prerequisite: EE 605, NIS 605, or CS 505. Cross-listed with CPE 655 and NIS 655.

### **CS 656 Queuing Systems with Computer Applications II**

This course is a continuation of CS 655. Prerequisite: CS 655, CPE 655, or NIS 655. Cross-listed with CPE 656 and NIS 656.

### **CS 660 Graph Algorithms**

Basic graph-theoretic notions; data structures for graph representation; running time analysis; review of depth-first search, breadth-first search, and minimum spanning trees; network flow problems; graph connectivity; matchings; Eulerian graphs and digraphs; de Bruijn graphs; Hamiltonian graphs; traveling salesman problem; planar graphs; planarity testing; vertex and edge colorings; chromatic polynomials; five-coloring algorithm; and the Four-color problem. Prerequisites: MA 502 or equivalent and CS 600.

### **CS 665 CyberSecurity Forensics**

Security forensics involves the identification, preservation, and analysis of evidence of attacks in order to identify the attackers and document their activity with sufficient reliability to justify appropriate technological, business, and legal responses. Forensic activity takes place in a complex, ever-changing legal and social context, which must be understood to fully appreciate the power and value of that activity. This course covers both technological and legal components of the topic. The technical aspect addresses analysis of intruder types and the intrusion process, review of logs and profiles and their

types, identification of attack signatures and fingerprints, application of data mining techniques, study of various traceback methods, and the preservation of the information and evidence acquired through the use of forensic tools and techniques. The legal aspect addresses the impact of forensics on the legislative, judicial, and regulatory proceedings that collectively articulate and promote public policy goals, determine civil and criminal liability, and define and assess regulatory compliance. It also discusses the point that the effective and appropriate use of forensics requires that such investigations be conducted under law, with particular regard to civil rights, rights of privacy, property, and contract rights. The course draws on pertinent concepts of law and legal procedure, at a level accessible by non-lawyers, to explore the broader significance of forensics in the private and public sectors. Prerequisite: CS 573 or MIS 645. Cross-listed with MIS 665 and TM 665.

### **CS 666 Information Networks I**

The first of a two-course sequence on modern computer networks. Focus is on the physical and data link levels of the OSI layers. Trace the evolution of client/server computing to the Internet. Topics covered include OSI layering, TCP/IP overview, the application of Shannon's and Nyquist's bandwidth theorems, Discrete Wave Division Multiplexing, wireless transmission, local loops, QAM, TDM, SONET/SDH, circuit switching, ATM switching, knockout switch, ISDN, STM, framing, error detection and correction, CRC, ARQ protocol, sliding window protocols, finite state machines, Universal Modeling Language, PPP, ALOHA, CSMA, LANs, fast and gigabit Ethernet, Bridges, and FDDI. A significant amount of time is spent on designing 802.3 LANs. Prerequisite: Graduate: CS 505; Undergraduate: MA 222. Cross-listed with NIS 678 and CPE 678.

### **CS 667 Information Networks II**

Learn the technologies that make the Internet work. You will understand the TCP and IP protocols and their interaction. You will study the TCP slow start in low noise and high noise environments, the use of proxy servers, web caching, and gain understanding of the technologies used to make routers perform well under load. These include shortest path routing, new routing protocols, TCP congestion control, leaky bucket and token bucket admission control, weighted fair queueing, and random early detection of congestion. Networks are described in terms of their architecture, transport, routing, and management. Quality of Service (QoS) models are integrated with communication models. The course requires problem-solving and extensive reading on network technology. After an introduction to bridges, gigabit Ethernet, routing, and the Internet Protocol, a fundamental understanding of shortest path and distance vector routing is taught. A "problem/solution" approach is used to develop how and why the technology evolved to keep engineering tradeoffs in focus. Continuation of Information Networks I with a focus on the network and transport layers of the OSI layers. Protocol definitions for distributed networks and performance analysis of various routing protocols, including Bellman-Ford, BGP, and OSPF. TCP/IP is discussed. Other topics include pipelining, broadcast routing, congestion control and reservations, Leaky and Token Bucket algorithms, weighted fair queueing, tunneling, firewalls, IPv4, and IPv6. Network layers in SAN including the different service categories are discussed. The TCP and UDP transport protocols are

discussed in depth along with network security, DNS, SAN, SLIP, firewalls, and naming. Prerequisite: CS 666. Cross-listed with CPE 679 and NIS 679.

### **CS 669 Network Management**

Hierarchical network management for LAN and distributed discrete and integrated services networks; network management concepts; administrative and operational management; performance management; fault management; maintenance management; and security management and architectural management of different ownerships. Concept of managed objects, manager-agent relationship, and applications of network management protocols. Standard management protocols: SNMP and CMIP. Prerequisite: CS 666.

### **CS 670 Information Theory and Coding**

An introduction to information theory methods used in the analysis and design of communication systems. Typical topics include: entropy, relative entropy, and mutual information; the asymptotic equipartition property; entropy rates of a stochastic process; data compression; Kolmogorov complexity; channel capacity; differential entropy; the Gaussian channel; maximum entropy and mutual information; rate distortion theory; network information theory; and algebraic codes. Prerequisite: CS 505.

