

## 교과목 개요

### SEP503 소프트웨어 이론 개요 (Computer Science for Software Engineers)

소프트웨어 공학 석사 과정을 시작하기 위해 공학도에게 필요한 고급 이산 수학, UML 2.0과 디자인 도구, 고급 데이터 구조 및 알고리즘, 프로그램 실행, 객체 지향 방법과 같은 다양한 주제를 공부한다. 덧붙여, 소프트웨어 공학 분야에서 오늘날 많은 관심을 받고 있는 시간제하에 대한 부분(미들웨어, 디자인 패턴, 부 품기술)을 다룬다. 이 과정의 공식 프로그래밍 언어는 Java이다.

### SEP537 소프트웨어 시스템 모델링 (Models of Software Systems)

전산학자들은 오랫동안 소프트웨어 개발을 명세에서부터 프로그램까지 자동화하는 문제를 연구해 왔다. 지금까지 완전히 성공적이라고 볼 수는 없지만, 많은 결과들이 작은 프로그램과 큰 프로그램의 중요한 부분 들에 적용되어 효과적으로 사용될 수 있음이 입증되었다. 이 과목에서는 이러한 결과들을 공부하는데, 그 내용으로 소프트웨어 시스템을 형식 언어로 모델링하는 방법, 사용자가 원하는 시스템의 성질을 검증할 수 있는 모델을 만드는 방법, 그리고 사용자가 요구하는 성질들을 모델들이 갖고 있는 지를 엄밀하게 검증할 수 있는 방법을 공부한다.

### SEP539 소프트웨어 개발 방법론 (Methods: Deciding What to Design)

효과적인 소프트웨어 개발을 위해서는 해결해야 할 문제와 사용할 소프트웨어 시스템 간의 차이를 연결할 성공적인 방법의 이해가 필요하다. 이 수업에서는 학생 스스로 풀어야 할 문제를 이해하는 다양한 방법과 가능한 해결책들을 제약하는 다양한 요소 및 대안 중 결정에 이르는 다양한 방법들을 공부한다.

### SEP591 소프트웨어 개발 관리 (Managing Software Development)

대규모 소프트웨어 개발을 위해서는 개발과정 기간 내내 자원(인간과 컴퓨터)을 관리할 능력이 필요하다. 이 수업은 규모적인 요구에 의해 개설된 수업이고, 과학적으로 훈련된 소프트웨어 공학도들이 프로젝트 팀 을 이끌기 위해 필요한 지식과 기술을 얻는데 도움을 주기 위해 고안되었다. 또한, 총체적으로 프로젝트에 서 소프트웨어 개발의 관계를 이해하고, 시간과 비용을 예상하며, 소프트웨어 과정을 이해하는데 도움을 주는 것도 이 과정이 추구하는 목표이다.

### SEP601, 602, 603 소프트웨어 개발 스튜디오 I, II, III (Software Development Studio I, II, III)

소프트웨어 개발 스튜디오 과목은 학생들이 전공필수 및 선택과목에서 얻은 지식을 실제 소프트웨어 개발 과정에 적용하며 체험하도록 하는 것을 목표로 한다. 팀워크, 조직, 그리고 체계적인 소프트웨어 프로세스 의 사용에 특히 중점을 두어 프로젝트를 수행한다. 이 과목은 MSE 과정 전반에 걸쳐 소프트웨어 개발 스 튜디오 I, II, III으로 나누어 각각 가을, 봄, 여름 학기에 진행된다. 이 과목에서는 학생에 대한 개별적인 지 도뿐만 아니라 프로젝트 팀별로 조언을 주기 위한 전문가가 멘토로 선정된다. 매 주 팀원 및 멘토와의 미팅 을 가지며, 여기서 학생들은 자신들의 선택과 결정이 프로젝트에 미치는 영향에 관하여 서로 질문하고 토 론하게 된다.

### SEP604 CMU Software Studio

이 과목은 SEP 603 소프트웨어 개발 스튜디오 III과 연계된 것으로서 여름학기 스튜디오 프로젝트 수행 과 정에서 학생들은 CMU 멘토들의 원격 및 방문 지도를 받게 된다.

### SEP608 Analysis of Software Artifact

분석과정에서는 소프트웨어 산출물들에 대한 체계적인 검사를 통해 그것들의 속성을 파악하는 작업을 수 행한다. 이 과목에서는 코드 뿐 만 아니라 소프트웨어 디자인, 아키텍처, 테스트 관련 소프트웨어 산출물에 대한 분석 방법에 대해 다룬다. 특히 기능적인 속성들 뿐 만 아니라 성능, 보안 등 비기능적 속성들에도 초점을 둔다. 핵심적인 분석 개념들에 대한 심도 있는 논의를 위해 이 과목에서는 정적 프로그램 분석 기 법에 중점을 둔다. 그러나 테스팅, 모델 검증, 정리 입증(Theorem Proving), 동적 분석, 타입 시스템 등의 폭 넓은 주제도 다루게 된다. 또한 이 과목에서는 이론적인 논의와 실험적 학습을 병행함으로써 학생들이 배 운 것을 실제 산출물 분석에 적용해 볼 수 있도록 한다.

### SEP609 Architecture for Software System

복잡한 소프트웨어 시스템의 성공적인 설계를 위해서는 시스템의 명세, 분석, 생성이 아키텍처 수준의 추 상화를 통해 이루어 져야 한다. 이 과목에서는 복잡한 소프트웨어 시스템의 아키텍처 설계 방법을 소개한 다. 일반적으로 사용되는 소프트웨어 시스템 아키텍처 형태, 이러한 아키텍처를 설계하고 구현하기 위한 기술, 아키텍처의 특성을 파악하고 추론하기 위한 모델과 형식적 표현법, 구체적인 아키텍처 인스턴스

(instance)를 생성하기 위한 도구, 실제 시스템 아키텍처에 대한 사례 연구 등이 이 과목에서 다루어지게 된다. 그리고 이 과목에서는 학생들이 기존 시스템의 아키텍처를 평가하고 새로운 시스템을 원칙에 따라 설계하기 위해 필요한 기술과 기본 지식을 가르친다.

#### **SEP605 Practicum Proposal**

이 과목을 통해 학생들은 프랙티컬 과정에서 수행할 프로젝트를 제안하게 된다. 이 과목의 진행 과정에서 학생들은 프로포절 문서를 작성하여야 한다. 프로포절에서는 수행할 프랙티컬에 대한 요약을 설명하고 목표로 하는 작업에 대한 정의를 산출물, 스케줄, 검토, 최종 보고서 등을 통해 기술해야 한다.

#### **SEP606, 607 Practicum I, II**

이 과목은 학생들이 소프트웨어 엔지니어로서 필요한 능력들을 캡스톤 시연(capstone demonstration) 과정을 통해 보여주기 위한 것이다. 이 과목의 목표는 학생들이 전공필수 및 선택과목에서 배운 것들은 실제 문제 해결에 적용해 보는 것에 있다. 학생들은 실제적인 환경에서의 비중 있는 문제를 소프트웨어 개발 라이프 사이클의 주요 관점에서 이해함으로써 그 해결책을 도출하게 된다. 프랙티컬은 개별적으로 수행하거나 3-4 명의 학생들이 팀을 이루어 수행할 수 있다.

\* 선택 및 공통필수과목, 연구과목 개요는 소프트웨어대학원 프로그램 교과목과 동일