



Department of Computer and Information Sciences
COSC 435 – Mobile Application Development

Fall 2024

Instructor: Randy Valis
Office: Blackboard Collaborate
Office Hours: Th 6:30-7:30 (Blackboard Collaborate)
Email: rvalis@towson.edu
Phone:

Credits: 3 credits
Course Hours: M 7:00 – 9:40
Course Location: YR402
Course Website: Blackboard

Prerequisite: COSC 336 – Data Structures and Algorithm Analysis

Required Textbook: None

Rec Textbook: None

Course Description: This is a project-based course for computer science majors that provides students with the experience of working as part of a project team using current software design and development tools and environments. Projects will consist of all aspects of software development, including requirements analysis, design, and implementation. Students will manage the project themselves following appropriate project management techniques and paying attention to quality issues.

This semester's course will focus on the development of mobile applications using the Android platform and using an agile engineering development approach. Specifically, the software development environment employed will use the Android Development Kit, the Android Studio integrated development environment for development and configuration management; the computing platform will be the Android 13.0+ open-source, mobile operating system; and the development methodology will be a modified version of the agile software development framework Scrum. The course will also investigate prototyping tools, testing tools and bug reporting tools aiding in the development of mobile applications.

Course Context / Background: Mobile computing platforms (e.g., smartphones, tablets, etc.) are becoming a prominent and integral part of computing as users are increasingly relying on mobile applications for their communications, business and social activities, entertainment and other scenarios. Mobile application development presents different challenges than “traditional” software systems like tight constraints on response times, space, and computing cycles that require a unique skill set. The Android operating system is currently the fastest growing platform for mobile application development and supports an active community of developers. This class will explore the Android operating system as a mobile application development platform and allow students to experience the development of an Android application in the context of a real-world setting.

Course Objectives: Upon completion of the course, students will understand and experience the complexities of developing large software projects in a “real world” setting. Specifically, this course provides students the opportunity to:

- Experience a specific software development environment, computing platform (i.e., programming languages, operating systems, software and hardware architectures and application programming interfaces) and software development methodology not provided in other course offerings.
- Experience the design, development and implementation of a significant software system as part of a project team.

Course Topics: The conceptual topics covered during the semester will include the following:

- Android activities, intents, services; resource management; application lifecycle
- Database design, object relational mapping
- Software testing: unit testing, integration testing
- Configuration management, version control, software refactoring
- Project planning and management

Additional resources that students may find helpful during their project development include:

- **Android Developers**, <https://developer.android.com>
- **StackOverflow**, <http://stackoverflow.com/questions/tagged/android>
- **Vogella**, <http://www.vogella.com/tutorials/android.html>
- **CodeMentor**, <https://www.codementor.io/android>
- **Udacity**, <http://www.udacity.com>

Students will also be responsible for finding their own resources in support of their application development.

Course Format: Active learning techniques, lectures, discussion sessions (in class and online), presentations, lab exercises, guest speakers and projects may be used. Students are expected to find and use current content on the course subjects using the library, internet and provided resources. Some assignments may necessitate collaborative learning while others will require individual research and presentation.

Class Plan and Schedule: Students will receive tutorials to the Android platform, agile engineering, software configuration management and software testing. In the first portion of the semester, students will be assigned several smaller programming exercises / assignments so that they can demonstrate basic development proficiency in Android and gain experience using the Android Development Tools in Eclipse, the software configuration management tool Subversion and the software testing tools for Android applications.

Project work: The bulk of the course will be dedicated to project work. Students will develop individual project idea that will be assessed by the teams of three or four students and one idea per team will be chosen for development. Development activities will include the creation of design documentation, specification, user interface mockups, status reports and presentations. During the development phase, students must meet regularly with the instructor(s) and, if applicable, related stakeholders invested in the application. These meetings are intended to generate feedback through demonstrations, a report of project status and interactions with the instructor and other interested stakeholders. Each group will be required to formally present and demonstrate their final application at the conclusion of the semester.

Assignments / Labs: There will be several programming assignments given throughout the semester. All work should be thoroughly backed up before turning it in and all submissions should be *well documented giving credit to respective sources with proper citations*. Homework may also be assigned in the lecture at the instructor's discretion and as the need arises. These assignments will be explained in greater detail as the course progresses. Programming labs are meant to provide a "hands on" experience to students in order to emphasize the material recently covered during lecture. Assignments will be given *every week*. These assignments are individual and graded according to the homework rubric. Labs will be given *every week*. Labs are meant to get hands on practice with my help. Labs will be graded on a complete/incomplete status.

