

CS385 Lecture 12 – Project update

**Thanks to everyone who has
indicated their group formation
for the CS385 project**



- **IDEA – you should have an idea of what your application is going to be**
- **Some wireframe diagrams, some flowcharts, some drawings..... something to indicate the overall architecture of your application.**
- **Is your application within the theme of “A sustainable environment”?**

Labs – until end of semester

- There are no more weekly assignments in the labs.
- Labs are provided for you to work on your project.
- **Your group/individual project MUST check in with one of the demonstrators or me EVERY week in the lab.** This is part of REAL WORLD project management

Next three labs.....

- **Lab – Friday 10th November 2023** – prep for Lab Exam 1
- **Lab – Friday 17th November 2023** – first full project lab – you **MUST** check in with one of the demonstrator teams.
- **Lab – Friday 24th November 2023** – second full project lab – you **MUST** check in with ME.
TARGET – some type of a working demo (however simple)

Can we create our own data?

- **ABSOLUTELY YES**
- This is CS385 and our emphasis is on the mobile application development side
- You can create fake data – you can put it onto GitHub to simulate an API.
- Don't spend lots and lots of time searching for “real” data – focus on your application!

CS385 Lecture 12 – Demo Lab Exam 1 – Answers to selected questions

**If you still haven't attempted
Demo Lab Exam 1 – look away
now**

Q6 (63% attempts correct)

Based on the code presented below, what is rendered to the screen after the "Press me!" button has been clicked or pressed FOUR times?

```
1  import React, { useState } from "react";
2  // question code Q3A
3  function App() {
4    const [anon, setAnon] = useState(10);
5
6    function changeState() {
7      setAnon(anon + 3);
8    }
9    return (
10     <>
11       {anon > 15 && <Sunday />}
12       {anon % 2 === 0 && <Saturday />}
13       <button onClick={changeState}>Press Me!</button>
14     </>
15   );
16 } // end App() function or component
17
18 function Sunday() {
19   return (<><h1>Sunday</h1></>);
20 }
21 function Saturday() {
22   return (<><h1>Saturday</h1></>);
23 }
```

Variable **anon** begins = 10

Each time the button is clicked we add 3 to **anon**.

1st time – **anon** = 13

2nd time – **anon** = 16

3rd time – **anon** = 19

4th time – **anon** = 22

Therefore, line 11 is TRUE

Line 12 is TRUE (**anon** is even)

So, **Sunday Saturday** is printed

- ☐ a. Sunday
- ☐ b. None of the options provided
- ☒ c. Sunday Saturday
- ☐ d. Saturday
- ☐ e. Saturday Sunday

Q7 (65% of attempts correct)

Given the code below. What is rendered after FOUR clicks or presses of the "Press Me!" button?

```
1  import React, { useState } from "react";
2  // question code Q4A
3  function App() {
4    const [anon, setAnon] = useState(10);
5    const [cs385, setCS385] = useState(10);
6
7    function changeState() {
8      setAnon(cs385 + 3);
9      setCS385(anon + 3);
10
11    }
12    return (
13      <>
14        {anon > 15 && <Sunday />}
15        {cs385 % 2 === 0 && <Saturday />}
16        {cs385 - anon === 0 && <h1>Everyday!</h1>}
17        <button onClick={changeState}>Press Me!</button>
18      </>
19    );
20  } // end App() function or component
21
22  function Sunday() {
23    return (<><h1>Sunday</h1></>);
24  }
25  function Saturday() {
26    return (<><h1>Saturday</h1></>);
27  }
```

The key here is the switching of the state variables in the set methods or functions

Start anon = 10, cs385 = 10

1st time anon = 13, cs385 = 13

2nd time anon = 16, cs385 = 16

3rd time anon = 19 cs385 = 19

4th time anon = 22 cs385 = 22

Line 14 – TRUE

Line 15 - TRUE

Line 16 - TRUE

Sunday Saturday Everyday

- ☐ a. Saturday Everyday
- ☐ b. Sunday Saturday
- ☐ c. Sunday Everyday
- ☐ d. Everyday
- ☒ e. Sunday Saturday Everyday

