Software Development

school of computer science and mathematics

DIRECTOR, SOFTWARE DEVELOPMENT PROGRAM, COMPUTER SCIENCE

Onkar P. Sharma, Ph.D.
(845) 575-3000, ext. 3610 or 2523
onkar.sharma@marist.edu

MISSION AND OBJECTIVES

The 31-credit Master of Science in Computer Science/Software Development (SD) program is designed to provide advanced knowledge and experience in the various disciplines of computer science to individuals who hold a bachelor's degree in computer science, mathematics, physics, engineering, or some other closely allied field.

A natural extension of the undergraduate program in Computer Science, Marist's SD program prepares its students for a career in industry, government, or education. Individuals already employed within the industry acquire the advanced professional expertise necessary in today's rapidly changing technological environment. This latter group consists of applications and systems programmers, systems developers, design engineers, database designers, technical managers, network specialists, field engineers, test specialists, and others who wish to broaden their understanding of the computer science field, particularly in the area of software development.

Primary areas of study include object-oriented methodologies; software design, development, and implementation; algorithm analysis; theory and structure of programming languages; computer architecture; systems development; database design and management; networking; graphics and animation; artificial intelligence and robotics; distributed systems, and formal languages and computability. The program focuses on both theoretical and practical aspects of computer science. Team building and collaborative skills are emphasized in courses entailing projects. Independent problem solving and analytical thinking skills, which are so vital in the discipline of computer science, are integrated throughout the curriculum.

FACILITIES AND EQUIPMENT

An IBM System/390 (G5) Computer located in Donnelly Hall supports the Marist College time-sharing system. This system is used for administrative applications, instruction, and research. Students, faculty members, and staff members communicate with the mainframe computer through LANs installed at various locations on campus. Software available includes the programming languages PASCAL, ASSEMBLER, PROLOG, C++, REXX, COBOL, JAVA, and FORTRAN; statistical packages SAS, MINITAB, and SPSSX; graphics package GDDM; data retrieval packages DB2 and QMF; modeling and simulation packages GPSS and RESq; the word processing package SCRIPT, and CMS Pipeline.

The student laboratories house over 200 PCs for student use and several classrooms are equipped with a PC and monitor to facilitate instruction. The department administers microcomputer laboratories including object-oriented technologies, networking, and databases. Software available in the departmental laboratory includes the programming languages JAVA, C, PASCAL, C++, LISP, EIFFEL, ORACLE, and other programming languages.

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 appropriate fee.
- Official copies of all undergraduate (including two-year colleges) and graduate transcripts.
- An updated résumé specifying programming languages known.

Admissions requirements for international students may be found at *www.marist.edu/graduate*.

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 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. Non-matriculated students must matriculate before graduation. It is the responsibility of the student to determine when matriculated status should be requested.

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 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. Starting with the second semester, it is recommended that part-time students take two courses per semester to ensure early completion of the degree requirements.

ADVISEMENT

The Director of the Software Development Program serves as the advisor for all students in the SD program. The program director provides advice on course sequencing, and approves all registration requests. 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

Programming and Data Structures in JAVA
Programming and Data Structures in C++
Programming in Assembly Language
Logic Design and Computer Architecture
Advanced Data Structures in C++

Mathematics

Differential and Integral Calculus
Discrete Mathematics
Probability/Statistics

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 (15 credits)		**Semester Offered**
MSCS 510	Software Design and Development	Spring
MSCS 530	Algorithms	Fall
MSCS 560	Computer Networks I	Fall
MSCS 610	Structure of Programming Languages	Spring
MSCS 590	Distributed Systems	Fall
Laboratory Course (1 Credit)		
MSCS 561	Computer Networks Laboratory	Fall, Spring
Elective Courses (12 credits/select 4 courses)		
MSCS 515	Operating Systems	Fall
MSCS 518	Compiler Design I	Spring
MSCS 521	Computer Architecture	Spring
MSCS 531	Automata, Computability & Formal Languages	Spring
MSCS 542	Database Management	Spring
MSCS 550	Artificial Intelligence	Fall
MSCS 555	Computer Graphics I	Fall
MSCS 560	Computer Networks I	Fall
Additional Electives (offered when there is sufficient demand)		
MSCS 516	Concurrent Programming	
MSCS 520	Performance Evaluation	
MSCS 545	Logic Programming	
MSCS 596-600	Special Topics in Computer Science	Spring
MSCS 618	Compiler Design II	
MSCS 630	Theory of Computation	
MSCS 640	Distributed Database Systems	
MSCS 652	Modeling & Simulation	
MSCS 655	Computer Graphics II	
MSCS 660	Computer Networks II	
MSCS 670	Applied Artificial Intelligence	
MSCS 700	Thesis	
Project Course (3 credits)		
MSCS 710	Project	Fall
Internship Courses		
MSCS 690, 691, 692	Graduate Internship in Software Development	All Semesters

