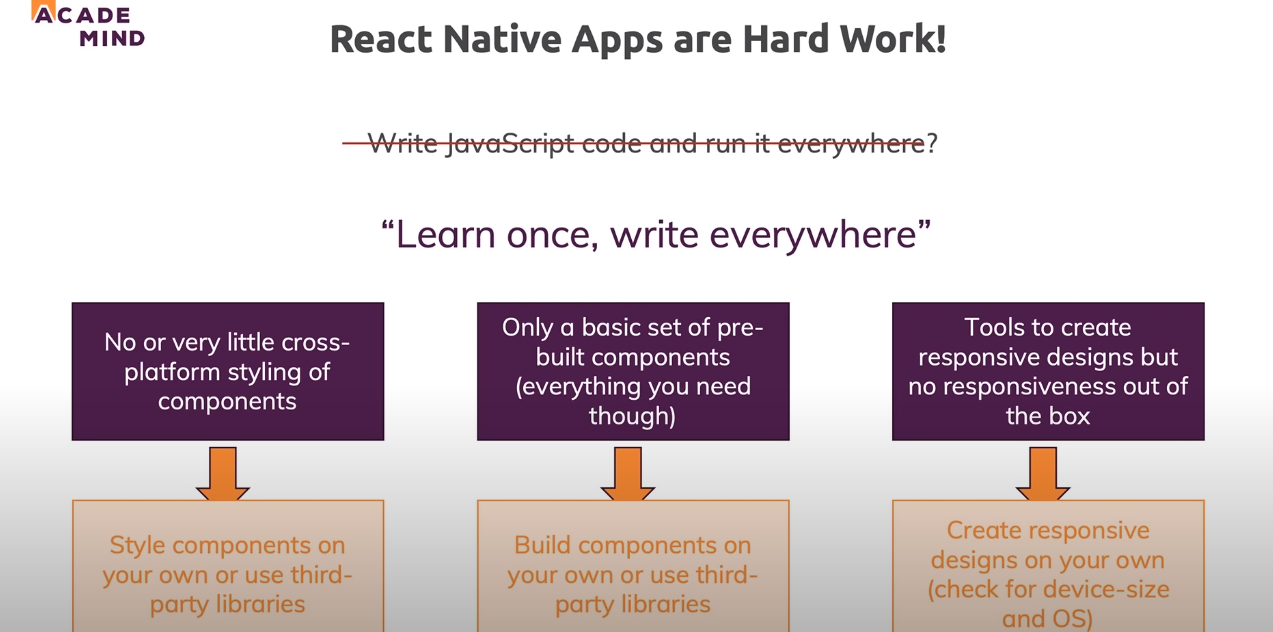
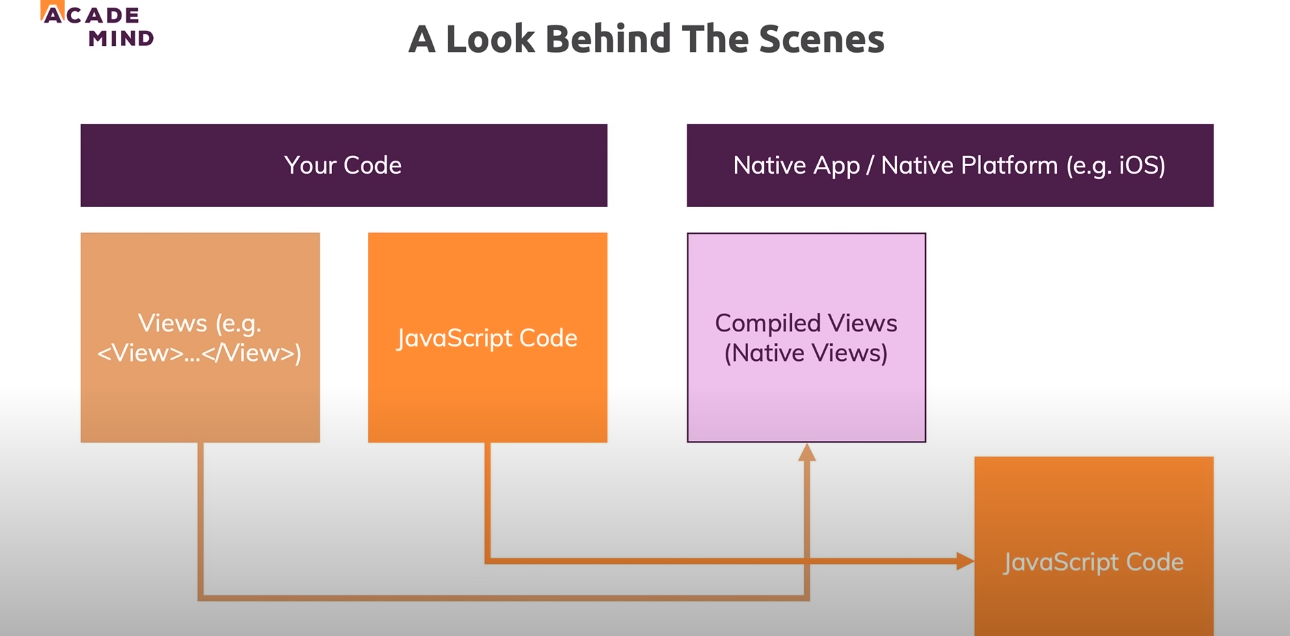
Period-3 An introduction to Cross-Platform App Development with React Native

Note: This description is too big for a single exam-question. It will be divided up into separate questions for the exam

Explain Pros & Cons with React Native + Expo used to implement a Mobile App for Android and IOS, compared to using the Native Tools/languages for the two platforms.  
  
