Georgia Programming, Games, Apps, and Society Syllabus

High School (125 Contact Hours)

Course Overview and Goals

Mobile applications play an increasingly important role in how we consume media, access news, interact socially, and learn. In this *Georgia Programming, Games, Apps, and Society* course, students will learn how to design and build mobile applications using React Native, a popular platform-agnostic framework developed by Facebook and used by leading technology companies such as Airbnb, Instagram, Tesla, and more.

Students will explore the foundations of the React Native framework, including components and how they can be combined to create scalable, customizable, and high-performance mobile applications. Along the way, students will develop essential computer science skills such as managing state changes, working with stylesheet objects, mapping through data, rendering dynamic content, and creating modular app layouts using Flexbox and the Dimensions API.

As an online blended high school course, students will apply these skills by designing and building applications that run on their own smartphones, using modern tools and industry-relevant technologies for mobile app development.

Learning Environment

The course utilizes a blended classroom approach that combines web-based instruction with hands-on, in-person activities. Students will modify existing code and run it in the browser and on their personal mobile devices using Expo Go, a free, open-source toolchain for running React Native applications with JavaScript. Students will also create personalized apps, develop digital presentations, and participate in collaborative, in-person activities with classmates. Teachers leverage the tools and resources provided by CodeHS to maximize instructional time, facilitate meaningful collaboration, and provide focused one-on-one support for students.

Programming Environment

Students modify and run programs in the browser using the CodeHS online editor. Students will be able to modify text-based programs in JavaScript using the React Native framework. Students will run their apps directly on the CodeHS platform as well as their own personal mobile devices with Expo Go.

More Information

Browse the content of this course at https://codehs.com/course/27448/overview

Prerequisites

The Georgia Programming, Games, Apps, and Society course is designed for students with prior experience in computer science and web development.

Students should have the following background knowledge:

- **JavaScript**, such as the JavaScript taught in *Computer Science Principles*. React Native is a professional-grade framework that requires an understanding of core JavaScript concepts, including variables, objects, mapping through data, and basic program logic.

- **HTML and CSS**, such as the web design content taught in *Georgia Introduction to Software Technology*. Familiarity with HTML and CSS helps students understand component structure, layout, and styling, as React Native relies on tag-based components and uses a StyleSheet system similar to CSS.

Students will also be expected to read technical documentation and use external resources to support debugging, validation, and continued learning throughout the course.

Technology Requirements

To complete all activities and exercises in this course, students must have access to the 3rd party sites and tools listed here: Course Whitelist

Course Breakdown

Module 1: Introduction to Mobile Apps (0.5 weeks/2 hours)

In this module, students will be introduced to the mobile apps course and the React Native framework and it's program structure and syntax. Students will also preview some of the tools and technologies they will use to build and run their apps.

Topics Covered	● Introduction to React Native and Expo
Example Assignments	● Introduction to React Native and Expo ○ Why build mobile apps? ○ What is React Native? ○ How is React Native structured and built? ○ What is Expo and how do React Native apps run? ○ Example activity: ■ Running Apps on Expo

Module 2: Components and the Stylesheet (2 weeks/10 hours)

In this module, students are introduced to components, the backbone of building apps with React Native. They learn about the stylesheet object and how to add custom style attributes to their apps.

Topics Covered	● Introduction to Components ● Introduction to the Stylesheet API ● Styling View Components ● Styling Text Components
Example Assignments	● Introduction to Components ○ What are components? ○ How are components imported? ○ Where are components stored in the CodeHS editor? ○ How do you debug components? ○ Example activity:

	■ Customizing Text ■ Debugging Components ● Introduction to the Stylesheet API ○ What is the Stylesheet API? ○ How do you style apps? ○ How do you connect components to the Stylesheet API? ○ Example activity: ■ Connecting Components to the Stylesheet ● Styling View Components ○ How do you style View components? ○ How are components justified? ○ How are components aligned? ○ What is flex direction? ○ Example activity: ■ Nested Views ● Styling Text Components ○ How are Text components styled? ○ What are the common styling attributes for Text components? ○ Example activity: ■ News Article Styling ■ My Favorites List

Module 3: Buttons and Text Boxes (2 weeks/10 hours)

In this module, students use the TouchableHighlight component to create interactive “buttons” that add functionality to their apps. They also learn how to use the TextInput component to allow users to enter and submit text. Additionally, students examine intellectual property and key cybersecurity issues related to app development.