Although not required, students may elect to pursue a concentration in Systems Software by taking elective courses in operating systems, compiler design, and computer architecture; or a concentration in Applications by taking elective courses in database management, artificial intelligence, and computer graphics.

Elective courses may be selected from the software development courses listed in the graduate catalog, including the Thesis course. Occasional special topics courses, when offered, will satisfy the Elective requirement. ***Internship Courses do not satisfy the Elective requirement.***

Catalog Description of Courses

MSCS 510
Software Design and Development
3 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 515
Operating Systems
3 Credits
Operating systems provide service to users to simplify their programming and data processing tasks, and they also manage systems resources to assure their efficient utilization. This course mainly presents operating systems as resource managers. In order to gain hands-on practical experience, students will write a multiprogramming operating system as an integral part of the course. *(Offered: Fall semester)* **Prerequisites:** CMSC 335 Advanced Data Structures; CMSC 415 Computer Architecture

MSCS 516
Concurrent Programming
3 Credits
This course introduces the technique of concurrent programming. Concurrent programming deals with programming in which several activities are processed in parallel. It is essential in the design of operating systems. Students will write concurrent programs. **Prerequisites:** CMSC 335 Advanced Data Structures; CMSC 415 Computer Architecture

MSCS 518
Compiler Design I
3 Credits
Both the design and implementation of compilers will be studied. The lexical, syntactic, and semantic analyses of formal languages will be developed. Theoretical tools such as finite-state and push-down automata, and regular and context-free grammars will be presented as needed. Additionally, symbol-table construction and code-generation techniques will be required to develop a compiler for a selected subset of an instructor-specified small programming language. *(Offered: Spring semester)* **Prerequisites:** CMSC 335 Advanced Data Structures; CMSC 415 Computer Architecture

MSCS 520
Performance Evaluation
3 Credits
A survey of techniques of modeling concurrent processes and the resources they share. Includes levels and types of system simulation, performance prediction, bench marking and synthetic loading, and hardware and software monitors. **Prerequisite:** CMSC 335 Advanced Data Structures

MSCS 521
Computer Architecture
3 Credits
The objective of this course is to introduce concepts related to the organization and structure of the major hardware components. The functions and implementations of, and communication between, the major components of a computer system are described.

Developments to make special types of processing more efficient or reliable such as pipe lining and array processing are covered; special emphasis is placed on RISC and multiprocessing architectures. *(Offered: Spring semester)* **Prerequisites:** CMSC 330 Logic Design; MATH 221 Differential and Integral Calculus

MSCS 530
Algorithms
3 Credits
This course will develop students' abilities as writers and critics of programs. The student will be introduced to a variety of program-design techniques including recursion, heuristics, divide-and-conquer, and dynamic programming. Methods of performance analysis and the theory of NP-completeness will be covered. *(Offered: Fall semester)* **Prerequisite:** CMSC 335 Advanced Data Structures.

MSCS 531
Automata, Computability & Formal Languages
3 Credits
Formal language theory will be presented, including the Chomsky hierarchy of formal languages with their corresponding grammars and automata. The study of formal language leads naturally to considerations related to the notion of algorithm and to the nature of limits of algorithmic computation. Various forms of models of computation will be explored. *(Offered: Spring semester)* **Prerequisite:** CMSC 335 Advanced Data Structures

MSCS 542
Database Management
3 Credits
A study of the concepts and issues related to managing data in an information system. The evolution of computerized information systems from early file systems to current decision-support systems is examined. Major database design philosophies along with their corresponding data models are explored. Specific examples of current database management systems, as well as issues such as recovery, integrity, concurrency, and security are discussed. *(Offered: Spring semester)* **Prerequisites:** MATH 250 Discrete Mathematics; CMSC 335 Advanced Data Structures

