

Statement Purpose:

To familiarize the students with

- ▣ React Lists
- ▣ React Class
- ▣ React Conditionals
- ▣ React props

List items

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = (
  <ul>
    <li>Honda</li>
    <li>Toyota</li>
    <li>Mercedes</li>
  </ul>
);

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
export default myElement;
```

## Attribute Class

1. Create class in “App.css”

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './App.css';

const myElement = <h1 className="myclass">Hello World</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
export default myElement;
```

## Conditions

1. Sample code

```
import React from 'react';
import ReactDOM from 'react-dom/client';
```

```

const x = 5;
let text = "Goodbye";
if (x < 10) {
  text = "Congratulation! Victory";
}

const myElement = <h1>{text}</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
export default myElement;

```

## 2. Sample code 2:

Use ternary

```

import React from 'react';
import ReactDOM from 'react-dom/client';

const x = 5;

const myElement = <h1>{(x) < 10 ? "Congratulations!Victory" : "Goodbye"}</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
export default myElement;

```

## Exercise

1. Render a <p> element without using JSX.  
Hint
  - i. const paragraph = React.createElement(\_\_\_\_, {}, 'This is a paragraph without using JSX!');
  - ii. const paragraph = \_\_\_\_ This is a paragraph using JSX! \_\_\_\_
2. write a code to complete this statement that renders a JavaScript expression inside JSX.
  - a. const myelement = \_\_\_\_ React is \_\_\_\_ 10\*10\_\_\_\_ times better with JSX! \_\_\_\_;
3. write a code to complete this expression to include a class attribute the way JSX supports
  - a. const title = <h1 \_\_\_\_ = "primary">Hello World!</h1>;

## React Component

### 1. Create a Class component called Branch

```
class Branch extends React.Component {  
  render() {  
    return <h2>Bank Branches!</h2>;  
  }  
}
```

### 2. Create a Function component called Branch

```
function Branch() {  
  return <h2> Bank Branches </h2>;  
}
```

## Rendering Components

### Sample Code

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
  
function FYP() {  
  return <h2>Final Year Project</h2>;  
}  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<FYP />);  
  
export default FYP;
```

## Props

Components can be passed as props, which stands for properties.

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
  
function Branch(props) {  
  return <h2>This is a {props.color} Branch!</h2>;  
}  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Branch color="F7"/>);  
export default Branch;
```

### Component in a component

```
import React from 'react';
```

```

import ReactDOM from 'react-dom/client';

function FYP() {
  return <h2>Semester Project</h2>;
}

function Uni() {
  return (
    <>
      <h1>Introduction to Algorithms?</h1>
      <FYP />
    </>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Uni />);

export default FYP;

```

### Exercise:

1. Write a code to print person name using React component.
2. Print Hello, Ahmed by creating a variable named **name** and pass it to the **Message** component.