Q10 (65% attempts correct)

The code below demonstrates parent-child communication. What is the value of x rendered by the Parent component after the button 'CS385' has been pressed FIVE times. Please note - for readability, all of the code has been rearranged so that it can be viewed within a small browser window.

```
function App() {
  const [x, setX] = useState(-1);

  function doSomething() {
    setX(x - 1);
    setX(x + 1);
  }

  return (
    <>
      <h1>Parent x = {x}</h1>
      <Alison handle={doSomething} />
    </>
  );
}
```

```
function Alison(props) {
  return (
    <>
      <h1>Alison</h1>
      <Julia handle1={props.handle} />
    </>
  );
}

function Julia(props) {
  return (
    <>
      <h1>Julia</h1>
      <button onClick={props.handle1}>CS385</button>
    </>
  );
}
```

This type of question will not appear on the Lab Exams

- ☐ a. 3
- ☐ b. 9
- ☒ c. 4
- ☐ d. 5
- ☐ e. -5

PROBLEM!

The issue here is that the state updates are asynchronous in React, and when you call `setX(x - 1)` and then `setX(x + 1)` immediately after, **React doesn't guarantee that the state will be updated immediately.**

Unexpected behavior when making multiple state updates in quick succession.

Q10 (65% attempts correct)

The code below demonstrates parent-child communication. What is the value of x rendered by the Parent component after the button 'CS385' has been pressed FIVE times.

```
JS App.js x
1 import React, { useState } from "react";
2 // Question code Q7A
3
4 function App() {
5   const [x, setX] = useState(-1);
6
7   function doSomething() {
8     setX(x - 1);
9     setX(x + 1);
10  }
11
12  return (
13    <>
14      <h1>Parent x = {x}</h1>
15      <Alison handle={doSomething} />
16    </>
17  );
18 }
```

Browser Tests Terminal

https://rc2rj2.csb.app/

Parent x = 4

Alison

Julia

CS385

Console 0 Problems 0

Console was cleared

This type of question will not appear on the Lab Exams

- ☐ a. 3
- ☐ b. 9
- ☒ c. 4
- ☐ d. 5
- ☐ e. -5

PROBLEM!

The issue here is that the state updates are asynchronous in React, and when you call `setX(x - 1)` and then `setX(x + 1)` immediately after, **React doesn't guarantee that the state will be updated immediately.**

Unexpected behavior when making multiple state updates in quick succession.

Q11 (58% attempts correct)

What is rendered, from line 11, to the screen **after the final iteration of the map function**? Please select one of the options below.

```
1  import React from "react";
2
3  // Question code 8A
4  function App() {
5    let myData = [{ a: 42 }, { a: 52 }, { a: 82 }, { a: 92 }];
6
7    return (
8      <>
9        {myData.map((a, index) => (
10          <p key={index}>
11            {a.a}{index - 2}
12          </p>
13        ))}
14      </>
15    );
16  }
17  export default App;
```

The index will always start at 0

Printed IN ORDER

42-2 (index = 0)

52-1 (index = 1)

820 (index = 2)

921 (index = 3)

- ☐ a. 821
- ☐ b. 820
- ☐ c. 922
- ☒ d. 921
- ☐ e. 920

Lab Exam 1
Friday 10th November 2023
11:00 – 12:00

**Please read the Lab Exam Guidance
Slides (from Topic 5 and within Lab
Exam 1 on Moodle)**

Lab Exam 1 – mini guide

- **Be there on time! PLAN YOUR JOURNEY TO MU**
- **Please sit in your assigned ROW** (see Lab Exam 1 for seating chart)
- **OPEN BOOK** – but only Moodle allowed to be used as a website
- **ONE ATTEMPT – 60 MINUTES – 12 QUESTIONS**
- **Leave the room if you finish before 60 minutes**
- **You can use your own laptop** (if you wish)
- **Demonstrators available 10 – 11 to answer questions**

See you on Friday!