SOFTWARE ENGINEERING (SWENG)

SWENG 311: Object-Oriented Software Design and Construction

3 Credits

Design, documentation, testing, and construction of software using software engineering strategies embodied in object-oriented programming languages. SWENG 311 Object-Oriented Software Design and Construction (3) Object-oriented design and programming embody powerful design strategies that are based on practical and proven software engineering techniques. In this class, students will learn how existing object-oriented languages support these strategies, how to apply these strategies to moderately-sized systems, and how to use a visual object-oriented modeling tool such as the Unified Modeling Language (UML). Students will build on programming skills acquired in prerequisite programming courses by programming in a major object-oriented programming language.

Enforced Prequisite at Enrollment: CMPSC 122

SWENG 395: Internship

1-18 Credits/Maximum of 18

Supervised off-campus, nongroup instruction including field experiences, practica, or internships. Written and oral critique of activity required.

Enforced Prerequisite at Enrollment: Prior approval of proposed assignment by instructor

SWENG 396: Independent Studies

1-18 Credits/Maximum of 18

Creative projects, including research and design, that are supervised on an individual basis and that fall outside the scope of formal courses.

SWENG 397: Special Topics

1-9 Credits/Maximum of 9

Formal courses given infrequently to explore, in depth, a comparatively narrow subject that may be topical or of special interest.

SWENG 399: Foreign Studies

1-12 Credits/Maximum of 12

Courses offered in foreign countries by individual or group instruction.

International Cultures (IL)

SWENG 400: Introduction to Software Engineering Studio

3 Credits

Provides an introduction to the principles of software engineering and includes complementary instruction in one programming language.

Enforced Prerequisite at Enrollment: Senior standing or above

SWENG 411: Software Engineering

3 Credits

Software engineering principles including life cycle, dependability, process modeling, project management, requires specification, design analysis, implementation, testing, and maintenance. SWENG 411 Software Engineering (3) This is an introductory course in software engineering, addressing the software development process, including aspects such as software requirements documentation, design specification, implementation, system integration, testing, and maintenance by individuals and teams. Topics include software process modeling, requirements elicitation and documentation, software architecture design and analysis, detailed design and programming, graphical user interface (GUI's) design and prototyping, software quality assessment, software testing, software maintenance and evolution management, personal and team-based development. In lab students gain practical experience by completing programming assignments and utilizing computer-aided software engineering (CASE) tools for their personal projects tailored to each stage of the software life cycle. A semester long team-based project is required that reinforces teamwork fundamentals and the concepts covered in lecture. The projects and assignments provide an opportunity for student teamwork, document writing, and oral presentations.

Enforced Prerequisite at Enrollment: CMPSC 122 and Concurrent: SWENG 311

SWENG 421: Software Architecture

3 Credits

The analysis and design of software systems using canonical design patterns. SWENG 421 Software Architecture (3)This course introduces the frequently-used software infrastructures in software development by experienced engineers. The formal UML notations are utilized to design software architecture and help communicate the design visually. Students will learn the real practice of architectural styles, design patterns and design reuse. As to certain complex problems, alternative architectures will be proposed and their design trade offs will be evaluated. For instance, students compare two-tier with three-tier client/server architectures for distributed systems, and employ multi-process and multi-thread concurrent architectures for high performance computation systems. Moreover, students learn to conduct high level quality analysis from the design artifacts. The quality evaluation will focus on a number of attributes, including reusability, extendibility and performance. A great deal of effort is placed on the major categories of design types containing dozens of separate design patterns. Students first review the most fundamental design patterns. Afterwards, they apply creational patterns to effectively create objects, partitioning patterns to categorize objects, structural patterns to allocate objects, behavioral patterns to interface the communication between objects, and concurrent patterns to handle tasks simultaneously. These skills will enable students to extend their own knowledge after graduation by giving them the skills to learn new patters on their own. Finally, students will integrate their programs with native code applications to enlarge the application domains. To achieve best reusability, they also learn modular designs to develop component-based software. These help them meet today's software needs of cross applications and architectures.

Enforced Prerequisite at Enrollment: SWENG 411

SWENG 431: Software Verification, Validation, and Testing

3 Credits

Introduction to methods of software verification, validation, and testing; mathematical foundations of testing, reliability models; statistical testing. SWENG 431 Software Verification, Validation, and Testing (3) Provides a background necessary for verification, validation, and testing of software systems. Verification addresses the question: 'are we building the product right?' In other words, does the product meet the engineering specifications? Validation addresses whether the right product is being built and if it meets the design requirements. The testing aspect of the course addresses many of the methods available to test software systems. The levels of testing explored are 1) unit level (each module is tested independently), 2) integration testing (where the modules are integrated together and tested as a complete system), and 3) acceptance testing (the testing requirements of the users). Following this, specific test methodologies are addressed. By the end of this course the student should also be able to develop an appropriate test plan.

