

# SW설계 및 테스트

한성대학교 컴퓨터공학부

신 성



# 수업 안내 및 교과목 소개

## 수업 안내

- 담당교수 : 신성 (컴퓨터공학부)
- E-mail : sihns@hansung.ac.kr
- 휴대폰 번호 : 010-8873-8353 (개인적으로 질문은 카톡을 선호하지만 문자, 메일도 가능합니다)
- 연구실 : 우촌관 702호 (내가 연구실에 없을 수 있으니 방문 시는 미리 연락 주세요~)

- 우리 학생들뿐만 아니라

교수자 입장에서든 학생들과의 소통은 정말 중요합니다.

# 수업 안내

- <SW설계 및 테스트> 3학점, 대면 강좌 3
  - 이론과 실습을 대면 강의 형태로 진행
- 면담가능시간 및 장소
  - 상시 연락

분류번호 : 2001020226\_23v6

능력단위 명칭 : 애플리케이션 테스트 관리

능력단위 정의 : 애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력이다.

# 수업 안내

분류번호 : 2001020226\_23v6

능력단위 명칭 : 애플리케이션 테스트 관리

능력단위 정의 : 애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력이다.

NCS(National Competency Standards) 국가직무능력표준

- NCS는 개인의 직업능력을 평가하고 개발하기 위한 기준으로, 직업군별 필요한 지식, 기술, 태도 등을 체계적으로 정리한 것, 직업 교육, 채용, 인사 관리, 직무 평가 등 다양한 분야에서 활용

- "애플리케이션 테스트 관리"는 소프트웨어 개발 과정에서 중요한 단계 중 하나로, 애플리케이션 또는 시스템이 요구사항을 만족하는지, 오류 없이 정상적으로 작동하는지 확인하기 위해 실시하는 테스트 과정을 계획, 실행, 관리하는 것을 말함, 이는 품질 보증 및 위험 관리를 위해 필수적인 과정

# 교과목 소개

## • 교과목 개요

- 이 교과목에서는 요구사항 확인과 애플리케이션 테스트 관리에 대해 학습한다. 요구사항 확인 학습을 위해 다양한 테스트 기법에 대해서 학습한다. 또한 요구사항을 단위 테스트로 표현하고 작성하는 방법을 학습한다. 애플리케이션 테스트 관리 기법 학습을 위해 ISO 29119를 기반으로 명세기반 테스트, 구조기반 테스트 및 키워드 기반 소프트웨어 테스트 기법에 대해서 학습한다. 이 교과목에서는 JUnit 단위 테스트 프레임워크와 Robot 키워드 기반 테스트 프레임워크 등을 포함한 오픈소스 소프트웨어를 활용한 실습을 통해 이해를 높이고 현장에 바로 적용할 수 있도록 한다.

## • 수업 목표

- 요구사항 확인
- 애플리케이션 테스트 관리

# 수업 안내

- 강의 진행 방법

- 이론 강의와 실습

- 선수과목

- 객체지향프로그래밍(JAVA)

- 주교재, 강의 교안

- 정인상 2020 소프트웨어 테스팅: 이론과 실습 한성대학교출판부
- 부교재 : 오픈소스 소프트웨어로 실습하는 소프트웨어공학 정인상 생능 출판 2017

- 기타 안내 사항

- 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	강의 소개 및 오리엔테이션
2	요구사항 분석 1
3	요구사항 분석 2
4	ISO 29119 소개 - 소프트웨어 테스팅 개요 1.1 소프트웨어 테스팅 1.2 소프트웨어 에러(error), 결함(fault) 그리고 오작동(failure) 1.3 RIP 모델 1.4 테스트 케이스 설계 1.5 테스트 유형(Test Type) 1.6 테스트 레벨 1.7 리그레션 테스팅과 리테스팅
5	5장 Gradle 소개 177 5.2 JUnit 5 소개 182 5.3 실습 예제 183 5.4 @DisplayName 어노테이션 196 5.5 테스트 라이프 사이클 199
6	5장 JUnit 프레임워크 5.6 예외 테스팅 207 5.7 @Nested 어노테이션 210 5.8 Parameterized 테스트 219 5.9 @Tag 대면 수업 - JUnit을 이용한 테스트 주도 개발(TDD: Test Driven Development) 강의 및 실습



# 수업 안내

- 수업 일정 ※ 강의계획서의 주차별 내용은 상황에 맞추어 변동 가능성이 있음을 알려드립니다.

