MISSION AND OBJECTIVES

Computer Science encompasses five subject areas: systems, algorithms and data structures, programming languages, databases, and artificial intelligence.

Software development is the practice of crafting and maintaining computer codes, the activity of which may be applied to and/or borrowed from one or more Computer Science subject areas.

The United States Bureau of Labor Statistics has forecast that prospects for software development are two of the top 30 fastest growing occupations in the nation for the period 2008 through 2018 (Lacey and Wright, 2010). Applications development and systems development were ranked 15 and 22, respectively, with expectations for very high median wages.

We can reasonably expect training at the Bachelors and Masters levels in software development to remain in demand at least through this period and, most likely, beyond.

The challenge is to keep curricula up-to-date and relevant for a rapidly evolving job market for students. Furthermore, there is a need to anticipate future needs consistent with emerging trends in the field. The ubiquity of smart phone and tablet devices, wireless communication, cloud computing, security protocols, and advances in programming languages, to name a few recent technological innovations, have propelled the field even farther and in directions that ten years ago were unimaginable.

The Masters of Science Computer Science/Software Development program was founded in the 1990s. The program has served its primary mission, that is, of exposing students to graduate-level studies in software development. However, the program needs an upgrade.

The Computing Technology Department recently instituted major changes in the undergraduate curriculum. Those changes have provided a basis going forward to review and update the graduate program.

Thus, we have identified the following three goals for revising the MSCS/SD curriculum:

1. Improve recruiting.
2. Inject technologically relevant coursework.
3. Appeal to career-oriented student populations locally, nationally, and internationally.

APPLICATION REQUIREMENTS

A baccalaureate degree from an accredited college or university with a GPA of 2.75 or higher is required for admission to the graduate program in computer science. Additionally, applicants should submit the following:

- A completed graduate application and application fee;
• Official copies of all undergraduate and graduate transcripts;
• An updated résumé specifying programming languages known.

Admissions requirements for international students include:
• Submission of iBT TOEFL/IELTS score;
• Bank statement and affidavit of support;
• Recommendation letters.

Formal admission to the master’s degree program will be granted to students who have satisfied these requirements. Some students may, however, be permitted to enroll in graduate courses in a non-matriculated status upon satisfactory evidence of specific prerequisites. Questions concerning mathematical/computer science competency and non-matriculated status should be addressed to the Program Director.

MATRICULATION STATUS
Applicants who satisfy all admissions requirements are admitted as matriculated students. Applicants who are required to complete undergraduate prerequisite courses are admitted as either matriculated or non-matriculated students at the discretion of the Program Director. Occasionally, conditional admission is granted; students must meet the specified conditions to continue in the program.

DEGREE REQUIREMENTS
To qualify for the Master of Science in Computer Science, students must matriculate and complete 31 credits as described below. Degree requirements must be satisfied within seven years of acceptance into the program, with a cumulative index of no less than 3.0. Requests for an extension of the seven-year limitation must be made in writing to the Program Director. Each student is expected to complete the requirements as outlined in the catalog in effect at the time of admission to Marist College. Students may choose to follow a subsequently revised catalog.

All courses leading to the master’s degree in Software Development are offered in the late afternoon or in the evening. Part-time students are limited to registering for one course during their first semester unless prior approval is granted by the Program Director. Full-time study is defined by a semester load of at least nine credits.

ADVISEMENT
The Director of the Software Development Program serves as the advisor for all students in the program. The Program Director provides advice on course sequencing, approves all registration requests, and performs graduation audits. Students should discuss any questions or concerns they may have about their studies with the Director.

PREREQUISITES
All applicants are expected to be proficient in computer programming, computer architecture, and mathematics. The level of competence can ordinarily be demonstrated by appropriate courses in the areas noted below.

Computer Science
  Object Oriented Programming In JAVA
Mathematics
Discrete Mathematics
Probability/Statistics
Differential and Integral Calculus

Graduate Courses in Software Development

MASTER OF SCIENCE IN COMPUTER SCIENCE/ SOFTWARE DEVELOPMENT

Course Requirements
Candidates for the Master of Science in Computer Science/Software Development must complete the following:

Core Courses (16 credits)
- MSCS 510 Software Design and Development
- MSCS 542 Database Management Systems
- MSCS 560 Networking
- MSCS 630 Security Algorithms & Protocols

Tracks (8 credits)
- Cloud Computing
  - MSCS 679 Parallel Processing
  - MSCS 621 Cloud Computing I
- Mobile Computing
  - MSCS 565 Game Development I
  - MSCS 722 Enterprise Mobile Development

Capping Project (4 credits)
- MSCS 710 Project

Electives (4 credits)
Choose 1 from:
- MSCS 665 Game Development II
- MSCS 555 Computer Graphics
- MSCS 550 Artificial Intelligence
- MSCS 721 Software Verification & Maintenance
- Independent Study

Recommended Sequence:
- Semester One – Fall
  - MSCS 560 Networking
  - MSCS 542 Database Management Systems
  - Track Course #1
Semester Two – Spring
MSCS 630  Security Algorithms & Protocols
MSCS 510  Software Design & Development
Track Course #2