MSCS 545
Logic Programming
3 Credits
This course will present an overview of logic programming, especially as it relates to the programming language PROLOG. The main emphasis of the course will be on the theoretical aspects of logic programming with applications of PROLOG playing a secondary role. **Prerequisite:** CMSC 335 Advanced Data Structures

MSCS 550
Artificial Intelligence
3 Credits
This course introduces students to basic concepts and techniques of artificial intelligence, or intelligent systems, and gives insight into active research areas and applications. Emphasis is placed on representation as a central and necessary concept for work in intelligent systems. *(Offered: Fall semester)* **Prerequisite:** CMSC 121 Computer Science II

MSCS 555
Computer Graphics I
3 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. *(Offered: Fall semester)* **Prerequisites:** CMSC 310 Object-Oriented Programming Using C++; MATH 221 Differential and Integral Calculus

MSCS 560
Computer Networks I
3 Credits
This course will focus on the seven layers of the OSI Network Model. Students are introduced to hardware components of a network such as client and server machines, transmission media, bridges, routers and gateways, as well as network software components, and in particular the TCP/IP Protocol Suite. Topics covered include: Switching Techniques, Data Link Protocols,

Media Access Control, TCP/IP Protocol Suite. *(Offered: Fall semester)*
Prerequisites: CMSC 415 Computer Architecture; MATH 221 Calculus I; MATH 130 or 330 Probability/Statistics
Corequisite: MSCS 561 Computer Networking

MSCS 561
Computer Networks Laboratory

3 Credits

This is a hands-on course in the building and troubleshooting of both peer-to-peer and client/server networks at all levels of the OSI and Internet Network models. The course consists of two halves: the first focuses on networking basics and peer-to-peer networks, the second on client/server networks. Microsoft Windows NT 4.0 combined with the Back Office Small Business Server is the platform for the client/server portion. The objective will be to develop a typical Intranet as would be used by a small business or department that provides web serving, e-mail, proxy/firewall protection, and DNS. This course provides the student with the hands-on knowledge and experience to allow the building, development, and troubleshooting of an Intranet Network for the typical small business or department. *(Offered: Fall and Spring semester)*
Corequisite: MSCS 560 Computer Networks I

MSCS 590
Distributed Systems

3 Credits

This course studies Open Standards distributed systems based on the TCP/IP protocol. The course exposes students to Internet addresses, sockets, streams, universal character codes, threads, and Internet protocols like SMTP, HTTP, MIME, etc. The course also investigates several object and service distribution methods like Remote Method Invocation (RMI), Common Object Request Broker (CORBA), JavaSpaces, and Jini. The course pursues a practical approach to these ideas through simple Java programs as well as a larger project written in Java in which small student teams analyze, design, and build a distributed system using software development practices. *(Offered: Fall semester)* **Prerequisites:** MSCS 510 Software Design and Development; MSCS 560 Computer Networks I

MSCS
Special Topics

3 Credits

MSCS 596 Systems Software
MSCS 597 Computer Architecture
MSCS 598 Database
MSCS 599 Artificial Intelligence
MSCS 600 Computer Science

Special-topics courses serve as a vehicle by which a division may offer a topical or thematic study not included in the regular course offerings. The specific content is indicated when the course is listed in the schedule of classes. *(Offered upon demand or instructor interest in Spring semester.)*
Prerequisite: Permission of Instructor

MSCS 610
Structure of Programming Languages

3 Credits

Data and control abstractions are considered. Advanced control constructs including backtracking and non-determinism are covered. Emphasis is on machine-independent implementation of programming language constructs. *(Offered: Spring semester)*
Prerequisites: CMSC 335 Advanced Data Structures; CMSC 415 Computer Architecture

MSCS 611
Formal Methods in Programming Languages

3 Credits

This course deals with the formal representation of programming language constructs, which are then utilized to describe the major methods for proving program correctness and for giving formal program specifications.
Prerequisite: CMSC 335 Advanced Data Structures