Enforced Prerequisite at Enrollment: SWENG 411 and (STAT 301 or STAT 318)

SWENG 452W: Embedded Real Time Systems

3 Credits

The design and implementation of real time systems. SWENG 452W Embedded Real Time Systems (3) Real time operating systems is the study of hardware/software systems in which timing constraints must be met for correctness. Real time systems are embedded in applications ranging from the antilock brakes in cars to the flight control systems for jetliners. Students are first introduced to the concept of systems with real time constraints by examining case studies. The unified modeling languages (UML) with real time extension is introduced allowing students to capture the constraints present in the systems in a variety of models allowing the problem to be described at several levels of abstraction. Tasks and messages are introduced as programming structures which can satisfy the constraints described by the UML models. With a basic understanding of real time systems and how to implement them, the focus of the course shifts away from these technical concerns towards understanding the documentation of the requirements using the Volere Requirements Specification template. All the writing assignments in the class will revolve around Volere in one way or another. Increasingly complex case studies will give the class the opportunity to explore more sophisticated inter-task communications mechanisms as well as common pitfalls present in RTOS applications. Students will learn how to verify the correctness of their applications in order to guarantee that the real time constraints can be met when the system is deployed. Discussion will turn to application programmer interfaces used by hardware vendors to port hardware into a RTOS. The class will end by designing and building a complex RTOS by a team of students using the techniques learned in the class. The project will outline the needs of the RTOS application in a project proposal using the Volere template. The proposal will be developed iteratively with the faculty member until its scope and definition are clear. This proposal will be developed into a complete requirement specification including a time-line and identification of development benchmarks. This system resulting from the development will be documented in the final report write-up.

Enforced Prerequisite at Enrollment: CMPSC 472 or CMPSC 473 or CMPEN 441
Writing Across the Curriculum

SWENG 455: Engineering Quality and Security in Software

3 Credits

Software systems are becoming an integral part of our daily life. In many domains the failure of a software system in not acceptable. Such a failure will have catastrophic effects. Hence, the quality of such software systems becomes so crucial. This course introduces software quality, the associated quality factors, software errors and the software quality assurance system. As software systems advance in complexity so does the threats that they face. Hence, the security of these systems has become so essential. Traditionally, security has been thought off as an add-on to an existing system. In this course, students will understand that this is not sufficient. Students will learn to consider security throughout the development process by extending the traditional software development life cycle and UML. For the requirements phase, this course will teach students to collect secure requirements, perform security requirements planning, and perform vulnerability mapping. For the design phase, this course will teach students to design a secure software architecture and perform architectural risk analysis. For the implementation phase this course will introduce students to secure coding. For the testing phase, the course will introduce students to secure testing. Finally students will be introduced to secure deployment and secure maintenance. A semester long team-based project will be part of this course to practice the concepts covered in the class.

Prerequisites: SWENG 411

SWENG 465: Web Services

3 Credits

This course introduces the students to a contemporary computing paradigm called "service-oriented computing." SWENG 465 Web Services (3) This course focuses on a new computing paradigm called "service-oriented computing", which has been greatly impacting a wide array of software systems. It covers "service-oriented architecture." Students will not only gain an in-depth understanding of the concepts and technical issues underpinning Web services, but also gain hands-on experience of the development of software systems built upon Web services.

Enforced Prerequisite at Enrollment: SWENG 311 or CMPSC 221

SWENG 480: Software Engineering Design

3 Credits

Concepts of engineering ethics, economy, and project management, senior capstone project selection, and technical communication skills. SWENG 480 Software Engineering Design (3)This course prepares senior software engineering students for industrial engineering design and project management. It covers the engineering design process, project planning and evaluation, engineering ethics, and engineering economy. In addition, students select, specify, and start their capstone design project, which is completed, in the follow-up course, SWENG 481. Students are expected to carry out a group design project that is on par with industrial expectations. Upon completion of this course a student should have a solid understanding of the engineering design process, a clear capstone project description, should have completed some preliminary design work, and be adequately prepared to complete the project in SWENG 481.

Enforced Prerequisite at Enrollment: SWENG 431

SWENG 481: Software Engineering Project

3 Credits

Capstone group design projects in software engineering. SWENG 481 Software Design Projects (3) In this course students complete their group senior design project started in SWENG 480. Design groups meet regularly with a faculty advisor to report progress and resolve technical issues. Oral and written progress reports are expected at selected times. The class culminates with a final technical defense of the project.

Enforced Prerequisite at Enrollment: SWENG 480

SWENG 494: Research Project

1-12 Credits/Maximum of 12

Supervised student activites on research projects identified on an individual or small-group basis.

SWENG 495: Internship

1-18 Credits/Maximum of 18

Supervised off-campus, nongroup instruction including field experiences, practica, or internships. Written and oral critique of activity required.

Enforced Prerequisite at Enrollment: Prior approval of proposed assignment by instructor

SWENG 496: Independent Studies

1-18 Credits/Maximum of 18

Creative projects, including research and design, that are supervised on an individual basis and that fall outside the scope of formal courses.

SWENG 497: Special Topics

1-9 Credits/Maximum of 9

Formal courses given infrequently to explore, in depth, a comparatively narrow subject that may be topical or of special interest.

SWENG 499: Foreign Studies

1-12 Credits/Maximum of 12

Courses offered in foreign countries by individual or group instruction.

International Cultures (IL)