Procs:  
- Cheaper in the sense that you “only” need program language.  
- React Native is very close to React/JavaScript and CSS in some ways.  
- You can create a layout and design for both iOS and Android at the same time. So again you don’t need to create 2 separate apps. Therefore it is cheaper and faster.  
- In theory you should be able to spin something up fast that another user quickly can test/use.  
  
Cons:  
- It is a little bit slower than using their own platform.  
- Personally expo+react native seems even though they have been around for years. Like a brand new company. The amount of errors/issues I’ve had is absurd.

What is meant by the React Native Paradigm "Learn once, write anywhere" compared to for example the original (now dead) idea with Java "Write Once, run everywhere".  
- They build on tools that we have already learned with extra features/little changes in the “language”. So if someone already with React / JavaScript experience they should be able to get started quickly. I would guess that the comparison to the Java one, it also has a very smaller scope. They are mostly marketed for mobile/web. Where Java is almost anything in the world.  
  
*With React Native, the new paradigm is “learn once, write anywhere.” With this approach, an experienced web React developer can get up and running, and write Android or iOS apps at a much faster pace.*

*Currently, there is a little to no code sharing possible between React apps on the web, Android, and iOS React apps. With time, Facebook hopes to allow code sharing between target platforms. To understand how this will be facilitated, we can look at how React for web was broken into two packages: React and React DOM.*  
<https://thenewstack.io/react-native-learn-write-anywhere/#:~:text=With%20React%20Native%2C%20the%20new,Android%2C%20and%20iOS%20React%20apps>.   
  
  
*But coming back to the tagline, “write once, run everywhere” things have changed since 1995. TLC’s not hot anymore and same goes for Java. Why? First of all, Java developers are expensive. They are harder to find, they are usually good and coming from well established companies, thus demanding lots of money.*  
<https://medium.com/@Maephisto/javascript-is-the-new-java-d14f0585d05e>   
  
<https://youtu.be/qSRrxpdMpVc?t=2434>

In React Native, which parts of your code gets compiled to Native Code (Widgets) and which parts do NOT?  
  
  
<https://youtu.be/qSRrxpdMpVc?t=759>

<https://www.netguru.com/what-is-react-native>

Explain the basic building block in a React Native Application and the difference(s) between a React Application and a React Native App.  
- React Application  
A library that supports both frontend and backend.  
Can be used for both mobile apps and websites.  
  
  
- React Native Application  
No HTML elements/tags.  
CSS is non existing. But the styles is on some levels the same. There are very different features though.  
Cross platform mobile framework. We can get things to work on several mobile systems.  
Uses native components instead of web components.  
  
<https://www.cognitiveclouds.com/insights/what-is-the-difference-between-react-js-and-react-native/#:~:text=React%20is%20a%20framework%20for,a%20high%20performing%20UI%20layer.&text=React%20Native%20doesn't%20use%20CSS%20either>.   
<https://medium.com/@alexmngn/from-reactjs-to-react-native-what-are-the-main-differences-between-both-d6e8e88ebf24>   
<https://www.simform.com/reactjs-vs-reactnative/>

Explain and demonstrate ways to handle User Input in a React Native Application  
- See Period 3/Day2/geoDemoNoDB-Client-startcode1

Explain and demonstrate how to handle state in a React Native Application  
- See Period 3/Day2/geoDemoNoDB-Client-startcode1

Explain and demonstrate how to communicate with external servers, in a React Native Application  
- See Period 3/Day2/geoDemoNoDB-Client-startcode1  
<https://reactjs.org/docs/hooks-state.html>

Explain and Demonstrate ways to debug a React Native Application  
- Well with our backend we can use debug. Else we can jsonify data and show them as an alert to see what data we’re working with. We can also use console.log()  
TO WATCH: <https://youtu.be/qSRrxpdMpVc?t=11414> (React Native Tutorial for Beginners - Crash Course 2020 - Debugging)

Explain and demonstrate how to use Native Device Features in a React Native/Expo app.  
- See Period 3/Day2/geoDemoNoDB-Client-startcode1   
We used react-native-maps, expo-location and more.  
Should also be some stuff in \Period 3\Day1\react-native-day1

<https://medium.com/@rocknegi53/comprehensive-guide-for-react-native-device-features-maps-gallery-and-camera-8d59d608a258>

Explain and demonstrate a React Native Client that uses geo-components (Location, MapView, etc.)  
- Frontend:   
<https://expo.io/@bringordie/projects/client-location-noDB>   
<https://github.com/Bringordie/SEM4_FullStack/tree/master/Period%203/Day2/geoDemoNoDB-Client-startcode1>   
- Backend:  
<https://expressdemos.cphfb.codes>   
<https://github.com/Bringordie/SEM4_FullStack/tree/master/Period%202/week40-44/in-memory>

Demonstrate both server and client-side, of the geo-related parts of your implementation of the ongoing semester case.  
- See above