Examinations: There will be one exam for this course covering the conceptual topics of the course. The date of the exam will be **Monday, December 16th from 7:30-9:30pm.**

Class Discussions / Group Meetings: A significant portion of the class will be dedicated to discussing issues/challenges related to the design, development and implementation of the semester project. As a part of active learning, students must participate in the discussions (both in class and online) to share their own experiences, issues encountered and plan for the next iteration of the application.

Attendance Policy: Students are expected to attend all classes given the team-project nature of the course. It is the student's responsibility to remain current on all aspects of the project. The instructor will allow only students with documented excuses (see below) to make up missed work or assignments *when it is feasible*. If the student is absent from an exam during the scheduled time for that exam, the student will automatically receive a grade of zero (0) for the exam unless: (a) the student notifies the instructor of the absence prior to the exam; (b) the student is ill and supplies a written doctor's excuse explaining the absence; or (c) there is an extraordinary situation which the instructor allows as an acceptable excuse. Only under one of these circumstances will arrangements for a makeup exam be made.

Grading Policy: Students will be evaluated on the following basis:

Assignments/Labs:	40%
Attendance/Participation:	10%
Project:	35%
Exam:	15%

Final course grades will abide by the following scale:

A	95-100	B-	80-82.9	D	60-66.9
A-	90-94.9	C+	75-79.9	F	Below 60
B+	87-89.9	C	70-74.9		
B	83-86.9	D+	67-69.9		

Professionalism: All materials submitted for this course should look professional including the use of correct grammar and spelling.

Posting of Grades: The instructor will not report grades via email or in response to phone calls. Grades for the semester can be accessed online.

Late work: Late work will **not** be accepted unless otherwise specified.

Cheating and Plagiarism: Cheating or plagiarism will result in a zero grade or an 'F' in class. Any aspect of the team project that has been determined to be plagiarized will be the responsibility of all members of the team. Students are responsible for reading and knowing Towson University's policy regarding academic dishonesty, located in Appendix F in the Undergraduate Catalog and familiarizing themselves with the policies detailed at <http://libraries.towson.edu/citing-sources>.

Computer Lab Information: Lab hours and policies for the Department of Computer and Information Sciences can be found at <http://www.towson.edu/fcsm/departments/computerinfosci/resources/labs.html>, and for the computer labs in Cook Library at <http://libraries.towson.edu/computers-library>. As a part of this course, students may ask the on-duty lab monitor to open YR 402 when it is not in use for work related to this course.

Classroom and Lab Policy: Food and drink are not allowed in the labs and classrooms with the exception of water in the classroom only.

Email Policy: All email correspondence with the instructor must be conducted using the student's Towson University email account (i.e., username@students.towson.edu). The instructor will not read/respond to any email messages from outside accounts.

Repeat Policy: Students may not repeat a course more than once without prior permission of the Academic Standards Committee.

If you have a learning disability and/or need accommodation for any reasons, please advise the instructor as early as possible in the course.

Course Rationale: The course will strengthen the existing Computer Science Undergraduate Program by providing students the opportunity to experience the "real-world" issues regarding team projects, as well as deepen their knowledge in software design, analysis and development through the development of a significant software system. This course integrates topics covered in several courses and synthesizes them towards the development of complex systems.

The department stipulates the following learning goals for the Computer Science major:

- 1 Students can use their proficiency in theoretical and applied computing principles & practices to solve a variety of problems.
- 2 Students can explain the theoretical and applied principles that underlie computer science.
- 3 Students will understand the ethical and societal concerns and dilemmas facing computer scientists and can formulate appropriate solutions and courses of action.
- 4 Students can work effectively in teams and communicate effectively.

Specifically, in relation to the stated learning goals, this proposed course's objectives contribute as follows:

- Experience a specific software development environment, computing platform (i.e., programming languages, operating systems, software and hardware architectures and application programming interfaces) and software development methodologies not provided in other course offerings. (Learning Goals: 1, 2, 3 and 4)
- Design, development and implement a large software system within the context of a "real world scenario". (Learning Goals: 1, 3 and 4)

Overlap Statement: The proposed course integrates topics covered in a number of computer sciences classes and synthesizes them in application towards the development of a large software system. While the overlap with other existing Computer Science courses will vary depending on the particular software development environment, computing platform and development methodology selected by the instructor, the following is an estimated overlap:

- COSC 290 – Principles of Computer Organization –5%
- COSC 350 – Data Communications and Networking – 5%
- COSC 439 – Operating Systems – 5%
- COSC 457 – Database Management Systems – 10%
- COSC 436 – Object-Oriented Design and Programming – 10%
- COSC 412 – Software Engineering – 20%
- COSC 484 – Web-Based Programming – 5%