Topics Covered	● TouchableHighlight and the onPress Function ● TextInput ● Protecting Data and Ideas ● Risks and Responsibilities
Example Assignments	● TouchableHighlight and the onPress Function ○ How do you add a TouchableHighlight component? ○ How is the TouchableHighlight component used as a button? ○ What is the onPress function? ○ Example activity: ■ Build a Homepage ● TextInput ○ What is the TextInput component? ○ Example activity: ■ Google Search Page ■ Feedback Form ● Protecting Data and Ideas ○ How is your data used? ○ What are the pros and cons of intellectual property laws? ○ Example activity: ■ Take a Stand: Location Tracking ● Risks and Responsibilities ○ How can we secure our apps? ○ How can a data breach affect app users?

	○ Example activity: ■ AAA Security Framework ■ The Tea App Breach

Module 4: Advanced Layouts and Images (3 weeks/15 hours)

In this module, students learn how to add images to their apps to further customize the building experience. They gain a deeper understanding of mobile app layouts using Flex values and the Dimensions API.

Topics Covered	● The Image Component ● The ImageBackground Component ● Flex Layouts ● Dimensions API
Example Assignments	● The Image Component ○ What is the Image component? ○ How are image components sized? ○ What is the ImageBackground component? ○ How are ImageBackground components sized? ○ Example activity: ■ Google Search Page with Images ■ Phone Background ● Flex Layouts ○ What are flex layouts? ○ How do you calculate flex layout ratios? ○ Example activity: ■ Simple Flag with Flex ■ Advanced Flag with Flex ● Dimensions API ○ What is the Dimensions API? ○ How can I set the size of components, images, and text using screen height and screen width variables? ○ Example activity: ■ SnapChat Login Page using Dimensions

Module 5: App Build: Recreate a Popular App (1 week/5 hours)

In this module, students apply their knowledge of components, images, buttons, text boxes, and layouts to recreate a screen of a popular app. They will go through this process in steps, taking time to plan their layout and think about the functionality of the app. They may return to the app later to add on additional functions as new concepts are learned.

Topics Covered	● App Build: Recreate a Popular App
Example Assignments	● App Build: Recreate a Popular App ○ How do you plan for an app build? ○ What functionality does the app have? ○ How will you use your knowledge of flex values and the Dimensions API to make your app responsive?

Module 6: Events and State (2.5 weeks/12 hours)

In this module, students learn how to use state values and how to update the state of their app in various ways to create quick, dynamic programs.

Topics Covered	● Introduction to State ● Updating State with onPress ● Using Mathematical Equations to Update State ● Modifying Images and Audio Using State ● Using Methods to Update String States
Example Assignments	● Introduction to State ○ What is state? ○ What is the state object? ○ How is the state added using JSX? ○ Example activity: ■ Setting the Location with State ● Updating State with onPress ○ How is the state of an app updated? ○ How do we call functions to update state? ○ Example Activity: ■ Name to Nickname ● Using Mathematical Equations to Update State ○ How can mathematical equations be used to update state values? ○ Example activity: ■ Easy Calculator ● Modifying Images and Audio Using State ○ How can we allow the user to modify images and audio included in our apps? ○ Example activity: ■ Fading an Image Using Opacity ● Using Methods to Update String States ○ How can we use string methods to update the state of a string? ○ Example activity: ■ Spellchecker

Module 7: Creating Multiple Screens (1.5 weeks/7 hours)

In this module, students learn how to take their apps to the next level by adding in functionality that allows for content and interactivity on multiple screens.

Topics Covered	● Creating a Navbar ● Using Buttons to Navigate Screens
Example Assignments	● Creating a Navbar ○ What is a navbar? ○ How do I create a navbar? ○ How can I use images as buttons in my navbar? ○ Example activity: ■ iPhone Call Navbar ● Using Buttons to Navigate Screens ○ How can state values be used to create multiple screens? ○ How can I use buttons in my navbar to navigate to different screens? ○ Example activity: ■ RSVP Tracker

Module 8: Working with Conditionals (2 weeks/10 hours)

In this module, students use conditionals to allow for more flexible apps. They also learn how parameters are used inside functions.

Topics Covered	● Conditionals and State Changes ● Conditionals using Parameters
Example Assignments	● Conditional and State Change ○ What are conditionals? ○ Where are conditionals used? ○ How can the state be updated conditionally? ○ Example activity: ■ Venmo Balance Transfer: Check Balance ● Conditionals using Parameters ○ How are parameters passed in React Native? ○ Example activity: ■ Conditional Winner

Module 9: App Build: Currency Converter App (1 week/5 hours)

In this module, students apply their knowledge of components, events, state, conditionals, and more to build a currency converter app. Students use the Stylesheet to create a user friendly layout that can convert multiple currencies!

