

# 소프트웨어공학

한성대학교 컴퓨터공학부

신 성



# 수업 안내 및 교과목 소개

## 수업 안내

- 담당교수 : 신성 (컴퓨터공학부)
- E-mail : sihns@gachon.ac.kr
- 휴대폰 번호 : 010-8873-8353

- 우리 학생들뿐만 아니라  
교수자 입장에서라도 학생들과의 소통은 정말 중요합니다.

# 수업 안내

- **〈소프트웨어공학〉 3학점, 온라인 강좌 1.5, 대면 강좌 1.5**
  - 이론과 실습을 온라인(동영상) 강의 + 대면 강의 형태로 진행
  - 반드시 온라인 강의를 들은 후(출석체크) 대면 강의에 참석하여야 함
- **온라인 강의는 ‘한성 e-class’에 매주 올려드리는 강의를 주차별로 수강하여 출석**
  - 월요일 수업이므로 (대면 수업이 끝난 후) 월요일 ~ 일요일 자정까지 수강
  - 기간이 지난 이후에도 반복 수강은 가능하나 출석은 인정 불가
  - 온라인(동영상) 강의 시간은 매주 38분 이상 수업

# 수업 안내

## • 한성대학교 원격수업 운영 규정

### - 제2장 5조

☞ 수업 내용에 따라  
주 38분을 넘는 강의도 다수 있습니다.

(온라인 강의는 오프라인 강의 2배 이상의  
내용이 진행되므로 크게 무리가 되지 않는  
범위 안에서 진행)

☞ 매주 e-class에 올려드리는 강의를  
모두 수강해야 출석 인정

구분		수업 학 점	학점 구 성	시간 구성			운영 주차
				수업 동영상	전체 학습시 간 (온라인 기준)	오프라 인	
온라 인 수업	수업동 영상 녹 화 수업	3학점	-	주 75분 이상	주 150분 이 상	-	15주 차
		2학점	-	주 50분 이상	주 100분 이 상	-	15주 차
블렌디드 러닝			2+1	주 50분 이상	주 100분 이 상	50분	15주 차
		3학점	1.5+1.5	주 38분 이상	주 75분 이상	75분	15주 차
			1+2	주 25분 이상	주 50분 이상	100분	15주 차
		2학점	1+1	주 25분 이상	주 50분 이상	50분	15주 차
FL+PBL		3학점	-	주 10분 이상(필수) 주 20분 이상(권장) ※ 플립러닝으로 운영하는 주 차의 오프라인 수업 시간은 온라인 콘텐츠 시간에 준하여 축소 운영할 수 있음	150분	5주차 이상	
		2학점			100분		

※ 전체 학습시간이란 수업 동영상과 학습활동 시간을 모두 포함한 시간을 의미함

※ 실시간 원격화상 수업은 기본 수업운영 방식에는 포함되지 않고 다른 수업운영 방식에 보조적으로 사용될 수 있다.

# 수업 안내

- 여러분들은 다음 온라인 강의의 장점을 살려서 학습에 활용

- (시간 맞춰서 학교에 나오지 않아도)

시간과 장소에 구애받지 않고 편한 시간에 자유롭게 수업을 들을 수 있다.

- 언제든지 반복 청취가 가능하다.

- 자기 학습진도에 맞는 자기주도적 학습이 가능하다.

# 수업 안내

- 온라인 강의의 단점은? 담당교수와의 소통

- 학생들 뿐만 아니라

교수자 입장에서도 학생들과의 소통은 정말 중요합니다.

