

# **Graduate Certificate** **in Software Development and Software Design**

## **University of Massachusetts Dartmouth** **College of Engineering** **Computer and Information Science Department**

Upon completion of this certificate program, participants will have gained:

- Knowledge of tools, techniques, and theory associated with software development.
- Understanding and breadth of knowledge of software engineering
- Understanding of fundamental issues associated with software engineering of parallel and distributed software systems
- A route into the MS programs in either Computer Engineering or in Computer Science at UMD.

### **The Certificate Courses::**

CIS 525 Software Development of Parallel and Distributed Systems  
CIS 531 Software System Specification  
CIS 532 Software System Design  
CIS 552 Database Systems  
CIS 580 Paradigmatic Software Development

The **Software Development and Software Engineering** certificate program is offered by Computer and Information Science. The five courses that students take for this certificate are compatible with MS degree requirements for Computer Science, enabling certificate recipients to transition to degree program without loss of credit.

### **Admission to the Certificate Program**

At UMass Dartmouth, applicants apply for acceptance to a graduate certificate program through the Office of Graduate Studies. Applicants for acceptance to this certificate are required to hold an appropriate undergraduate degree such as a BS in Computer Engineering, Computer Science, or Electrical Engineering, or BS in other academic disciplines. Applicants to this certificate also follow the general procedures for graduate degree acceptance into the MS in Computer Science, but in a curtailed form. Thus, they fill out the same application form as do degree applicants and supply an essay and official transcripts of all post-secondary work, as instructed therein. However, they are not required to submit GRE scores or recommendation letters.

Prospective students can take up to two courses for the certificate before completing their official acceptance to the certificate program, with permission of the instructor.

A joint faculty committee will review applications for acceptance to this certificate program. One element in the review will be to assess whether a student has met the stated prerequisites and in general has the appropriate combination of background and experience to succeed in the program. It is possible for acceptance to be offered with a contingency that the student must also take one or more undergraduate prerequisite courses.

**Admissions Contact** (Feel free to call, e-mail, or write to receive an application and catalogue)

Office of Graduate Studies  
University of Massachusetts Dartmouth  
285 Old Westport Road  
North Dartmouth, Massachusetts 02747-2300

graduate@umassd.edu  
phone: 508-999-8604  
fax: 508-999-8183

**[www.umassd.edu/graduate](http://www.umassd.edu/graduate)**

### **Courses for the Certificate in Software Development and Software Engineering**

Certificate students receive the same benefits as do degree students as regards access to the full range of university services and facilities and may register along with graduate degree students. The record of their studies is recorded on an official transcript, which identifies the specific certificate program. Other graduate policies, such as those for longevity of credit, acceptable grades, and repeatability of courses, apply to certificate students as to degree students. Certificate students pay graduate tuition and fees.

**CIS 525** three credits

#### **Parallel and Distributed Software Development**

3 hours lecture

Prerequisite: CIS 361 or permission of instructor

Design and development of parallel and distributed systems. This course provides state-of-the art presentation of software development for parallel and distributed systems. A systematic model-based approach has been applied across stages of software development. Various versions of Petri nets are used to model, specify, validate, and verify correctness of parallel and distributed systems. Performance is also assessed based on stochastic Petri nets. Rapid prototyping of parallel and distributed systems with automatic code generation is an ultimate goal of his course. Comparison with other approaches is also provided.

**CIS 531** three credits

**Software System Specification**

3 hours lecture

Prerequisite: CIS 480 or equivalent

Formal foundation of the theory and practice of software specification; production of correct, consistent, and reliable software systems by expressing the requirements of the system in formal ways. Formal and informal requirements analysis and specification techniques, the relation of analysis and specification to concerns of validation and verification, maintenance, and reusability.

**CIS 532** three credits

**Software Systems Design**

Prerequisite: CIS 480 or equivalent

Principles and techniques for obtaining an architectural design from a system specification. Where appropriate, automated software design tools are used to demonstrate particular methodology. The relation of various design methods to the production of quality software that meets its specification, and the relation of design method to other life-cycle aspects. Design methods, design tools, the design process, and particular application domains for design techniques.

**CIS 552** three credits

**Database Design**

Prerequisite: CIS 452 or equivalent

The relational, hierarchical, and network approaches to database systems, including relational algebra and calculus, data dependencies, normal forms, data semantics, query optimization, and concurrency control on distributed database systems.

**CIS 580** three credits

**Paradigmatic Software Development**

3 hours lecture

Prerequisites: CIS 480 or equivalent

Software development in the context of various paradigms. The strategies and methods of the procedural, object-oriented, and functional paradigms are studied and practiced. The modeling of software processes will be considered from both the process and product views, as will the appropriateness and measures of effectiveness of these processes in the design of software systems. Students will apply these measures to the course exercises, determining and reviewing the impact of these methods on individual design.