

**College of San Mateo**  
**Official Course Outline**

1. **COURSE ID:** CIS 135    **TITLE:** Android Programming  
**Units:** 4.0 units    **Hours/Semester:** 48.0-54.0 Lecture hours; 48.0-54.0 Lab hours; and 96.0-108.0 Homework hours  
**Method of Grading:** Grade Option (Letter Grade or Pass/No Pass)  
**Recommended Preparation:**  
    CIS 254
  
2. **COURSE DESIGNATION:**  
**Degree Credit**  
**Transfer credit:** CSU; UC  
**AA/AS Degree Requirements:**  
    CSM - GENERAL EDUCATION REQUIREMENTS: E2b. Communication and Analytical Thinking
  
3. **COURSE DESCRIPTIONS:**  
**Catalog Description:**  
    Introduction to the architecture, API and techniques used to create robust, high-performance applications for Android mobile devices. An overview of the most common tools and techniques for writing Android applications. Other topics include user interfaces, local storage, maps, multimedia, content providers, sensors, and user events. Storage strategies for persistent information are introduced, including the use of SQLite database features. Introduction to Kotlin programming. Intended for students with previous Java programming experience.
  
4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**  
    Upon successful completion of this course, a student will meet the following outcomes:
  1. Explain the Android OS architecture.
  2. Install and use appropriate tools for Android development, including IDE, device emulator, and profiling tools.
  3. Build user interfaces with fragments, views, form widgets, text input, lists, tables, and menus.
  4. Employ advanced UI widgets for scrolling, tabbing, and layout control.
  5. Store application data on the mobile device, in internal or external storage locations.
  6. Create an advanced mobile application employing sensors, maps, and other features.
  
5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**  
    Upon successful completion of this course, a student will be able to:
  1. Explain the Android OS architecture.
  2. Install and use appropriate tools for Android development, including IDE, device emulator, and profiling tools.
  3. Build user interfaces with fragments, views, form widgets, text input, lists, tables, and menus.
  4. Employ advanced UI widgets for scrolling, tabbing, and layout control.
  5. Store application data on the mobile device, in internal or external storage locations.
  6. Create an advanced mobile application employing sensors, maps, and other features.
  
6. **COURSE CONTENT:**  
**Lecture Content:**
  1. Android Overview and History
    - Origin of Android
    - Importance of Android
    - Relationship of Java and Android
  2. Android Stack
    - Overview of the stack
    - Linux kernel
    - Native libraries
    - DalvikApp framework
    - Apps
  3. SDK Overview
    - Platforms

- Tools
- Versions
- 4. Creating an App
  - The manifest file
  - Layout resource
  - Running an app on the Emulator
- 5. Main Building Blocks
  - Activities
  - Activity lifecycle
  - Intents
  - Services
  - Content Providers
  - Broadcast Receivers
- 6. Basic Android User Interface
  - XML versus Java UI
  - Dips and sps
  - Views and layouts
  - Common UI components
  - Handling user events
- 7. Android System Overview
  - File System
  - Preferences
  - Notifications
  - Security model
- 8. Advanced UI
  - Selection components
  - Adapters
  - Complex UI components
  - Building UI for performance
  - Menus and Dialogs
  - Graphics & animations
- 9. Multimedia in Android
  - Multimedia Supported audio formats
  - Simple media playback
  - Supported video formats
  - Simple video playback
- 10. SQL Database
  - Introducing SQLite
  - SQLiteOpenHelper and creating a database
  - Opening and closing a database
  - Working with cursors
  - Inserts, updates, and deletes
- 11. Basic Content Providers
  - Content provider MIME types
  - Searching for content
  - Adding, changing, and removing content
  - Working with content files
- 12. Advanced Topics
  - XML Parsing
  - JSON Parsing
  - Fragment API
  - Including external libraries in an application
  - Maps via intent and MapActivity
  - Network connectivity services
  - Sensors, Camera
  - Using Wi-Fi & Bluetooth
  - Testing
  - Threads
  - Web Apps
  - Introduction to Kotlin programming

## **Lab Content:**

1. App Widgets
2. The Android Manifest
3. User Interface
4. App Resources
5. Animation and Graphics
6. Computation
7. Handle User Events
8. Create Menus and Dialogs
9. Media
10. Connectivity
11. Text and Input
12. Create and Use Data Storage
13. SQLite Database
14. Local Storage
15. Administration and Testing
16. Web Apps
17. Kotlin Programming

## **7. REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

- A. Lecture
- B. Lab
- C. Activity
- D. Discussion
- E. Observation and Demonstration
- F. Other (Specify): • Student reading of textbooks and supplemental course materials • Individual and team programming projects • Review of subject matter videos

## **8. REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

### **Writing Assignments:**

Students will complete and submit exercises and programming assignments on a weekly or biweekly basis. Assignments will cover:

- A. App Widgets
- B. The Android Manifest
- C. User Interface
- D. App Resources
- E. Animation and Graphics
- F. Computation
- G. User Events
- H. Menus and Dialogs
- I. Media
- J. Connectivity
- K. Text and Input
- L. Create and Use Data Storage
- M. SQLite Database
- N. Local Storage
- O. Administration and Testing
- P. Web Apps
- Q. Kotlin Programming

### **Reading Assignments:**

Students will read assigned chapters in the textbook and supplemental handouts.

## **9. REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

- A. Class Work
- B. Exams/Tests
- C. Group Projects

- D. Homework
- E. Lab Activities
- F. Projects
- G. Quizzes
- H. Written examination

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

- A. Smyth, N. *Android Studio 3.6 Development Essentials Kotlin Edition*, 2nd ed. Payload Media, 2020
- B. Marsicano, K. and Gardner, B. *Android Programming: The Big Nerd Ranch Guide*, 4th ed. Big Nerd Ranch Guides, 2019
- C. Horton, J. *Android Programming with Kotlin for Beginners*, 1st ed. Packt, 2019

**Origination Date:** November 2020

**Curriculum Committee Approval Date:** November 2020

**Effective Term:** Fall 2021

**Course Originator:** Melissa Green