Course Syllabus

[Jump to Today](#)

MOBICOM

Mobile Computing

Type of Course : Foundation Course

Pre-requisites : CCAPDEV - Web Applications Development (H)

Co-requisite : n/a

Prerequisite to : n/a

Equivalent to : n/a

Term / Academic Year : Term 3, AY 2024-2025

Class Schedules : S16 T 0915-1045 (Online) F 0915-1045 (GK301)
S17 T 0915-1045 (Online) F 1100-1230 (GK301)

Instructors : Oliver Berris (S16, S17) - **oliver.berris@dlsu.edu.ph (mailto:oliver.berris@dlsu.edu.ph)**

Consultation Hours : By appointment
Please book an appointment via email at least 1 day before your preferred consultation schedule. Consultation hours vary per instructor.

Course Group : AnimoSpace - **https://dlsu.instructure.com/**

Estimated time for study outside of class : 4.5 hours per week

Course Description

This course exposes the students to the field of mobile application development and its integration with existing applications, be it mobile, online, or standalone. The course also exposes the students to the latest developments in the field of mobile computing. The course will primarily tackle Android development.

Learning Outcomes (LOs)

Upon completion of this course, the student is expected to be able to do the following:

Expected Lasallian Graduate Attributes	Learning Outcomes
Critical and Creative Thinker, Reflective Lifelong Learner	**LO1.** Understand the issues and design considerations of a mobile application in relation to recent developments in technology.
Critical and Creative Thinker, Service-Driven Citizen, Reflective Lifelong Learner	**LO2.** Be aware of ground developments and advancements in technology and mobile application design to address real-world requirements and problems in order to promote social awareness of social responsibility.
Critical and Creative Thinker	**LO3.** Improve on the ability to read and appreciate API documentation.
Effective Communicator	**LO4.** Work harmoniously with teammates, exhibiting versatility in developing mobile applications.

Major Course Output (MCO)

As evidence of attaining the above learning outcomes, the student is required to do and submit the following during the indicated dates of the term.

Learning Outcome	Required Output	Due Date
LO1, LO2, LO4	MCO1. Project Proposal*	Week 4 (31 May 2025, S)

LO1, LO2, LO3, LO4	MCO2. Interactive Prototype (UI only)	Week 7 (21 Jun 2025, S)
LO1, LO2, LO3, LO4	MCO3. Final Application + Demo	Week 13 (28 Jul 2025, M)

MCO1 will have no weight in the grading scheme. Submitting MCO1 late will result in a deduction of 2% per day late to the final assessed grade. MCO2 and MCO3 will not be checked if MCO1 was not submitted. Late submission for MCO2 and MCO3 will not be accepted.

Although the major course outputs are all to be done collaboratively in groups, every student is expected to contribute to their group's work and will be graded accordingly. All group members are also expected to keep track of their own work contributions (preferably through version control) and should be ready to discuss these with the teacher whenever the need arises.

Rubrics for MCO Assessment

To follow once finalized.

Other Requirements and Assessments

The course has the following requirements on top of the major course output described above:

1. **Class Exercises**
   - Exercises involve addressing issues / tasks based on given specifications. Usually, an exercise covers one topic.
2. **Android Challenges**
   - Android Challenges involve the creation of an application based on given specifications. These are more summative in nature and cover at least two topics. There are two challenges throughout the term. All Android Challenges will be conducted in a physical setting.

Grading System

To pass this course, one must accumulate at least 60 points through the course requirements discussed above. The breakdown of the assessments is shown below.

Assessment Task	Maximum Points
Class Exercises	30

Raw	GPE	Raw	GPE
≥ 94	4.0	≥ 72	2.0

Android Challenge #1	15		≥ 89	3.5	≥ 66	1.5
Android Challenge #2	15		≥ 83	3.0	≥ 60	1.0
MCO1. Project Proposal	0		≥ 78	2.5	< 60	0.0
MCO2. Interactive Prototype (UI only)	5					
MCO3. Final Application + Demo	35					
TOTAL POINTS	**100**					

Teaching Methods/Strategies

1. Lectures
2. Recitation and Hands-on Coding
3. Online Reading
4. Graded / Non-graded Exercises
5. Android Challenges
6. Machine Project

Learning Plan

Learning Outcomes	Topics and Readings	Week/Date	Learning Activities
	Course Orientation Syllabus, Requirements, Class policies, Machine project grouping, Machine project specifications	Week 1	
LO1, LO2, LO3	**Introduction to Android Mobile Development** Overview of Android Development, Android Architecture, Setting up Android Studio and IDE familiarity, Application structure, Tools for debugging	Weeks 1	Lecture, Hands-on, Formative exercises
LO1, LO2, LO3	**Basic Layouting Development** Resources, Views, Layouts, Data-driven views, MVC pattern	Weeks 2 - 5	Lecture, Hands-on, Formative exercises