주차	수업내용 및 학습활동
7	6장 Mockito 231 6.1 Mockito 소개 237 6.2 Mockito 설치 및 예제 프로그램 240 6.3 모의 객체 생성 및 스텀빙 244 6.4 호출 검증 251 6.5 when(~).thenReturn(~)과 doReturn(~).when(~) 255
8	중간 고사
9	2장 명세 기반 테스트 2.1 동등 분할(Equivalence Partitioning) 2.3 경계 값 분석(Boundary Value Analysis), 2.4 선택스 테스트(Syntax Testing) 2.5 조합 테스트(Combinatorial Testing), 2.6 결정표 테스트(Combinatorial Testing) 2.8 상태 전이 테스트(State-Transition Testing)
10	ISO 29119 소개 - 소프트웨어 테스트 개요 1.1 소프트웨어 테스트 1.2 소프트웨어 에러(error), 결함(fault) 그리고 오작동(failure) 1.3 RIP 모델 1.4 테스트 케이스 설계 1.5 테스트 유형(Test Type) 1.6 테스트 레벨 1.7 리그레션 테스트와 리테스팅
11	구조 기반 테스트 3.1 제어 흐름 그래프(Control Flow graph) 3.2 문장 테스트(Statement Testing) 3.3 분기 테스트(Branch Testing) 3.4 결정 테스트(Decision Testing), 3.7 다중 조건 테스트(Multiple Condition Testing) 3.8 변형된 조건/결정 테스트(Multiple Condition Decision Testing) 대면 수업 - JaCoCO를 이용한 커버리지 실습

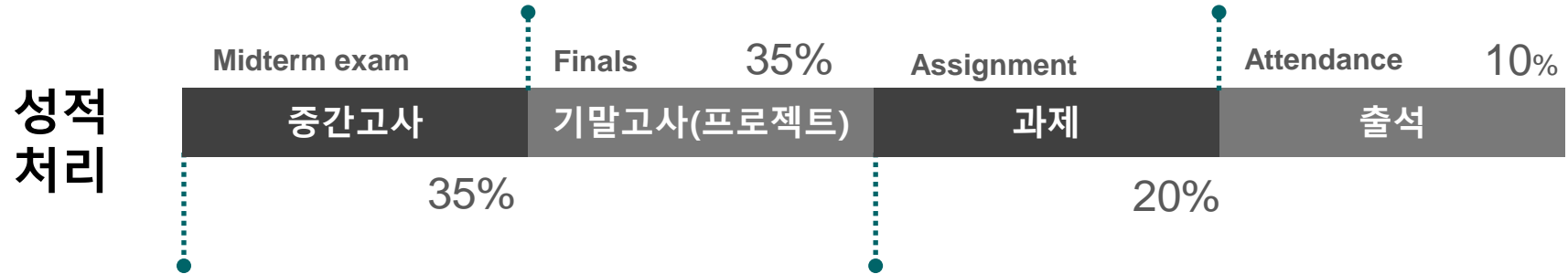
# 수업 안내

- 수업 일정      ※ 강의계획서의 주차별 내용은 상황에 맞추어 변동 가능성이 있음을 알려드립니다.

주차	수업내용 및 학습활동
12	구조 기반 테스트 3.9 기본 경로 테스트(Basis Path Testing)
13	키워드 기반 테스트(Keyword Driven Testing) 147 4.1 개요 151 4.2 키워드와 계층(layer) 157 온라인 실습: 키워드 기반 테스트 9.1 Robot 프레임워크 설치 및 실습 환경 구성 9.2 첫 번째 테스트 케이스 작성하기 9.3 변수 9.4 키워드 기반 테스트 케이스 만들기
14	키워드 기반 테스트(Keyword Driven Testing) 4.4 데이터 기반 테스트(Data-driven Testing)
15	기말고사

# 수업 안내

## • 성적 평가



※ 출석 (매시간 본인이 직접 '출결 여부'를 잘 확인해 주셔야 합니다.)