Semester Three – Fall
MSCS 710  Project
MSCS/SD  Grad Elective

Internship Courses
MSCS 680, 681, 682 Graduate Internship in Software Development
Description of Courses

MATH 250
Discrete Mathematics
4 Credits
This course introduces the algebraic concepts, methods, and techniques that form the theoretical basis of computer science, including the relevant areas of logic; set theory and the theory of relations; functions; and permutations.
Prerequisite: Three years of high school mathematics

MSCS 501
Object-Oriented Programming
4 Credits
This course is designed to quickly bring students up to speed with current programming practice including problem-solving methods, algorithm development, and the object-oriented paradigm. Students will practice program design, coding, debugging, testing, and documentation using accepted style conventions. This course establishes a foundation for further studies in computer science. Programming projects are assigned.
Prerequisite: None

MSCS 502
Advanced Data Structures
4 Credits
Linear and non-linear data structures will be discussed and implemented including dynamic lists, linked lists, stacks, queues, trees, heaps, and hashing. Complexity will be considered and efficient structures will be covered including balanced binary search trees and priority queues. Advanced Java topics will be covered including abstract classes, interfaces, and iterators. Students will be expected to construct and/or use generic data structure classes.
Prerequisites: MSCS 501 Object Oriented Programming, MATH 250 Discrete Math

MSCS 503
Computer Organization and Architecture
4 Credits
The purpose of this course is to acquire an understanding and appreciation of a computer system’s functional components and their characteristics. Students will learn instruction set architecture, the internal implementation of a computer at the register and functional level, and understand how main activities are performed at machine level as well as gain an appreciation for hardware design at micro level. This course will not meet any graduation requirements and will generally be taken as a foundation course.
Prerequisite: CMPT 220 or MSCS 501

MSCS 510
Software Design and Development
4 Credits
This course introduces a formal approach to the design and development of software systems. The various phases of the software development process are covered and students are introduced to an object-oriented design methodology using Unified Modeling Language. The course is project-driven and student teams design and implement a complex software system that utilizes a well-designed user interface. Java is the language of development and Java Swing will be covered. (Offered: spring semester)
Prerequisite: CMSC 335 Advanced Data Structures

MSCS 542
Database Management Systems
4 Credits
A study of the issues related to the design and administration of modern database systems, with special emphasis on relational database systems. This course will study data modeling, query languages, schema refinement and tuning of existing databases, physical implementation of databases, and systems issues in the management of data.
**Prerequisite:** MSCS 501 Object-Oriented Programming

**MSCS 555**  
**Computer Graphics I**  
*4 Credits*  
This course introduces students to all aspects of computer graphics: hardware, software, and applications. In the course, students will learn the basic concepts underlying computer graphics and gain experience with at least one graphical application programming interface.  
**Prerequisites:** MSCS 501 Object-Oriented Programming; MATH 221 Differential and Integral Calculus

**MSCS 560**  
**Networking**  
*4 Credits*  
This course will focus on OSI and internet protocols that provide a reliable communication channel over the physical network linking a heterogeneous collection of computing resources. Topics covered include: Switching Techniques, Wireless Protocols, Media Access Control, TCP/IP and multicasting.  
**Prerequisites:** MSCS 501 Object Oriented Programming; MSCS 502 Advanced Data Structures; MSCS 503 Computer Architecture and Organization

**MSCS 565**  
**Game Design and Programming 1**  
*4 Credits*  
The objectives of this course are to introduce a disciplined, practical approach to computer game design and programming. The game design aspect focuses on interactive and interface design, world design, play mechanics and rules, and integration of visual and audio components. The game programming aspect focuses on game-specific 2D graphics, animation, physics simulations, algorithms, data structures, and libraries. The course is project-oriented and students working in small teams design and implement their own video games from a list of options.  
**Prerequisite:** MSCS 501 Object Oriented Programming

**MSCS 621**  
**Cloud Computing (formerly Distributed Systems)**  
*4 Credits*  
This course studies cloud computing based on open standards including Internet addresses, sockets, streams, universal character codes, threads, and Internet protocols like REST, XML, JSON client-server database management systems, and content management systems. The course pursues a practical approach to security issues such as the application and use of digital certificates, encryption, and transactional security. Small student teams analyze, design, and build a cloud-based system using software-development best practices.  
**Prerequisites:** MSCS 501 Object-Oriented Programming; MSCS 560 Networking