- 앞으로 수업받으면서 궁금한 점이나 질문 있으면 언제든지 관찮으니

편한 방법(메일, 문자, 카톡, 전화)으로 연락주세요~

- 또는 대면 수업 시간(75분)을 활용

# 교과목 소개

## • 교과목 개요

- 이 과목에서는 waterfall 모델, 애자일 프로세스, DevOps, Lean StartUp 등을 포함한 소프트웨어 개발 라이프 사이클 모델을 소개하여 소프트웨어 개발 과정을 이해한다. 또한 요구사항 단계를 요구사항 개발 및 관리 단계로 구분하여 설명한다. 소프트웨어 프로젝트에서 모델링은 아주 중요한 수단이며 이를 위하여 UML의 주요 다이어그램에 대해 설명한다. 소프트웨어 설계 단계에서는 응집도 및 결합도를 소개하고 SOLID 5개의 설계 원칙에 대해 예를 들어설명한다. 또한 다양한 소프트웨어 아키텍처 패턴에 대해 소개한다. 소프트웨어의 품질을 높이는 수단으로 널리 사용되고 있는 소프트웨어 테스트 기술들을 소개하고 오픈소스 소프트웨어를 활용한 실습을 통해 이해를 높이고 현장에 바로 적용할 수 있도록 한다.

## • 수업 목표

- 소프트웨어 개발 라이프 사이클 모델 - UML을 이용한 모델링 - 요구사항 개발 및 관리 - 응집도와 결합도
- SOLID 설계원칙 - 소프트웨어 아키텍처 - 테스트 방법



# 수업 안내

- 강의 진행 방법

- 이론 강의와 실습

- 선수과목

- Java

- 주교재

- 강의 노트

부교재/참고문헌

1. 정인상, 생능출판사, 오픈 소스 소프트웨어로 실습하는 소프트웨어공학, 2017

- 기타 안내 사항

- 매주 과제/수시시험이 있습니다.
- 모든 분반이 같은 시간에 시험을 볼 수 있으며, 이 경우 별도의 시간으로 정해서 시험을 봅니다.
- 본 강의계획서(이하 주차별 내용 포함)는 상황에 맞추어 변동가능성이 있음을 알려드립니다.

# 수업 안내

## • 수업 일정

※ 강의계획서의 주차별 내용은 상황에 맞추어 변동 가능성이 있음을 알려드립니다.

주차	수업내용 및 학습활동
1	온라인 ● 강의 소개 ● 오픈 소스 라이선스 대면 수업 ● 수업 오리엔테이션
2	온라인 ● 소프트웨어 위기와 소프트웨어 공학의 필요성 ● 예제로 본 소프트웨어 개발 프로세스 ● 소프트웨어 유지보수 유형 ● SDLC(소프트웨어 개발 라이프 사이클 모델) 소개 ● 폭포수 모델 대면 수업 ● 퀴즈 풀이
3	온라인 ● RAD(Rapid Software Development) SDLC ● Lean StartUp ● Agile ● DevOps 대면 수업 ● 퀴즈 풀이
4	온라인 ● 소프트웨어 요구사항 프로세스 ● 요구사항 단계 소개 ● 요구사항 분류 ● 요구사항 개발 프로세스 ● 요구사항 관리 프로세스 대면 수업 ● 퀴즈 풀이
5	온라인 ● UML을 이용한 소프트웨어 모델링 ● UML 소개 ● UseCase 다이어그램 소개 ● UseCase 다이어그램과 사용자 요구사항 모델링 ● UMLet 모델링 도구 소개 대면 수업 ● 퀴즈 풀이
6	온라인 ● UML을 이용한 소프트웨어 모델링 ● 클래스 다이어그램 소개 ● 클래스 다이어그램을 통한 소프트웨어 분석 ● 시퀀스 및 패키지 다이어그램 소개 대면 수업 ● 퀴즈 풀이
7	온라인 ● 소프트웨어 설계 원칙 ● 응집도와 결합도 ● 소프트웨어 복잡도 : 사이클로매틱 복잡도 대면 수업 ● 퀴즈 풀이
8	중간고사