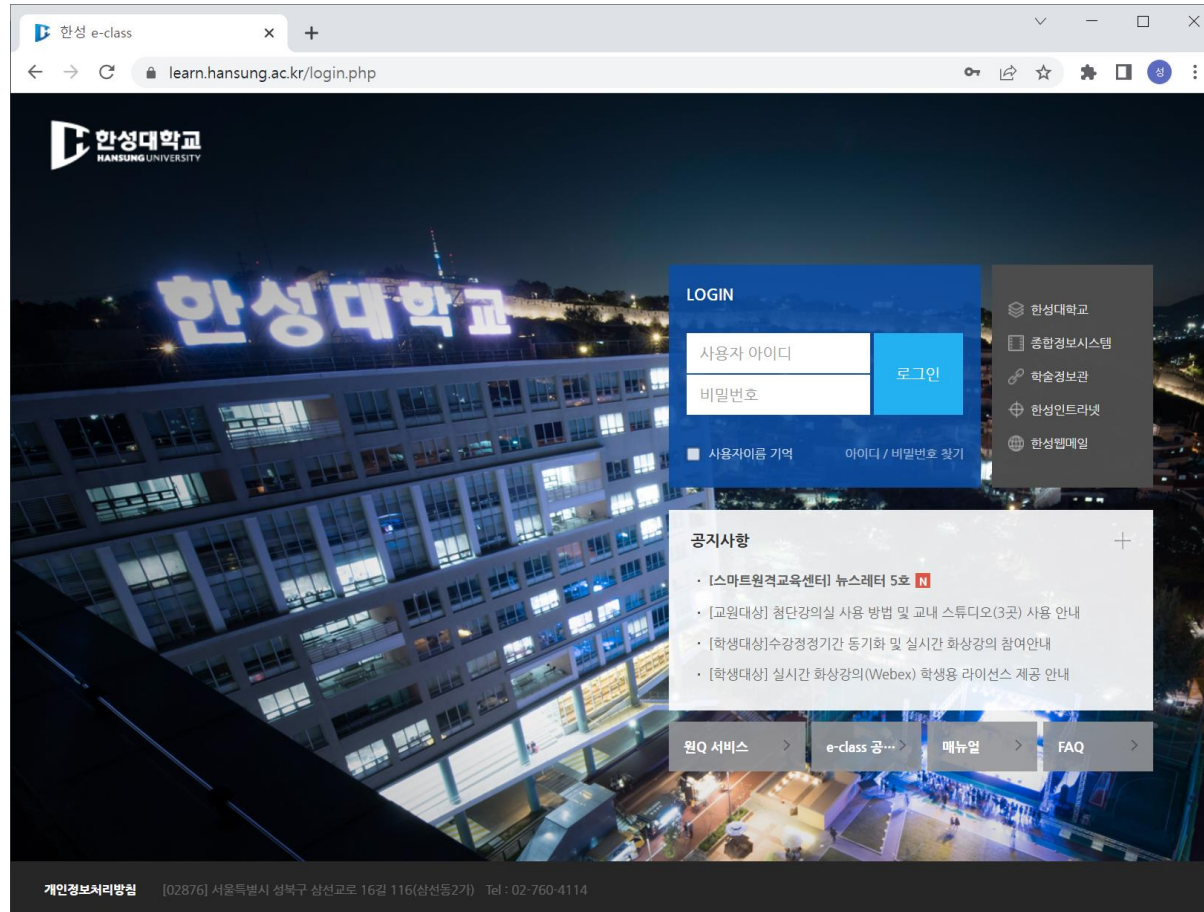
- 결석 1회당 1점 감점(1주당 최대 2점 감점, 온라인 1점, 대면 1점 ), 결석 4회 이상 F학점
- 지각 3회는 결석 1회로 간주

# 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 있으셨으면 좋겠습니다.

단순히 암기하지 말고, 근본적인 원리와 개념부터 이해를 하려고 노력하시고(기본이 제일 중요),  
또한 계속 주어진 문제를 해결하면서 반복된 성취감을 느끼다 보면 재미가 생김

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  
plicabo ipsum, labore sed tempora ratione asperiores des  
quenerat bore sed tempora rati jgert one bore sed tem!