Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

Engineering of Mobile Systems

Final Exam

September 7th, 2022

FIRST NAME		**LAST NAME**	
STUDENT NUMBER		**SIGNATURE**	

Instructions for students:

Write First Name, Last Name, Student Number and Signature where indicated. If not, the examination can not be marked.

Do not speak to any other student during the examination. If you speak to another student, your examination will be cancelled.

Use a pen, not a pencil.

Write neatly and clearly.

Part 1: JavaScript, TypeScript, Functional Programming (12 points)

Question 1: TypeScript

Add all the type annotations necessary for the following piece of JavaScript code convert it to a TypeScript piece of code that type checks without errors (do not use the "any" type). Copy or reference line numbers as necessary, and don't forget that annotations may be necessary outside of the piece of code itself:

```
1.  const area = (shape) => {
2.     switch (shape.kind) {
3.         case "Circle": return Math.PI * (shape.radius ** 2)
4.         case "Square": return shape.side ** 2
5.         case "Rectangle": return shape.width * shape.height
6.     }
7.  }
8.
9.  const larger = (shape, factor) => {
10.    switch (shape.kind) {
11.        case "Circle": return {...shape, radius: shape.radius * factor}
12.        case "Square": return {...shape, side: shape.side * factor}
13.         case "Rectangle": return {...shape,
14.                                   width: shape.width * factor,
15.                                   height: shape.height * factor}
16.    }
17. }
```

Question 2: Code Critique

Critique the following piece of (JavaScript) code, and explain how you would improve it. It can be in words only, not necessarily with source code. Reference line numbers as necessary. One obvious improvement would be to add TypeScript annotations, you don't need to mention it.

```
1.  function playWordle(input){
2.    var specialCommand = false;
3.    while(initialConfig.stats.tries > 0){
4.      for (var i = 0; i < initialConfig.specialWords.length; i++) {
5.        if (initialConfig.specialWords[i] === input) {
6.          specialCommand = true
7.        }
8.      }
9.      if(specialCommand){
10.       console.log('called a special command')
11.       additionalFeatures(input);
```

```
    } else {
      initialConfig.stats.tries = initialConfig.stats.tries - 1;
      playAGuess(input)
    }
    specialCommand = false
    input = prompt("enter a command")
  }
  console.log('The game is over!');
}
```

Question 3: Functional Programing

Rewrite the following imperative piece of code using Functional Programming principles, including higher-order functions. It processes a list of point. Points have x and y coordinates, both numbers.

```
const processData = (data, maximum) => {
  for (var i = 0; i < data.length; i ++ ) {
    data[i].x = data[i].x - 5
  }
  var result = []
  for (var j = 0; j < data.length; j ++ ) {
    if (data[j].y > 0) {
      result.push(data[j])
    }
  }
  for (var k = 0; k < result.length; k++) {
    var distance = Math.sqrt(result[k].x * result[k].x + result[k].y * result[k].y)
    if (distance > maximum) {
      console.log("the condition is met!")
      return true
    }
  }
  console.log("the condition is NOT met!")
  Return false
}
```

Question 4: Promises

Explain what a Promise is, and explain how they work (hint: mention the states of a promise). Provide a short example of usage of a promise, illustrating what it is used for.

Part 2: React Native (12 points)

Question 5: React Native Components

The following component has 2 bugs and 1 additional issue. What are they? List and explain the problems. Feel free to mention line numbers while you explain your answers.

```
1.  const MyComponent = ({header1, header2, message}) => {
2.    const [toggle, setToggle] = useState(false)
3.    const myAlert = () => {
4.      alert("the message is: " + message)
5.    }
6.
7.    return (
8.      <View style={{flexDirection: "row", margin: 10, padding: 10}} >
9.          <Text style={{fontSize: 24}}>{header1}</Text>
10.         <Button title="message" onPress={myAlert} />
11.     </View>
12.     <View style={{flexDirection: "row",  margin: 10, padding: 10}} >
13.         {toggle?<Text style={{fontSize: 24}}>{header2}</Text>:null}
14.         <Button title="toggle" onPress={setToggle} />
15.     </View>
16.   )
17. }
```

Question 6: React vs Native

Explain the pros and cons of using a cross-platform solution such as React Native, versus a more traditional mobile application framework.

Question 7: Component Rendering

Explain in detail what is component rendering, and what can cause a component to re-render.

Question 8: The Rules of Hooks

What are React's "rules of hooks"? List the rules and explain them clearly.

Part 3: General Mobile Programming concepts (6 points)

Question 9: Permissions

Explain the concept of Permissions in the programming of a mobile system, and explain why they are needed. Feel free to use examples if necessary to support your arguments.

Question 10: Persistence

Explain what persistence is for a mobile application, and why it is essential for a mobile application to provide such a degree of persistence.