

FIT2081 모바일 애플리케이션 개발 - 2025년 1학기

과목 정보

환영합니다 FIT2081에 오신 것을 환영합니다. 제목에서 알 수 있듯이 이 과목은 모바일 애플리케이션 개발에 중점을 둡니다. 이 텍스트 왼쪽에 있는 비디오(원문 참조)에서는 과거 학기의 주요 피드백뿐만 아니라 이 과목의 핵심 의도, 평가 항목 및 구조를 소개합니다. 이 과목 전반에 걸친 여러분의 피드백은 우리가 학습 콘텐츠를 지속적으로 개선하는 방법을 알려주는 데 매우 소중합니다.

과목 운영팀을 대표하여 FIT2081과 여러분이 수행 중인 다른 모든 과목에서 행운이 있기를 바랍니다. 우리가 준비한 콘텐츠와 수업 활동을 즐기시기 바라며, 캠퍼스 활동에서 여러분을 만나기를 고대합니다.

과목 자료

핸드북 및 과목 미리보기 과목의 전반적인 학습 목표를 지원하지만 필수 도서 목록에는 포함되지 않는 자료들입니다. 여기에는 일반적인 참고 자료나 과목 전체에 적용되는 도구들이 포함될 수 있습니다.

Leganto 도서 목록 Leganto 도서 목록 링크가 포함되어 있습니다. 도서 목록과 관련된 질문이나 문제는 일차적으로 담당 강사에게 문의하십시오.

과목 개요 (FIT2081 2025)

이 과목은 스마트폰과 태블릿을 위한 모바일 애플리케이션 개발의 맥락에서 산업용 수준의 프로그래밍 언어(지원 소프트웨어 기술 및 표준 포함)와 객체 지향 애플리케이션 개발을 소개합니다. 접근 방식은 철저하게 애플리케이션 중심입니다. 여러분은 신중하게 단계별로 구성된 완성된 애플리케이션을 분석함으로써 선택한 언어의 문법과 의미론, 지원 기술 및 표준, 그리고 객체 지향 설계 및 코딩 기술을 배우게 됩니다. 또한 여러분은 자신만의 애플리케이션을 설계하고 구축하게 될 것입니다.

학습 성과 (FIT2081 2025)

1. 모바일 애플리케이션 인프라와 그 구성 요소의 수명 주기(Lifecycle)를 이해한다.
2. 최고의 산업 관행을 따르는 모바일 애플리케이션을 위한 효율적이고 반응형인 UI 및 데이터베이스를 설계한다.
3. 프레임워크, API 및 크로스 플랫폼 기능을 모바일 애플리케이션 설계에 통합한다.

4. 실행되는 장치의 하드웨어와 기능을 모두 활용하여 현실 세계의 실질적인 문제를 해결하는 모바일 애플리케이션을 만든다.

주차별 계획

1. 환영 및 소개

- **강사:** Delvin Varghese (책임 교수 및 과목 코디네이터), Joe Liu
- **교육팀:**
 - Norman Chen (강의 지원 및 Q&A 중재자)
 - Clair Pan (행정 튜터)
 - 실습 워크숍을 위한 다수의 튜터들
- **과목 개요:**
 - **과목 규모:** Clayton 캠퍼스 1,100명, 말레이시아 캠퍼스 400명
 - **과목 유형:** 많은 튜터가 있는 초대형 과목
 - **학습 접근 방식:** 실제 애플리케이션 구축에 중점을 둔 매우 실용적인 방식

2. FIT2081이란 무엇인가?

- **주요 목표:** 학생들이 실제 모바일 애플리케이션을 개발할 수 있는 기술을 갖추도록 함.
- **핵심 언어:** Kotlin (최신 기능으로 인해 Java에서 전환됨)
- **주요 프레임워크:** UI 개발을 위한 Jetpack Compose
- **학습 철학:** 애플리케이션 주도형, 실습 중심의 프로그래밍 경험
- **핵심 요점:** 안드로이드 개발은 단순히 Kotlin을 배우는 것 이상입니다. 이는 모바일 앱 아키텍처, 생태계, UI/UX 원칙을 이해하는 것입니다.

