



Course Information Sheet

CSCI 4060

Mobile Software Development

Brief Course Description
(50-words or less)

Introduction to software development for mobile devices, such as smartphones and tablets. Topics include life cycle of mobile applications, mobile UI design, views, widgets, location and maps, local data handling, and interaction with Web services and databases. Students design, implement, and analyze mobile applications.

Extended Course Description / Comments

This course is as an intensive introduction to the design and development of mobile software. The students learn the principles of software development for the mobile platform, such as Android. Topics include activities, activity lifecycle, variety of views typically used in mobile applications, layouts, fragments, navigational patterns, application resources, as well as themes and styles. Since mobile applications frequently require a variety of data sources, the students will also learn mobile applications that incorporate relational databases and Web services. Students will learn the application development processes with emphasis on agile development, and elements of mobile screens and navigation design. The course includes several individual and student pair projects.

Pre-Requisites and/or Co-Requisites

Prerequisite: CSCI 2720

Required, Elective or Selected Elective

Elective Course

Approved Textbook

Introduction to Android Application Development: Android Essentials, 5th Edition, by Jr. Anuzzi Joseph, Lauren Darcey, and Shane Conder. Addison-Wesley, 2016.
Edition: 5th Edition
ISBN-13: 978-0321940261

Specific Learning Outcomes (Performance Indicators)

1. Plan, design, implement, test, deploy a complete mobile application.
2. Utilize a suitable development environment, such as Android Studio, in app development.
3. Understand and use elements of a mobile application, including activities, views, layouts, fragments, and styles and themes.
4. Understand, select and utilize an appropriate screen designs and a proper navigational pattern.
5. Be able to manage application's resources and files and be able to utilize a relational database and an external Web Service.
6. Be able to apply a software process, such as an agile development, in app development.
7. Effectively describe and demonstrate a functioning app.

ABET Learning Outcomes

- A. Graduates of the program will have an ability to: Analyze a complex

computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

- B. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
- C. Communicate effectively in a variety of professional contexts.
- D. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- E. Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.
- F. Apply computer science theory and software development fundamentals to produce computing-based solutions.

NOTE: In the construction of the student learning outcomes for this course, the instructors interpreted “computing requirements” in (B) as the functional requirements for a software solution and not as specific hardware requirements for the target platform; likewise, the phrase “[a]pply computer science theory” in (F) was interpreted as using computer science principles.

Relationship Between Student Outcomes and Learning Outcomes

Specific Learning Outcomes	ABET Learning Outcomes						
		a	b	c	d	e	f
1	●	●					●
2	●	●					●
3	●	●					●
4	●	●					●
5	●	●					●
6	●	●	●			●	●
7			●			●	●

Major Topics Covered

1. Mobile Development Basics
 - a) Android history and versions
 - b) Android Architecture and Linux
 - c) Java and Event-driven programming
2. Android Development Environment
 - a) Android development environment setup
 - b) Simple Android application
 - c) Android project structure
 - d) Introduction to XML
3. Android Fundamentals
 - a) Android application components
 - b) Activity and Activity lifecycle
 - c) Common views
 - d) Android Layouts

- e) Fragments, fragment lifecycle, fragment transactions
 - f) Managing resources
 - g) Navigational patterns
 - h) Styles and themes
4. Interacting with Data Sources
 - a) Relational databases
 - b) SQLite
 - c) RESTful Web Services
 5. Application Development Process
 - a) Software process
 - b) Development workflow
 - c) Screen design, screen maps, wireframes

Knowledge Levels

The following is the ACM’s categorization of different levels of mastery: Assessment, Usage, and Familiarity. Note that Assessment encompasses both Usage and Familiarity, and Usage encompasses Familiarity.

Familiarity: The student understands what a concept is or what it means. This level of mastery concerns a basic awareness of a concept as opposed to expecting real facility with its application. It provides an answer to the question “What do you know about this?”

Usage: The student is able to use or apply a concept in a concrete way. Using a concept may include, for example, appropriately using a specific concept in a program, using a particular proof technique, or performing a particular analysis. It provides an answer to the question “What do you know how to do?”

Assessment: The student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem. This level of mastery implies more than using a concept; it involves the ability to select an appropriate approach from understood alternatives. It provides an answer to the question “Why would you do that?”

Course Master

Dr. Krzysztof J. Kochut

Modified

Approved

No