MSCS 618
Compiler Design II

3 Credits

The topics covered in Compiler Design I will be reviewed, followed by the consideration of type and scope analyses. A more detailed study of code generation will be conducted with regard to code optimization. Error recovery strategies and run-time environments will be discussed. As time permits, recent advances in compiler design will be reviewed. Each student will engage in a project agreed upon jointly by the instructor and student. **Prerequisite:** MSCS 518 Compiler Design I

MSCS 630
Theory of Computation

3 Credits

A survey of formal models of computation, including Turing Machines, partial recursive functions, recursive and recursively innumerable sets, the recursion theorem, abstract complexity theory, program schemes, and concrete complexity. **Prerequisite:** MSCS 531 Automata, Computability and Formal Languages

MSCS 640
Distributed Database Systems

3 Credits

Consideration of the problems and opportunities inherent in distributed databases on a network computer system. Topics covered include file allocation, directory systems, dead-lock detection and prevention, synchronization, query optimization, and fault tolerance. **Prerequisites:** MSCS 542 Database Management; CMSC 335 Advanced Data Structures

MSCS 652
Modeling and Simulation

3 Credits

A study of the construction of models which simulate real systems. The methodology of solutions will include probability and distribution theory, statistical estimation and inference, the use of random varieties, and validation procedures. A simulation language will be used for the solution of typical problems. **Prerequisite:** CMSC 335 Advanced Data Structures

MSCS 655
Computer Graphics II

3 Credits

This course introduces advanced modeling and viewing techniques in computer graphics such as surface patches, solid modeling, hidden surface removal, ray tracing, radiosity, and animation. **Prerequisite:** MSCS 555 Computer Graphics I

MSCS 660
Computer Networks II

3 Credits

The investigation in more depth of some of the topics introduced in Computer Networks I. Among some of the topics chosen by the instructor and the class to be the main concentration for that particular semester: queuing theory, performance analysis of basic access protocols, a detailed analysis of routing algorithms, flow control and buffer allocation algorithms, Internet working, protocol verification, and encryption techniques. **Prerequisite:** MSCS 560 Computer Networks I

MSCS 670
Applied Artificial Intelligence

3 Credits

This course builds upon the first level AI course by concentrating on a limited number of topics from AI, investigating these topics to considerable depth, and emphasizing the design and implementation of software pertaining to these topics. Selection of specific topics to be pursued will be determined by the instructor in consultation with the students in the class. **Prerequisite:** MSCS 550 Artificial Intelligence

MSCS 690, 691, 692
Graduate Internship in Software Development

One, two, and three credits respectively

The graduate internship will provide advanced professional experience in the field of computer science. This course enables students to integrate the elements of their formal preparation and to apply theoretical concepts to real-world software development. Graduate Internships cannot be used to meet any elective requirements. Arrangements should be made with the graduate director and internship coordinator. *(Offered Fall, Spring, and Summer semesters)* **Prerequisites:** Completion of 12 graduate credits and 3.0 GPA

MSCS 710
Project
3 Credits

This is a project-based course. Students will work in teams to analyze, design, and implement a large system chosen from a list of selected projects. Students will utilize the skills gained in previous courses, especially Software Design and Development, in working as a team going through the various phases of the software development process. Student teams will have milestone presentations, including a final presentation, throughout the course. This course meets on a weekly basis. *(Offered Fall semester)*
Prerequisites: Completion of at least 18 credits and MSCS 510 Software Design and Development

MSCS 720
Thesis
3 Credits

Thesis can only be taken by a student who has completed the project course.

During the semester prior to enrollment in Thesis, the student must submit a thesis proposal for approval to register for Thesis to his/her project advisor or graduate director six weeks prior to the end of the semester in which the student is enrolled for the project course. If approved, the Thesis advisor, the program director, and the student, acting together, solicit two additional faculty members to act as members of the student's Thesis Committee. The three faculty members constituting the committee may include not more than one adjunct faculty member.

