**3701ICT/7421ICT Mobile Application Development**

**MAD**

1. Exercise 2 (16 pts, 2pt each question)

Q1. What's one of the main roles of the built-in <View> component?

* 1. ~~Directly output text.~~
  2. **Structure/ Group other child components.**
  3. ~~Handle touch events.~~

Q2. Can you use HTML elements (e.g. <div>, <p>, <input>) in React Native apps?

1. ~~Yes, React Native is able to handle HTML components, but native components are preferred.~~
2. ~~No, React Native doesn’t provide any built-in components, you need to build all components from scratch.~~
3. **No, React Native doesn’t recognize these components – it doesn’t know how to compile them to native views.**

Q3, What’s the relation between React Native component styling and CSS (Cascading Style Sheets) for the Web?

1. **React Native styling is inspired by CSS (comparable/ similar property names and values)**
2. ~~React Native uses the exact same syntax but only supports a limited set of features.~~
3. ~~There is no connection.~~

Q4, Which of the following example style rules does NOT work in React Native?

1. ~~borderColor: ‘black’~~
2. ~~padding: 10~~
3. **‘background-color’: ‘#ccc’**

Q5, Why would you use const styles = StyleSheet.create({}) instead of a regular JavaScript object (const styles = {})?

1. ~~There is no reason to use StyleSheet.~~
2. **Using a StyleSheet adds validation and potential performance improvements.**
3. ~~Using a StyleSheet unlocks extra styling properties.~~

Q6, What is “Flexbox”?

1. ~~A styling property that allows you to order items in a grid.~~
2. ~~A component built into React Native which you can use to structure content.~~
3. **A concept/ set of styling properties that allows you to structure content (i.e. create layouts).**

Q7, What's the default styling/ layout behavior of a <View> component?

1. ~~It organizes child components in a grid.~~
2. ~~It doesn’t allow any ordering or structuring of child components.~~
3. **It uses Flexbox to organize its child components.**

Q8, If a <View> has flexDirection: 'column' (which is the default) - what does alignItems: 'flex-end' do in that case?

1. **It positions all child elements at the end of the column – on the horizontal axis.**
2. ~~It positions all child elements at the end of the column – on the vertical axis.~~
3. ~~It has no effect.~~