<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
<th>Term Offered</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE-505</td>
<td>Principles of Software Engineering</td>
<td>1.5</td>
<td>All Terms</td>
<td>None</td>
</tr>
<tr>
<td>SE-511</td>
<td>Object-Oriented Analysis</td>
<td>1.5</td>
<td>All Terms</td>
<td>SE-505</td>
</tr>
<tr>
<td>SE-512</td>
<td>Object-Oriented Design</td>
<td>1.5</td>
<td>Spring Term</td>
<td>SE-511</td>
</tr>
<tr>
<td>SE-515</td>
<td>Disciplined Software Development</td>
<td>3</td>
<td>All Terms</td>
<td>SE-505</td>
</tr>
<tr>
<td>SE-517</td>
<td>Engineering Web-Based Systems</td>
<td>3</td>
<td>All Terms</td>
<td>CS-501A</td>
</tr>
<tr>
<td>SE-565</td>
<td>Software System Requirements</td>
<td>3</td>
<td>All Terms</td>
<td>CS-501A, SE-505, and SE-511</td>
</tr>
<tr>
<td>SE-571</td>
<td>Software Design and Systems Architecture</td>
<td>1.5</td>
<td>All Terms</td>
<td>None</td>
</tr>
<tr>
<td>SE-572</td>
<td>Enterprise and Global Architecture</td>
<td>1.5</td>
<td>All Terms</td>
<td>SE-571</td>
</tr>
<tr>
<td>SE-565</td>
<td>Software System Requirements</td>
<td>3</td>
<td>All Terms</td>
<td>None</td>
</tr>
<tr>
<td>SE-565</td>
<td>Software System Requirements</td>
<td>3</td>
<td>All Terms</td>
<td>None</td>
</tr>
<tr>
<td>SE-575</td>
<td>Software Verification, Validation and Maintenance</td>
<td>3</td>
<td>Fall Term</td>
<td>CS-501A, SE-505</td>
</tr>
<tr>
<td>SE-580</td>
<td>The Process of Engineering Software</td>
<td>3</td>
<td>Spring Term</td>
<td>None</td>
</tr>
<tr>
<td>SE-571</td>
<td>Software Design and Systems Architecture</td>
<td>1.5</td>
<td>All Terms</td>
<td>None</td>
</tr>
</tbody>
</table>

Basic introduction to software engineering. Justification that software engineering is an engineering discipline. The two main components of the software engineering discipline- the software product and software process-will be discussed in detail. For design and verification, there will be a focus on traditional Structured-Analysis method. Limited to Software Engineering students only.

SE-511 Object-Oriented Analysis
Prerequisite: SE-505.
Term Offered: All Terms
Course Type(s): CISEL
Introduces the development of the requirements and analysis model for a software application. Uses object-oriented methodologies. This is not a programming course. Limited to Software Engineering students only.

SE-512 Object-Oriented Design
Prerequisite: SE-511.
Term Offered: Spring Term
Course Type(s): CISEL
Introduces the design of a software application. Uses object-oriented methodologies. This is not a programming course. Limited to Software Engineering majors only.

SE-515 Disciplined Software Development
Prerequisite: SE-505.
Term Offered: All Terms
Course Type(s): None
Introduces the need for an engineering approach to software, through a motivational discussion of the so-called ”software crisis” and a presentation of software development processes at the various degrees of granularity; this ranges from organizational processes to team and individual engineers’ processes. Provides practical experiences with the Personal Software Process. A software process is defined as a set of guidelines, activities, and methods that engineers use to develop and maintain software. Without a software process, most projects are over budget and behind schedule, since successful efforts cannot be guaranteed. The success of a project depends greatly upon the experience and discipline of the engineers. Limited to Software Engineering students.

SE-517 Engineering Web-Based Systems
Prerequisite: CS-501A.
Term Offered: All Terms
Course Type(s): None
A practical introduction to the principles, methods and tools required to create high-quality software applications for the distributed, client-server context of the Web. The course emphasizes on architectural designs, and language and data access methods that are common in web-based systems.

SE-565 Software System Requirements
Prerequisites: CS-501A, SE-505, and SE-511.
Term Offered: All Terms
Course Type(s): CISEL
Students will learn advanced methods in software system requirements. Ideally, these methods should be applied at the system, enterprise or global levels of software development. Methods in requirements elicitation, modeling of enterprises, prioritization and negotiation of requirements will be emphasized. Methods for including Commercial Off-the-Shelf (COTS) system applications and product families will be introduced. By the end of the course, students will master the standard documentation of system requirements. Many opportunities for hands-on experience with requirements tools will be provided throughout the course.