During the semester in which Thesis is taken for credit, the following must take place:

As the thesis course progresses, the student works on his/her thesis under the guidance of his/her thesis advisor. The student meets with his/her advisor periodically, as determined by the latter, to seek guidance and submit progress reports. The student submits the completed thesis to the three faculty members of his/her committee by the tenth week of the semester. The thesis must be found acceptable by the thesis advisor and at least one additional committee member. In the event that revision of the thesis is recommended, it may be necessary to issue an incomplete grade (a grade of X). This grade may be changed at any point in the future after the requirement listed above has been satisfied. The student will then receive a regular grade for Thesis.
Prerequisite: MSCS 710 Project

Computer Science/Software Development and Information Systems Faculty

HAROLD ANDERSON Associate Professor of Computer Science, 2000. *Degrees*: B.S., University of Rhode Island; M.S., Trinity College; Ph.D., Syracuse University

JOSE ARREOLA Assistant Professor of Computer Science. *Degrees*: B.S., Universidad Nacional Autónoma de México; M.S., Pennsylvania State University; Ph.D., University of Pittsburgh. *Specialties*: Parallel Algorithms; Optimization

ROBERT M. CANNISTRA Professional Lecturer of Computer Science, Information Systems and Information Technology. *Degrees*: B.S., State University of New York at Brockport; M.S., Marist College (in progress). *Specialties*: Network Infrastructure and Design, Policy-Based Routing, Network Security

RON COLEMAN Assistant 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; Software Testing

CRAIG FISHER Associate Professor of Information Systems, 1989. *Degrees*: B.S., State University of New York at Oswego; M.A., Ball State University; Ph.D., State University of New York at Albany. *Specialties*: System & Information Concepts; Problem Solving & Programming; Systems Analysis & Design; Database Management

JAN HARRINGTON Assistant Professor of Information Systems, 1989. *Degrees*: B.S., University of Washington; M.L., University of Washington; Ph.D., Drexel University. *Specialties*: Data Management; System Architecture; Object-Oriented Technologies

HELEN M. HAYES Assistant Professor of Mathematics and Computer Science, 1983. *Degrees*: B.A., College of St. Elizabeth; M.S., Fordham University; M.S.C.S., Syracuse University. *Specialties*: Formal Languages; Computability; Algorithms; Neural Networks

JOAN E. HOOPES Assistant Professor of Information Systems, 1990. *Degrees*: B.S., State University of New York at Binghamton; M.B.A., State University of New York at Binghamton; Ph.D., State University of New York at Binghamton. *Specialties*: Programming Concepts; Systems Analysis & Design

ALAN G. LABOUSEUR Professional Lecturer of Information Technology, 2003. *Degrees*: B.S., Marist College; M.S., Pace University. *Specialties*: Database Systems; Web Development; Programming for Business

EITEL J.M. LAURIA Visiting Assistant 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 the application in Information Technology Implementation

DANIEL MARCELLUS Associate Professor of Computer Science, 2000. *Degrees*: B.A., Cornell University; Ph.D., Harvard University. *Specialties*: E-Commerce; Web Technology

MAURICIO MARENGONI Assistant Professor of Computer Science, 2003. *Degrees*: B.S., College of Industrial Engineering, Brazil; M.S., Brazilian Institute for Space Research, Brazil; Ph.D., University of Massachusetts, Amherst. *Specialties*: Computer Vision; Reasoning under Uncertainty; Robotics

ANNE B. MATHEUS Professional Lecturer of Computer Science; Director of Computer Literacy, 2001. *Degrees*: B.A., Marist College; M.A., Marist College; M.S.C.S., Marist College; A.B.D., SUNY Albany. *Specialties*: Information Decision Systems; Organizational Studies; Information Quality

JEROME A. MCBRIDE Associate Professor of Information Systems, 1983. *Degrees*: B.S., Manhattan College; M.S.C.S., Purdue University. *Specialties*: Information Systems in Organizations; Data Base Management; Decision Support Systems; Systems Analysis & Design; Management Science

ROGER NORTON Associate Professor of Computer Science, 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

ONKAR P. SHARMA Professor of Computer Science, 1986; Director, Software Development Program. *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

JAMES TEN EYCK Assistant Professor of Computer Science, 1983. *Degrees*: B.S., Lafayette College; M.S., Syracuse University; Ph.D., Syracuse University. *Specialties*: Computer Networks; Simulation

JAMES WEIR Visiting Lecturer in Information Technology, 2003. *Degrees*: B.S., M.S., Lehigh University. *Specialties*: Multimedia; Web Applications