# 수업 안내

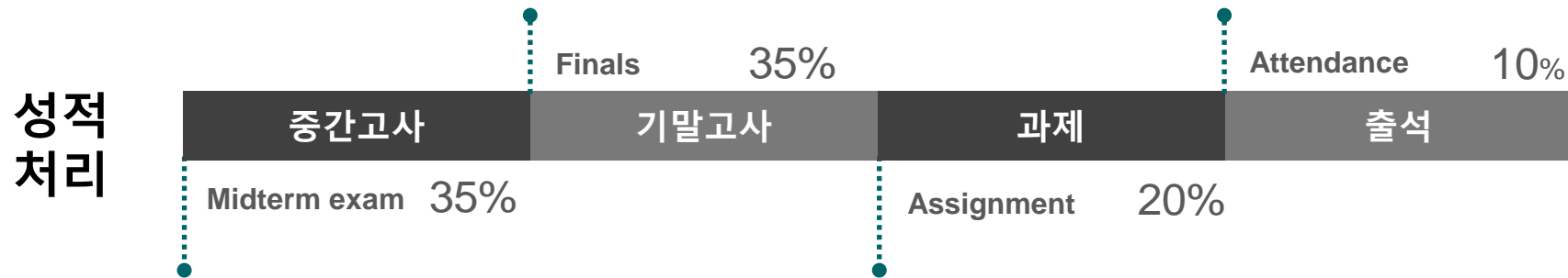
## • 수업 일정

※ 강의계획서의 주차별 내용은 상황에 맞추어 변동 가능성이 있음을 알려드립니다.

주차	수업내용 및 학습활동
9	온라인 ● 소프트웨어 설계 원칙 : SOLID ● SRP ● OCP ● LSP ● ISP ● DIP 대면 수업 ● 중간고사 풀이
10	온라인 ● 소프트웨어 아키텍처(1) ● MVC(Model-View-Controller) ● Layered Architecture ● EDA(Event Driven Architecture) 대면 수업 ● 퀴즈 풀이
11	온라인 ● 소프트웨어 아키텍처(2) ● MicroService Architecture ● Hexagonal architecture 대면 수업 ● 퀴즈 풀이
12	온라인 ● 소프트웨어 테스트 중요성 ● 테스트 기본 원리 ● 테스트 한계 ● 테스트 오라클 대면수업 ● 퀴즈 풀이
13	온라인 ● 테스트 케이스 설계 방법 소개 ● ISO29119 소개 ● 블랙박스 테스트 소개 ● 동등 분할 방식 ● 경계값 분석 방식 ● 페어와이즈 테스트를 포함한 조합 테스트 소개 대면수업 ● 퀴즈 풀이
14	온라인 ● 화이트 박스 테스트 ● 커버리지 개념 소개 ● 제어흐름그래프 소개 ● 블록 커버리지 소개 ● 분기 커버리지 소개 대면수업 ● 퀴즈 풀이
15	기말고사

# 수업 안내

## • 성적 평가



※ 출석 (온라인, 오프라인 수업 모두 매시간 본인이 직접 '출결 여부'를 잘 확인해 주셔야 합니다.)

- 결석 1회당 1점 감점(1주당 최대 2점 감점, 온라인 1점, 대면 1점)

- 지각 3회는 결석 1회로 간주(온라인 수업은 지각 개념 없음)