SE-571 Software Design and Systems Architecture
Prerequisite: SE-565.
Term Offered: Spring Term
Course Type(s): CISEL
Introduces software application design and system architecture in terms of the design process, design principles, design notations, design tools, design heuristics, and design patterns. Covers application and system design in detail concentrating on developing designs that are complete, correct, robust, implementable, and deployable. Limited to Software Engineering students only.

SE-572 Enterprise and Global Architecture
Prerequisite: SE-571.
Term Offered: All Terms
Course Type(s): CISEL
Introduces architectural design at the Enterprise and Global Architectural levels. Provides students with an understanding of how new systems are brought into an organization to interoperate with an existing system, how systems are maintained throughout their lifecycle, and how systems are retired at the end of their lifecycle. Also provides students with an understanding of how the computing capabilities of a company fits into the larger scope of the Internet. Limited to Software Engineering students only.

SE-575 Software Verification, Validation and Maintenance
Prerequisites: CS-501A and SE-505.
Term Offered: Fall Term
Course Type(s): None
Covers software verification, validation and maintenance. The first half of the course includes inspections of requirements, design and code as well as testing. The second half addresses the handling of change requests, software evolution, code comprehension, and change management. It will include hands-on experience with a change management system and an automated testing tool.

SE-580 The Process of Engineering Software
Prerequisites: CS-503 and SE-505.
Term Offered: Spring Term
Course Type(s): None
Students will learn how to define, apply and improve a software process. The basic components of a software process will be introduced, as well as the most influential process models. Students will experiment with both plan-based and agile methods of software development. The need for continuous process quality assessment and improvement will be described. Models such as CMM, ISO9001 will be introduced as reference models for organizations process models.
Software Engineering (SE)

SE-588  Software Practice and Experience  Credits: 1
Term Offered: All Terms
Course Type(s): None
Provides opportunity for Software Engineering graduate students to obtain related experience in employment at a local company or institution, with Monmouth University sponsorship. Available to Software Engineering graduate students who have completed at least eighteen credit hours of graduate courses (500-level), with a minimum GPA of 3.00. Does not satisfy elective requirements. Students may take this course a maximum of two times. This is a pass/fail course.

SE-599  Independent Study in Software Engineering  Credits: 1-3
Prerequisite: Completion of at least 12 credits in the program.
Term Offered: All Terms
Course Type(s): None
The development and execution of a significant research project designed by the student in consultation with a software engineering professor. This is an experience meant to focus learning on an area of special interest to the student.

SE-601  Outsourcing: Specifications and Strategies  Credits: 3
Prerequisites: SE-565 and SE-571.
Term Offered: Fall Term
Course Type(s): None
Covers issues associated with outsourcing software development in a fashion that assures delivery of acceptable products. The emphasis is on basic factors that determine the effectiveness of outsourcing, strategies for minimizing risk, project tracking, contract-specified break-points, and requirements.

SE-602  Technology Assessment  Credits: 3
Prerequisites: SE-565 and SE-571.
Term Offered: All Terms
Course Type(s): None
Covers the practicalities of assessing a technology for use in delivery of products. It examines how new tools, processes, and training fit into an overall technology-adoption decision.

SE-603  MOST Implementation  Credits: 3
Prerequisites: SE-565 and SE-571.
Term Offered: Spring Term
Course Type(s): None
A practical application of MOST concepts by following the process of adopting a new technology and putting the appropriate elements into an organization. Focuses on technology assessment, cost, scheduling, training, and project management.

SE-605  Software Implementation and Reuse  Credits: 3
Prerequisites: SE-565 and SE-571.
Term Offered: All Terms
Course Type(s): None
Explores contemporary topics in systematic software implementation and reuse. Includes the impact of Object-Based and Object-Oriented Design and Programming with Ada83, Ada95, and C++ along with Domain Engineering on the software development process. Concentrates on the practical aspects of applying architecture-centric, domain-specific, library-based reuse methodologies integrated with the software development process to create software systems in an efficient, cost-effective manner. Illustrates how object-oriented and domain-engineering techniques coupled with domain-specific libraries can be used to effectively develop significant software systems in a short period of time, frequently realizing reuse on the order of 70% or more. Libraries of object-based reusable software components will be used to design and implement solutions to problems.

SE-610  Software Systems Security  Credits: 3
Prerequisite: SE-505.
Term Offered: Fall Term
Course Type(s): None
Threats, vulnerabilities, and attacks to network-based systems, Heuristic procedures for breaking systems. UNIX and Internet case studies. Security modeling techniques including Bell-Lapadula, Biba, and Clark-Wilson. Composition of non-deducibility and noninterference security. Safeguard techniques including cryptographic protocols, online auditing and intrusion detection, access control, Internet firewalls, authentication, security kernel design, and trusted software development. Case studies in database security and secure network design.