	Activities Overview, Multiple activities, Intents, Fragments		
LO1, LO2, LO3	**Android Challenge #1**	Week 6	Hands-on Challenge
LO1, LO2, LO3	**Activities Cont.** Lifecycle callback methods **Basic Storage** Writing to/reading from SharedPreferences and DataStore	Week 7	Lecture, Hands-on, Formative exercises
LO1, LO2, LO3	**Local Storage** SQLite, Helpers **Remote Storage** Firebase console and setup, Firebase Real-time, Firestore, Storage	Weeks 8-10	Lecture, Hands-on, Formative exercises
LO1, LO2, LO3	**Process Management** Concurrency, Main and worker threads, Handlers, Background services, Service lifecycle, Alarms **Broadcasting Messages** Broadcasts, Receivers	Week 10-11	Lecture, Hands-on, Formative exercises
LO1, LO2, LO3	**Android Challenge #2**	Week 11	Hands-on Challenge
LO1, LO2, LO3, LO4	**Misc. Topics on Android Development** Notifications, Navigation, Screen orientation, Internal and External Storage, GPS, Gestures, Camera, Handling Multiple Screen Sizes, Publishing Applications **Consultation Sessions** **Wrap-up**	Weeks 11-12	Lecture, Hands-on, Formative exercises
	MCO Demos	Week 13	

Final Exam Week (Week 14; Aug 4 to 9, 2025)

* The lesson plan may be adjusted to accommodate changes / shifts made throughout the term. However, for the sake of finality, all dates are considered final unless otherwise announced.

References

- Textbooks
  - Deitel, Paul J. (2013). Android : how to program, [with an introduction to Java] Boston : Pearson. [QA76.76.A65 D44 2013]
  - Haseman, Chris. (2012). Creating Android applications : develop and design. Berkeley, Calif. : Peachpit Press. [QA76.59 .H37 2012]
  - Ostrander, Jason. (2012). Android UI fundamentals : develop and design Berkeley, CA : Peachpit Press. [QA76.76.A65 O87 2012]
  - Huddleston, R. (2011). Android fully loaded. Hoboken, NJ: Wiley. [QA76.76.A65 H84 2012]
  - Hoisington, Corinne. (2013). Android Boot Camp for developers using Java, comprehensive : a beginner's guide to creating your first. Boston, Mass: Course Technology/Cengage Learning. [**QA76.73.J*38 H*636 2013** ⤷ **(https://lib1000.dlsu.edu.ph/search~S19?/cQA76.73.J38+H636+2013/cqa+++76.73+j38+h636+2013/-3,-1**]
  - Deitel, Paul. (2012). Android for programmers : an App-driven approach. Upper Saddle River, NJ: Prentice Hall. [**QA76.76.A*65 A*54 2012** ⤷ **(https://lib1000.dlsu.edu.ph/search~S19?/cQA76.76.A65+A54+2012/cqa+++76.76+a65+a54+2012/-3,-1,,**]
  - Yener, M., & Dundar, O. (2016). Expert Android Studio. In *Expert Android Studio* (1st ed.). John Wiley & Sons, Incorporated. [**QA76.774.A53.Y46 2016** ⤷ **(https://animosearch.dlsu.edu.ph/permalink/63DLSUL_INST/g01tfu/cdi_skillsoft_books24x7_bks0001**]
  - DiMarzio, J. (2017). Beginning Android programming with Android Studio (Fourth edition). Wrox, a Wiley brand. [**QA76.774.A53 D55 2017** ⤷ **(https://animosearch.dlsu.edu.ph/permalink/63DLSUL_INST/1rjqea9/alma991002184869706351)**]
- Online References
  - Practical Android Projects [**Link** ⤷ **(https://animosearch.dlsu.edu.ph/permalink/63DLSUL_INST/1rjqea9/alma991001780459706351)**]
  - Android Recipes [**Link** ⤷ **(https://animosearch.dlsu.edu.ph/permalink/63DLSUL_INST/1rjqea9/alma991001848339706351)**]
- Blogs / Communities / MOOCs / Documentation
  - Android Documentation [**https://developer.android.com/docs** ⤷ **(https://developer.android.com/docs)**]