Topics Covered	● Project: Currency Converter App
Example Assignments	● Project: Currency Converter App ○ How do you plan for an app build? ○ What is the best layout for an app? ○ How do you pseudocode an app? ○ What makes an app presentation engaging?

Module 10: Using Collections of Data (1.5 weeks/7 hours)

Students learn how to use arrays to organize and use larger data sets in their programs. They'll learn how to use mapping to quickly assign styling and layouts to large amounts of data.

Topics Covered	● Using Arrays and Indexing to Set State ● Mapping Through Objects in Arrays
Example Assignments	● Using Arrays and Indexing to Set State ○ What are Arrays? ○ How can Arrays be used to store data? ○ How can I use indexing to set state? ● Mapping Through Objects in Arrays ○ What is the map function? ○ How can I map through my data sets?

Module 11: Working with ScrollView (1.5 weeks/7 hours)

Students learn how to use scrollview inside their programs to show more content on one page than can fit on the screen. Students learn how to scroll through images and text and how mapping can be used with scrollview to move through large sets of data.

Topics Covered	● Using ScrollView ● ScrollView using Objects

Example Assignments	● Using ScrollView ○ What is ScrollView? ○ Using ScrollView with text and images ● ScrollView using Objects ○ How can I use ScrollView with the map function? ○ Example activity: ■ Mapping my Class Schedule

Module 12: App Build: Image Feed App (2 weeks/10 hours)

In this module, students apply concepts from previous modules to build an image feed app. Students use control structures, state, and object mapping to display a number of photos, comments, and other information, with the ability to "like" photos!

Topics Covered	● Project: Image Feed App
Example Assignments	● Project: Image Feed App ○ How do you plan for an app build? ○ What is the best layout for an app? ○ How do you pseudocode an app? ○ What makes an app presentation engaging?

Module 13: Designing User Interfaces (3 weeks/15 hours)

This unit introduces students to the theory and practice of user interface design. Students learn about what makes an engaging and accessible user interface, and will employ an iterative design process, including rapid prototyping and user testing, to design and develop their own engaging web pages.

Topics Covered	● What makes an engaging interface? ● Various User Interface (UI) Design techniques ● Accessibility issues ● Readability ● Lite sites ● Rapid prototyping ● User testing
Example Assignments	● Research existing user interfaces ● Assess the user interfaces of various web sites ● Design a website using paper prototypes, test these prototypes and get feedback from your peers, and improve your design before implementing it with code ● UI Design Project ○ Find and present an article about a particular UI design technique ○ Create your own live examples using this technique

Module 14: Computer Science Careers (1 week/5 hours)

Students take some time to explore and discover different computer science careers. They will also examine inclusive coding and how to avoid bias in computer programming.

Objectives / Topics Covered	● Computer Science Careers ● Computer Science used in non-CS Careers ● Inclusive Coding
Example Assignments	● Career Research ● Computer Science concepts that are used in non-computer science positions ● How can a computer program's bias become dangerous? ● Why is it important to have a more diverse group of people in the computer science field?

Module 15: Final (0.5 weeks/2-3 hours)

Topics Covered	● Students will be tested on all topics included in the course ● Multiple choice, fill-in-the-blank, short answer, and coding questions included
Example Assignments	● Part 1- Multiple Choice: Can be taken online or on paper (Paper version found in resources ● Parts 2, 3 & 4- Fill-in-the-blank, short answer, coding: Paper versions available in resources

Supplemental Modules

Supplementary Units	**Prerequisite/Recommended Unit(s)**
Mobile Apps Prerequisite - HTML - CSS - Variables - Conditionals	Can be used to review JS, HTML, and CSS, or to fill gaps for students before beginning course content.
Challenges	Each item notes the recommended placement in the course.
Additional Topics - Changing Attributes through User Interaction	First, complete all units in the main course.
End of Year App Build	In this project-facing module, students will use everything they have learned in the course and design, prototype, and code their own custom app and present it to their class.
Midterm	This exam covers the following modules: Components and The Stylesheet, Buttons & Text Boxes, Advanced Layouts & Images, Events & State, Creating Multiple Screens, and Working with Conditionals. Parts 2, 3 & 4- Fill-in-the-blank, short answer, coding: Paper versions available in resources