CS 411 # Mobile Application Development Fall 2023

Instructor: Jonathan Branam **Office:** RB 459
Email: jonathan.branam@bsu.edu

Time and Place: Monday, Wednesday, Friday 5-6pm RB122
Canvas: https://bsu.instructure.com/courses/146045
Office Hours: by appointment; Zoom or in-person depending on schedule
Preferred Contact: 1. Canvas message 2. Email

Course Information

Catalog Course Description

Mobile application development using current technologies. Topics include use of mobile device API, responsive design, location aware applications, hardware sensor applications, resource management, and multimedia applications.

Prerequisite: CS 121 with a C- or better

Learning Objectives

Upon successful completion of this course, a student will be able to:

1. Describe unique aspects of mobile programming
2. Identify, compare, and evaluate various approaches to mobile application development
3. Create multiple mobile applications
4. Apply modern tools of software development to mobile application programming
5. Choose application and platform specific software development methodologies
6. Explain privacy concerns with respect to mobile devices
7. Deploy applications for distribution

Student Background

This is a 400-level course with an official prerequisite of CS 121. Students will be best prepared for this course by previously completing **CS 222 Advanced Programming** with a grade of C- or better. CS 222 covers essential fundamentals of project development such as clean code, source control (git), unit testing and test-driven development. All knowledge and skills required for the course in addition to CS 121 will be covered in the course, however students may find the pace and expectations in this course challenging if they are encountering these topics for the first time. If you have concerns about your ability to complete the course requirements, please talk to me immediately.

Course Materials

Textbooks

There are no required textbooks for this course. The resources listed here are suggestions if you would like additional materials to aid your learning.

- Bonnie Eisenman. Learning React Native: Building Native Mobile Apps with JavaScript, 2nd Ed. O'Reilly Media. 2017. ISBN-10: 1491989149.
- Nader Dabit. React Native in Action. Manning. 2019. ISBN-10: 1617294055.
- Yakov Fain, Anton Moiseev. TypeScript Quickly. Manning. 2020. ISBN-10: 1617295949.

- Josh Goldberg. Learning TypeScript: Enhance Your Web Development Skills Using Type-Safe JavaScript. O'Reilly Media. 2022. ISBN-10: 1098110331.

Computer Requirements

As for all computer science courses, a laptop computer is required for this class, either Windows or Mac, which is capable of using modern development tools. Frequently, you will be required to bring your computer to class. Students may use their personal Android or iOS device for testing or may check out an Android device for use during the course. There are a limited number of devices available so please contact me early if you need to borrow one.

Software Requirements

We will be using React Native in this course to build, test and deploy native mobile applications. Expo Go will be used to develop and debug applications with dependencies on node.js and does not require installing Xcode or Android Studio. Visual Studio Code is the recommended IDE for the course, however students may use any IDE or text editor they prefer.

Course Policies

Attendance Policy

This course is designed with weekly activities, quizzes, labs, and other assignments designed to build a scaffold and improve your understanding over time. The best way to learn the course material is to prepare for each class and actively participate in activities, and attendance at each session is expected. The University Attendance Policies mandate student attendance at lectures and provides accommodations for when you must miss class. While no separate attendance grade is recorded, attendance is indirectly measured through in-class lecture quizzes.

Grading System

Assignment Type	Percent of Final Grade
Projects (4-5)	70%
Lab exercises	10%
Lecture quizzes and homework	10%
Writing prompts - course content and student reflections	10%

Final course grades will be based on this grading scale

Percentage	Letter Grade	Percentage	Letter Grade	Percentage	Letter Grade
		100-93%	A	92-90%	A-
89-87%	B+	86-83%	B	82-80%	B-
79-77%	C+	76-73%	C	72-70%	C-
69-67%	D+	66-60%	D	59-0%	F

Lecture Quizzes

In-class quizzes will be given frequently covering lecture material and assignment content. Pre-lecture quizzes will also be given as an aid to student learning. Missed lecture quizzes cannot be excused or made up.

Homework

Short homework assignments will be given periodically to augment lecture content and will be due by the following class period. Their purpose is to cement knowledge acquired during the lecture. Late homework assignments will not be accepted.

Writing Prompts

Writing prompts will be assigned to encourage reflection on the course content, issues raised by mobile apps, and to reflect on students' goals and progress toward those goals.

Lab Exercises

Lab exercises will be assigned on Wednesday following class and due on Friday by midnight. These exercises can be completed on Friday when there is a scheduled lab or independently any time prior to the deadline. We will often, but not always, use the Friday class period as a lab section and students will be expected to bring a laptop with the class development tools installed when a lab is scheduled.