3. 실시간 설문: 학생들의 프로그래밍 배경

- **질문:** 이전에 어떤 프로그래밍 언어를 사용해 보셨습니까?
- **일반적인 응답:** Python, Java, JavaScript, C++, R, C#, Unity, 그리고 일부 Kotlin.
- **관찰:** 대다수가 Python과 Java 경험이 있지만, 이전에 Kotlin을 사용해 본 학생은 거의 없습니다.

4. 안드로이드 개발을 위해 왜 Kotlin인가?

- **Google의 안드로이드 공식 언어:**
 - 더 나은 널(Null) 안전성 처리 (NullPointerException 방지)
 - Java에 비해 간결한 문법
 - 간소화된 비동기 프로그래밍을 위한 코루틴(Coroutines)
 - 기존 Java 코드베이스와 상호 운용 가능

왜 UI를 위해 Jetpack Compose를 사용하는가?

- 선언형(Declarative) UI 접근 방식 (React와 유사)
- @Composable 함수로 UI 개발 간소화
- XML 기반 레이아웃 불필요
- 더 효율적인 UI 렌더링

5. 코스 구조 및 주차별 세부 내용

- **1주 차:** Kotlin 및 Android 소개
- **2주 차:** Kotlin 언어 기초
- **3주 차:** Android 프로젝트 구조 및 UI 레이아웃
- **4주 차:** Kotlin 객체 지향 프로그래밍(OOP) 및 UI 구성 요소
- **5주 차:** 내비게이션(Navigation) 및 인텐트(Intent)
- **6주 차:** 데이터 처리 및 영속성(Persistence)
- **7주 차:** 네트워킹 및 API
- **8주 차:** 코루틴 및 비동기 프로그래밍

- **9주 차:** MVVM 아키텍처
- **10주 차:** 고급 Compose UI 및 테마 적용
- **11주 차:** 배포 및 최종 프로젝트 쇼케이스
- **12주 차:** 크로스 플랫폼 앱 개발 및 퀴즈

6. 코스 운영 및 평가

평가 구성 요소:

- **과제 1:** 기초 앱 개발 (20%)
- **과제 2:** 심화 앱 개발 (30%)
- **학기 중 퀴즈:** 이론 기반 (20%)
- **동료 참여 활동:** 참여 기반 (10%)
- **앱 비평 보고서:** 기존 앱 평가 (10%)
- **제출 정책:** 퀴즈와 동료 참여 활동을 제외하고 연장 허용.
- **AI 사용:** 생성형 AI 도구 사용은 허용되나 반드시 명시(declare)해야 함.

7. 실습 워크숍 설정

라이브 데모: 안드로이드 스튜디오 설정

- **1단계:** 안드로이드 스튜디오 설치 및 구성
- **2단계:** 간단한 "Hello World" 안드로이드 앱 생성
- **3단계:** 프로젝트 구조, Gradle 종속성 및 에뮬레이터 설정 탐색
- **4단계:** @Composable 함수 및 미리보기 도구 소개

8. 리소스 및 지원

- **주요 학습 플랫폼:** Ed Discussion Forum
 - 이메일 대신 모든 질문을 이곳에 게시하십시오.
 - 공개 및 비공개 게시물 작성 가능
- **Moodle:** 주로 과제 제출용으로 사용
- **상담 세션:** 2주 차부터 이용 가능
- **튜터:** 실습 및 Ed Discussion에서 도움을 줄 수 있는 20명 이상의 튜터가 있음

9. 마무리 및 다음 단계

- **다음 강의:** Kotlin 문법 및 디버깅 기술 심층 탐구
- **실천 항목 (Action Items):**
 - 안드로이드 스튜디오 설치 및 환경 설정
 - Ed에 있는 코스 자료와 친해지기
 - Ed Discussion에 참여하고 자기소개하기
- **Q&A 세션:** 학생들의 질문과 명확한 설명을 위한 오픈 플로어