SE-611  Secure Web Services Design  Credits: 3
Prerequisite: SE-565.
Term Offered: All Terms
Course Type(s): None
Web applications present a complex set of security issues for architects, designers, and developers. The most secure and hack-resilient Web applications are those that have been built from the ground up with security in mind. This course focuses on principles of secure Web applications design. Topics include threats and counter measures, security in Web service frameworks, session control, access control, and data protection. Also listed as CS-611.

SE-615  Usability Engineering/Human-Computer Interaction  Credits: 3
Prerequisite: SE-565.
Term Offered: Spring Term
Course Type(s): None
Explores the requirements analysis, design and evaluation of the Computer User Interface in the context of Software Engineering Processes. Specific methods and design problems will be illustrated with real-world examples in information technology, the Internet, communications, mobility, multimedia, and speech technologies. Prepares students to perform usability tasks directly or to successfully manage and collaborate with usability experts.

SE-616  Extensible Markup Language (XML)  Credits: 3
Prerequisite: CS-501B.
Term Offered: Summer Term
Course Type(s): None
Presents XML fundamental concepts, XML as a document format, XML as a data format, and special topics in using XML.

SE-620  Networked Software Systems I  Credits: 3
Prerequisites: SE-565 and SE-571.
Term Offered: Fall Term
Course Type(s): None
Network Computing Models, Concepts and Requirement specification; Techniques for Interprocess Communication; Message Passing; Remote Procedures Calls; Directory Service; Synchronization; Task Partition and Allocation; Threads; Security and Authentication; Time Management.

SE-621  Networked Software Systems II  Credits: 3
Prerequisite: SE-620.
Term Offered: Spring Term
Course Type(s): None
SE-623  Management Information Systems  Credits: 3
Term Offered: All Terms
Course Type(s): None
A survey of the concepts of management information systems and the information needs of management. A user-oriented introduction of the fundamentals of information systems and their integration into business organizations. Also listed as MIS-623. Not open to students who have successfully completed BM-520.

SE-625  Information Systems Architecture  Credits: 3
Prerequisites: SE-565 or MIS-565.
Term Offered: Fall Term
Course Type(s): None

SE-626  Information Systems Engineering  Credits: 3
Prerequisite: SE-625.
Term Offered: Spring Term
Course Type(s): None
Theoretical and practical issues related to the implementation of information systems to support medium to large organizations. Databases within the Target Architecture and their role in building enterprise-wide information systems. Logical vs. Physical database design. Transaction Processing (TPS) and Decision Support Systems (DSS): similarities and differences. Knowledge-based systems. Implementation issues; reliability, integrity, security, performance, scalability and maintainability. Review of existing commercial tools and environments for building, using and maintaining Information Systems for the enterprise.

SE-630  Real Time Software Analysis and Specification  Credits: 3
Prerequisites: CS-503 and SE-505.
Term Offered: Fall Term
Course Type(s): None

SE-631  Real-Time Software Design and Implementation  Credits: 3
Prerequisite: SE-630.
Term Offered: Spring Term
Course Type(s): None
Criteria and trade-off for partitioning the functions of a real-time system among the hardware, firmware, and software. The interfaces of a real-time system to the external environments such as sensors and actuators. Designing real-time systems: structured and object-oriented methods. Case studies and comparison between the two. Design verification and validation. Implementation of Real-Time Systems. Development environments and tools: Simulators, Emulators and Debuggers. Testing Real-Time systems. Test coverage and regression testing. Special approaches for Real-Time software testing. Engineering and organizational issues in real-time software development. Performance testing and its importance in real-time systems.

SE-650  Software Project Management  Credits: 3
Prerequisite: SE-505.
Term Offered: Fall Term
Course Type(s): None

SE-651  Software Organization Management  Credits: 3
Prerequisite: SE-505.
Term Offered: Spring Term
Course Type(s): None

SE-652  Software Quality Management  Credits: 3
Prerequisites: SE-565 and SE-571.
Term Offered: Summer Term
Course Type(s): None
Designed to help students develop a broad understanding of the requirements and consideration in planning, managing, and implementing a software project with special emphasis on teams, quality, process, and measurement. This experiential course involves two iterations of a team project which help students learn how to apply specific software engineering practices and understand their value (or lack of value) to specific projects. The main themes include in-depth coverage of process models, such as CMMI and ISO 9000, as well as specific practices such as Risk Management, Configuration Management, Team Development, Inspections, and practical applications of Measurement.
SE-660 Computer System Architecture Credits: 3
Prerequisite: CS-503.
Term Offered: All Terms
Course Type(s): None
Computer system interconnection structures, central processing unit, control unit, microprogrammed control, memory organization, cache and virtual memory, computer arithmetic, RISC processor, introduction to parallel processing, and case studies.