Projects

Projects are the primary means by which students demonstrate mastery of learning outcomes for this course. Students are expected to produce 4-5 working mobile applications throughout the semester that reflect their grasp of course contents. Code is expected to be high quality, reflecting principles covered in the course and specified in the project rubric. Clear expectations for projects will be outlined when they are assigned.

Late Assignment Policy

Late projects will lose 10% for each portion of a day they are late, counting from the exact time that the assignment is due. The maximum penalty applied to late assignments is 40% provided they are turned in within 2 weeks of the original due date, except for the final project which has no maximum penalty.

Late lab exercises will lose 10% for each portion of a day they are late, counting from the exact time that the assignment is due. Lab assignments not turned in within 4 days will be receive no points.

Other assignments such as quizzes, homework, and writing prompts will not be accepted late.

Exams

There are no formal exams in this course. The learning outcomes achieved by students will be assessed primarily through project work and shorter assessments such as quizzes, homework, and writing prompts. This reflects the focus of the course which is to build mastery in developing mobile applications in an industry setting.

Course Schedule

This is a rough outline of the order that course material will be presented. A tentative schedule will be provided separately and revised during the semester to keep pace with student progress.

1. Introduction to React Native, JavaScript & TypeScript, course tools, and services (e.g. GitHub)
2. React Native and TypeScript basics
3. Building a UI from a wireframe mock-up
4. Working with APIs in React
5. Mobile application development lifecycle
6. Automated unit and integration testing; CI/CD; GitHub Actions
7. Advanced topics in React Native
8. User credentials and authentication with an API
9. Tracking and analyzing user behavior trends
10. Robust and performant API integration
11. Debugging mobile applications and measuring and improving performance
12. Building a responsive mobile application with support for different devices and screen resolutions
13. Accessibility and privacy in mobile applications

University Policies

Disability Services Statement

If you need course adaptations or accommodations because of a disability, please contact me as soon as possible. The Office of Disability Services coordinates services for students with disabilities; documentation of a disability needs to be on file in that office before any accommodations can be provided. Disability Services can be contacted at 765-285-5293 or dsd@bsu.edu.

Importance of Diversity and Inclusion

Ball State University aspires to be a university that attracts and retains a diverse faculty, staff, and student body. We are committed to ensuring that all members of the community are welcome, through valuing the various experiences and worldviews represented at Ball State and among those we serve. We promote a culture of respect and civil discourse as expressed in our Beneficence Pledge and through university resources found at the Multicultural Center. For Bias Incident Response information, please visit https://www.bsu.edu/campuslife/multicultural-center/bias-incident-reporting or e-mail reportbias@bsu.edu.

The Learning Center

The Learning Center offers online and in-person appointment-based tutoring, drop-in tutoring at satellite locations, supplemental instruction, and testing accommodations for students with disabilities. Contact them for more details.

AI Policy

As you work on and submit assignments this semester, know that using generative AI tools to complete your assignments is a violation of Ball State's Academic Ethics Policy unless the AI use is explicitly approved by the faculty member. In general, students may not submit any work generated by an AI program as their own.

AI-assisted programming tools are gaining in popularity and usefulness and will be part of software development work in the future. However, use of these tools in this course will keep students from mastering the learning outcomes and impede their educational goals.

Ethics Policy

Honesty, trust, and personal responsibility are fundamental attributes of the university community. Academic dishonesty and other forms of academic misconduct threaten the foundation of an institution dedicated to the pursuit of knowledge and will not be tolerated. To maintain its credibility and reputation, and to equitably assign evaluations of scholastic and creative performance, Ball State University is committed to maintaining a climate that upholds and values the highest standards of academic integrity. View the complete Student Academic Ethics Policy at the Academic Integrity website.

Plagiarism Policy

Plagiarism is not acceptable. You are probably familiar with how plagiarism applies to written work, such as essays. For a programming class, plagiarism is similar – your "essay" is your code. There are many different ways to successfully solve each programming problem, and so code that is independently written is usually different than other work submitted by your classmates. You can discuss with each other in general terms how to approach solving a problem, but you should never share specific programming code, either verbally or in written form. Unless otherwise indicated for all assignments, you must work independently by yourself. Using online resources to diagnose and solve problems is acceptable but you should adapt any code found online to your use case and understand how and why it works. **Never email or directly share your code with other students, post it online, or otherwise disseminate solutions to assignments.** There are often questions about how plagiarism translates to programming, so we will discuss this in class as well. Please ask questions if you are uncertain about what is acceptable and what is not.

Subject to Change Statement

This syllabus and schedule are subject to change in the event of extenuating circumstances. If such a change is needed, I will notify you as early as possible through Canvas announcements and/or email.