- Material Up: Material Design for Android [**http://www.materialup.com/** ⤷ **(http://www.materialup.com/)**]
- Styling Android: UI and UX for Android [**http://blog.stylingandroid.com/** ⤷ **(http://blog.stylingandroid.com/)**]
- Android Developers Blog: Latest Android Updates [**http://android-developers.blogspot.com/** ⤷ **(http://android-developers.blogspot.com/)**]
- Udacity course : Android for Beginners [**https://www.udacity.com/course/android-development-for-beginners--ud837** ⤷ **(https://www.udacity.com/course/android-development-for-beginners--ud837)**]
- Udacity course : Developing Android Apps [**https://www.udacity.com/course/developing-android-apps--ud853** ⤷ **(https://www.udacity.com/course/developing-android-apps--ud853)**]
- Udacity course : Advanced Android Development [**https://www.udacity.com/course/advanced-android-app-development--ud855** ⤷ **(https://www.udacity.com/course/advanced-android-app-development--ud855)**]

Software Tools

- Android Studio [**Link** ⤷ **(https://developer.android.com/studio)**]
- Emulators (*optional*, as Android Studio has the capacity to create emulators)
  - BlueStacks [**Link** ⤷ **(https://www.bluestacks.com/)**]
  - GenyMotion [**Link** ⤷ **(https://www.genymotion.com/)**]

General Policies

- All University policies on plagiarism, cheating, and proper decorum, among others, will be enforced. Refer to the Student Handbook for the details of these policies, as well as any reminder announcements made via HelpDesk.
- Additional course policies, as enumerated below, will also be enforced. Additional class policies may be provided by your respective instructors.
- Students are expected to read (at least skim) the course material prior to the discussion of the topic. Students are encouraged to ready and ask questions, particularly when concepts are not clear.
- Students are expected to practice applying concepts (e.g. studying new programming languages) outside of class time, even when there is no homework assigned.
- Students are expected to attend online classes on time and actively participate in class activities.
- A student who was absent in a previous meeting/s is expected to catch up on the missed discussion himself/herself (by asking his/her peers what he/she had missed).
- When using the intellectual property of others (such as the use of images, source code, tools, and other software), please be reminded to properly cite your sources. Be mindful of the bounds of submitting work of which the idea came from another.
- Materials made available to the class (lectures, exercises, slides, recordings, etc.) are only for the sole use of the students enrolled in this course this term. Unauthorized screen captures, video captures, and/or distribution/sharing of materials are not allowed.

- The information and data contained in the online learning modules, such as the content, audio/visual materials or artwork are considered the intellectual property of the author and shall be treated in accordance with the IP Policies of DLSU. They are considered confidential information and intended only for the person/s or entities to which they are addressed. They are not allowed to be disclosed, distributed, lifted, or in any way reproduced without the written consent of the author/owner of the intellectual property.
- It is recommended that you turn your webcam/camera OFF for online sessions.
- Be mindful of what you post in the chat messages or discussion pages and be considerate to others (i.e., show respect for others' opinions). Only post comments/questions pertinent to the discussion. Additionally, if you are to present something, be mindful of the information that is currently on your machine's screen.
- Not following instructions in the requirements for submission may result in non-acceptance of the submission (and thus, it is considered a missed exercise).
- All assessments are to be submitted based on the indicated due date on Canvas. When uncertain of a deadline, reach out to your instructor.
- By default, late submissions will not be accepted. Additionally, missed activities and deliverables (e.g. cases, exercises, short quizzes, projects) cannot be made up unless a valid excuse is presented.
- Only valid and verifiable reasons for missed examinations will be considered for make-up.
- No special projects will be given. Final grade computation is based on the same set of activities and deliverables as expected from the entire class.
- Concerns and/or disputes regarding the results of assessments should be raised by the student within a reasonable timeframe from the release of the grades. All results are considered final.
- Kindly observe proper coordination with your group (for group-based assessments). If there are issues with unresponsive members, kindly consult your instructor.

Prepared by:

Oliver Berris
Lecturer, Department of Software Technology

Arturo Caronongan
Assistant Professorial Lecturer, Department of Software Technology

Oliver Berris
Professional Lecturer, Department of Software Technology

Noted by:

Niel del Gallego, PhD
Associate Professor and Chair, Department of Software Technology

Course Summary:

Date	Details	Due
Sat May 31, 2025	📝 **MCO1: Proposal Submission** (https://dlsu.instructure.com/courses/209040/assignments/2063375)	due by 11:59pm
Sat Jun 21, 2025	📝 **MCO2: Interactive Prototype** (https://dlsu.instructure.com/courses/209040/assignments/2063376)	due by 11:59pm
Mon Jul 28, 2025	📝 **MCO3: Final App + Demo Grade Holder** (https://dlsu.instructure.com/courses/209040/assignments/2063377)	due by 7:31am
	📝 **MCO3: Final Application Submission** (https://dlsu.instructure.com/courses/209040/assignments/2063378)	due by 7:31am
	📝 **MCO3: Version Control Submission** (https://dlsu.instructure.com/courses/209040/assignments/2063379)	due by 7:31am