### **CS 679 Foundations of Cryptography**

This course provides a broad introduction to cornerstones of security (authenticity, confidentiality, message integrity, and non-repudiation) and the mechanisms to achieve them as well as the underlying mathematical basics. Topics include: block and stream ciphers, public-key systems, key management, certificates, public-key infrastructure (PKI), digital signature, non-repudiation, and message authentication. Various security standards and protocols such as DES, AES, PGP, and Kerberos, are studied. Prerequisites: CS 503 and either CS 434 or CS 600.

### **CS 687 Engineering of Large Software Systems**

Students will learn how to deal with issues impacting industrial software developments. A broad range of topics will be covered, emphasizing large project issues. Large software projects are those employing 50 or more software developers for three years or more. Throughout the course, emphasis will be placed on quantitative evaluation of alternatives. Specific examples and case histories from real projects in the telephone industry are provided. Students will learn how to create architectures for large systems based on the '4+1' model; how to use modern software connector technology; module decomposition; scaling of agile methods to large projects, the use of work flows to drive software process and database designs, test plans, and implementation; and configuration control and software manufacturing. The special issues of database conversion data consistency, database maintenance, and performance tuning will be addressed for large data bases. The physical environment of the computer systems, including multisite deployment, software releases, and special management report generation, are examined. Prerequisite: CS 540.

### **CS 689 Software Reliability Engineering**

Students will learn how to analyze, predict, design, and engineer the required and expected reliability of software systems. Case studies

will be used throughout, including studies of systems that worked well and of systems that failed in some crucial aspect. Examples of the types of systems which will be studied are the London Ambulance Dispatch System, the Lucent Telephone Switching Systems, and the Mars and Voyager missions. Prerequisite: CS 533.

### **CS 693 Cryptographic Protocols**

This course covers the design and analysis of security protocols, and studies different attacks and defenses against them. Topics include: signature and authentication protocols, privacy, digital rights management, security protocols for wired, wireless and distributed networks, electronic voting, payment and micropayment protocols, anonymity, broadcast encryption and traitor tracing, quantum cryptography, and visual cryptography. The course includes a project. Prerequisite: CS 668.

### **CS 694 E-Business Security and Information Assurance**

This course addresses the security of e-business and cyber environments from an end-to-end perspective, including data center security and access security. The information security phases of inspection, protection, detection, reaction, and reflection are emphasized. Topics also include: server and application security, virtual local area networks (VLANs), secure access and financial transaction techniques, and backup and disaster recovery techniques. The course also reviews financial Electronic Data Interchange (EDI) and smart card security in banking applications, and describes how the business and financial risks associated with security are estimated and managed. The course includes a project and related lab experiments. Prerequisite: CS 666 or TM 610 or equivalent. Cross-listed with TM 694.

### **CS 700 Formal Semantics of Programming Languages**

Methods for giving meaning to programming language constructs; Operational, Denotational, and Axiomatic semantics. Introduction to algebraic tools; recursive definitions and fixed-point semantics; proving program correctness; and program equivalence. Prerequisite: CS 630.

### **CS 765 Selected Topics in Computer Science\***

A participating seminar on topics of current interest and importance in computer science.

### **CS 800 Special Problems in Computer Science (M.S.)\***

An investigation of a current research topic at the pre-master's level, under the direction of a faculty member. A written report is required, which should have the substance of a publishable article. Students with no practical experience who do not write a master's thesis are invited to take advantage of this experience. One to six credits for the degree of Master of Science (Computer Science).

### **CS 801 Special Problems in Computer Science (Ph.D.)\***

An investigation of a current research topic beyond that of CS 800 level, under the direction of a faculty member. A written report is required, which should have importance in Computer Science and should have the substance of a publishable article. This course is open to students who intend to be doctoral candidates and wish to explore an area that is different from the doctoral research topic. One to six credits for the degree of Doctor of Philosophy.

**CS 802 Software Engineering Examination\***

This will test the software engineering knowledge of students who have completed Stevens Institute of Technology-approved training programs in software engineering. Upon successful completion (graded pass/fail), students will be awarded six credits towards the Master of Quantitative Software Engineering on their study plan and three on the approval form for the certificate of Quantitative Software Engineering. To obtain a pass in this course, the student is required to demonstrate proficiency equivalent to a grade of "B" (i.e. 3.0 out of 4.0) or higher. These credits are not transferable to other institutions. Prerequisite: Completion of an approved Software Engineering Training Program.

**CS 900 Thesis in Computer Science (M.S.)\***

A thesis of significance to be filed in libraries, demonstrating competence in a research area of Computer Science. Five to ten credits with departmental approval for the degree of Master of Science (Computer Science).

**CS 960 Research in Computer Science (Ph.D.)\***

Original research of a significant character carried out under the guidance of a member of the departmental faculty, which may serve as the basis for the dissertation, is required for the degree of Doctor of Philosophy. Credits to be arranged.

\*By request.

[back to top](#)

Copyright © Stevens Institute of Technology • All Rights Reserved  
One Castle Point on Hudson ■ Hoboken, NJ 07030 ■ 800-458-5323