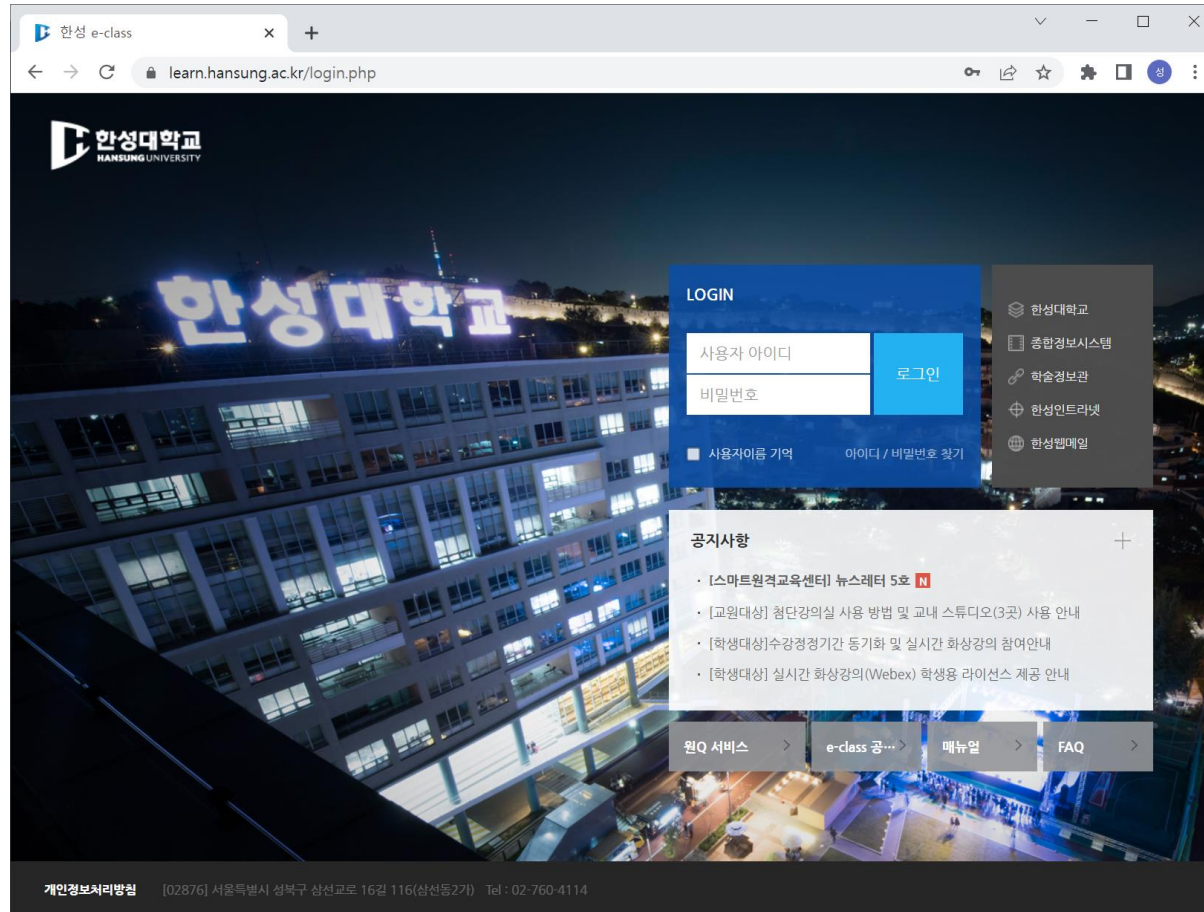
※ 중간고사, 기말고사는 필기시험

# 수업 안내

## • 한성 e-class

- 모든 수업자료, 과제 제출, 공지, 온라인(동영상) 강의 수강

## ※ 코스모스 앱



# 수업 안내

어차피 해야 할 공부라면

공부에 흥미를 느끼고 재미를 느낄 수 있으면 좋겠습니다.

단순히 암기하지 말고, 근본적인 원리와 개념부터 이해를 하려고 노력하시고,  
또 계속 주어진 문제를 해결하면서 반복된 성취감을 느끼는 것이 중요

우리 학생들 모두 우수하고 훌륭한 학생들이고,

기본 소양을 갖춘 학생들입니다.

같이 한번 열심히 해봅시다. 우리 학생들 화이팅입니다. ^^



 **T h a n k      y o u**

## **TECHNOLOGY**

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Velit ex  
plico ipsum, labore sed tempore ratione asperiores des  
quidat bore sed tempore rati jgert one bore sed tem!