**MSCS 630**  
**Security Algorithms and Protocols**  
*4 Credits*  
Internet Security is the study of mechanisms through which remote parties can authenticate each other's identity and then communicate securely with each other. Topics covered will include basic cryptographic mechanisms, Public Key Infrastructure (PKI) for distributed security, grammatical mechanisms used to specify security protocols, current government initiatives impacting Internet security. Also covered will be security artifacts that occur in large networks to support and enhance the PKI—these are things like virtual private networks, the secure sockets layer mechanism embedded in all browsers, and directory services such as LDAP which are used as distributed repositories for hold certificates. Included, as well, will be the study of methods that have been used by hackers to break into computer systems.  
**Prerequisites:** MSCS 501 Object Oriented Programming, MSCS 503 Computer Architecture and Organization, Math 250 Discrete Math
MSCS 665
Game Design and Programming 2
4 Credits
This course continues the study of video game design and programming, building on the techniques and methods developed in Game Design & Programming I. In particular, students learn to apply game engines and application program interfaces, strategic and behavioral artificial intelligence, and game optimization techniques for both space and time considerations. Design and programming issues for multiplayer on-line games are also covered, including event-based and turn-based games. The course is project-oriented, and students design and implement their own video games from a list of options. **Prerequisite:** MSCS 565 Game Design and Programming I

MSCS 679
Parallel Computing
4 Credits
Parallel computing has historically played a vital role in addressing the performance demands of high-end engineering and scientific applications. However, it has now moved to center stage in light of current hardware trends and device power efficiency limits. All computer systems – embedded, game consoles, laptop, desktop, high-end supercomputers, and large-scale data center clusters – are being built using chips with an increasing number of processor cores, with little or no increase in clock speed per core. This course will introduce the concept of multicore and multiprocessor parallel programming. Topics such as Amdhal’s law, speedup, efficiency, hyper-threading, task-level vs. data-level parallelism, shared memory vs. shared-nothing algorithms, concurrent vs. parallel collections, database sharding, and debugging and testing will be discussed. Small student teams analyze, design, and build a parallel computing application using software-development best practices. **Prerequisites:** MSCS 501 Object-Oriented Programming; MSCS 503 Computer Architecture and Organization

MSCS 721
Software Verification & Maintenance
4 Credits
Whether a simple standalone tool or complex networked systems, an essential part of the development cycle involves testing whether the software satisfies established requirements and produces the expected results. In this course, students study the methodology behind the testing and debugging of software systems, including: extracting test cases from requirements, assembling a test suite, validation, problem diagnosis, and developing fixes. Issues related to ongoing maintenance, porting software to new platforms, and handling feature requests are also considered. **Prerequisite:** MSCS 510 Software Design & Development

MSCS 722
Enterprise Mobile Dev
4 Credits
Modern software solutions increasingly require deploying applications and system components to mobile devices. Programming for the mobile environment presents a unique combination of challenges as we attempt to satisfy competing constraints of embedded platforms, mobile networking, and security. Students taking this course learn about the challenges and best practices involved in developing robust applications in a mobile environment. Students reinforce and integrate these concepts by designing and coding software to run on devices such as smartphones or tablets. **Prerequisites:** MSCS 510 Software Design & Development; MSCS 542 Data Management; MSCS 560 Networking
Computer Science/Software Development and Information Systems Faculty


RON COLEMAN Associate Professor of Computer Science, 2002. Degrees: B.S., The City College of New York; M.S., Ph.D., Polytechnic University. Specialties: Data Mining; Machine Learning; Distributed Systems; Software Design and Development; Game Design and Programming

D. CENK ERDIL Assistant Professor of Computer Science, 2015. Degrees: B.S., Marmara University, Istanbul, Turkey; M.E., Pennsylvania State University; Ph.D., State University of New York at Binghamton. Specialties: Grid Resource Matching, Self-Organizing Clouds


ALAN R. LABOUSEUR Assistant Professor of Computer Science 2003. Degrees: B.S., Marist College; M.S., Pace University; Ph.D., State University of New York at Albany. Specialties: Database Systems; Web Development; Programming for Business

EITEL J.M. LAURÍA Professor of Information Systems, 2002. Degrees: B.S., University of Buenos Aires, Argentina; M.B.A., Universidad del Salvador, Argentina / Universidad de Deusto, Spain; Ph.D., SUNY Albany. Specialties: Data Management; Information Decision Systems; Business Intelligence; OLAP; Data Mining; Statistical Machine Learning; Bayesian Belief Networks and their application in Information Technology Implementation

ANNE B. MATHEUS Associate Professor of Information Systems, 2001. Degrees: B.A., Marist College; M.A., Marist College; M.S.C.S., Marist College; Ph.D., SUNY Albany. Specialties: Information Decision Systems; Organizational Studies; Information Quality

ROGER NORTON Dean, School of Computer Science and Mathematics, 1980. Novell Certified NetWare Administrator & NetWare Engineer. Degrees: B.S., University of Massachusetts; M.A., Brandeis University; Ph.D., Syracuse University. Specialties: Semantics of Programming Languages; Object-Oriented Programming; Distributed Computing

DONALD R. SCHWARTZ Associate Professor of Computer Science, 2013. Degrees: B.S., University of Louisiana at Lafayette, M.S., University of Louisiana at Lafayette, Ph.D., University of Louisiana at Lafayette. Specialties: Software Engineering, Database, Service Learning

ONKAR P. SHARMA Professor of Computer Science, 1986. Degrees: B.S.E., Bihar Institute of Technology, Bihar University, India; M.S.C.S., University of California at Berkeley; Ph.D.C.S., New York University. Specialties: Computer Architecture; Systems Software