SE-691 Software Engineering Thesis Research Credits: 3
Prerequisites: SE-565, SE-571, and SE-580.
Term Offered: All Terms
Course Type(s): None
Independent research in software engineering, spanning a period of two consecutive semesters in an area not substantially covered in a regular course offering, under the supervision of a software engineering faculty member. A formal thesis is required. An oral defense of the thesis before a committee of two faculty members, plus the advising professor is required. Depending on the topic, a reviewer from outside the software engineering department may be asked to serve as a committee member. Limited to Software Engineering majors.

SE-692 Software Engineering Thesis Research Credits: 3
Prerequisites: SE-691 and prior permission of the advising professor.
Term Offered: All Terms
Course Type(s): None
Independent research in software engineering, spanning a period of two consecutive semesters in an area not substantially covered in a regular course offering, under the supervision of a software engineering faculty member. A formal thesis is required. An oral defense of the thesis before a committee of two faculty members, plus the advising professor is required. Depending on the topic, a reviewer from outside the software engineering department may be asked to serve as a committee member. Limited to Software Engineering majors.

SE-695A Software Engineering Practicum Credits: 3
Prerequisites: SE-565, SE-571, and SE-580.
Term Offered: All Terms
Course Type(s): None
A two-semester course sequence with major emphasis on hands-on, team-oriented large software development projects. Students will be asked to form groups of three to five persons, who will work as a software team whose purpose will be to develop a software product. The particular product will be chosen through negotiation with the instructor. Teams will operate in accordance with a well-defined set of policies and procedures (documented in an SE Handbook) that governs the entire development process. This document process addresses work products, roles, activities, entry and exit criteria, reviews and audits, documented procedures, and training. The practicum will also include the use of a set of approved tools. Additional software project management topics such as planning, estimation, and tracking will be covered. Project management techniques and their application to the management of software projects. How to define a software development plan, its associated tasks, milestones, and deliverables. Software project scheduling and the establishment of relationships among the different tasks. Task dependencies and conflict resolution. Resource management and allocation. Software project cost estimation. Algorithmic models for estimating costs: the COCOMO model and its derivatives. Risk assessment and its impact in the planning and scheduling of software projects. Software project measurement and tracking. Comparative review of software tools for software project management. Software configuration management and its importance in the management of large software projects. The students will be asked to serve in team roles, hold oral reviews, and prepare documentation appropriate to their project. Students in the class will participate in the reviews of other teams in the class. At the end of the second semester, each team will give a formal presentation on the project. Limited to Software Engineering majors.
Software Engineering Practicum (SE-695B) Credits: 3
Prerequisite: SE-695A.
Term Offered: All Terms
Course Type(s): None
A two-semester course sequence with major emphasis on hands-on, team-oriented large software development projects. Students will be asked to form groups of three to five persons, who will work as a software team whose purpose will be to develop a software product. The particular product will be chosen through negotiation with the instructor. Teams will operate in accordance with a well-defined set of policies and procedures (documented in an SE handbook) that governs the entire development process. This documented process addresses work products, roles, activities, entry and exit criteria, reviews and audits, documented procedures, and training. The practicum will also include the use of a set of approved tools. Additional software project management topics such as planning, estimation, and tracking will be covered. Project management techniques and their application to the management of software projects. How to define a software development plan, its associated tasks, milestones, and deliverables. Software project scheduling and the establishment of relationships among the different tasks. Task dependencies and conflict resolution. Resource management and allocation. Software project cost estimation. Algorithmic models for estimating costs: the COCOMO model and its derivatives. Risk assessment and its impact in the planning and scheduling of software projects. Software project measurement and tracking. Comparative review of software tools for software project management. Software configuration management and its importance in the management of a large software project. Students will be asked to serve in team roles, hold oral reviews, and prepare documentation appropriate to their project. Students will participate in the reviews of other teams in the class. At the end of the second semester, each team will give a formal presentation on the project. Limited to Software Engineering majors.

Special Topics in Software Engineering (SE-698) Credits: 3
Prerequisite: Approval of the department.
Term Offered: All Terms
Course Type(s): None
The subject matter varies with the interest of the students and of the professor teaching the course. The exact nature of the topic covered in any given semester is indicated in the student’s transcript.

Individual Research Project in Software Engineering (SE-699) Credits: 3
Prerequisite: Approval of the department.
Term Offered: All Terms
Course Type(s): None
The development and execution of a significant research project designed by the student in consultation with a software engineering professor. An experience meant to focus learning on an area of special interest to the student.