## **Objectives**

* Explain React components
* Identify the differences between components and JavaScript functions
* Identify the types of components
* Explain class component
* Explain function component
* Define component constructor
* Define render() function

Answers :

**Explain React components**

**- React components are reusable, self-contained pieces of UI.**

**- They return HTML (JSX) to be rendered on the screen.**

**function Welcome() {**

**return <h1>Hello, World!</h1>;**

**}**

**Differences between components and JavaScript functions**

**- JS functions do computations; React components return UI (JSX).**

**- React components must start with a capital letter.**

**- React components can use hooks (in functional components) or lifecycle methods (in class components).**

**Types of components**

**1. \*\*Function Component\*\***

**2. \*\*Class Component\*\***

**Explain class component**

**import React, { Component } from 'react';**

**class Welcome extends Component {**

**render() {**

**return <h1>Hello from Class Component</h1>;**

**}**

**}**

**- Uses ES6 class.**

**- Has lifecycle methods.**

**- Requires `render()` to return JSX.**

**Explain function component**

**function Welcome() {**

**return <h1>Hello from Function Component</h1>;**

**}**

**```**

**- Simpler and preferred for most use cases.**

**- Can use React Hooks.**

**Define component constructor**

**class Welcome extends React.Component {**

**constructor(props) {**

**super(props);**

**this.state = { message: "Hello" };**

**}**

**}**

**- Initializes state and binds methods.**

**- Called before `render()`.**

**Define render() function**

**render() {**

**return <h1>Hello</h1>;**

**}**

**- Mandatory in class components.**

**- Returns the JSX to be rendered.**

In this hands-on lab, you will learn how to:

* Create a function component
* Apply style to components
* Render a component

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **30 minutes.**

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

1. Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:



1. Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”
2. Type the following code in CalculateScore.js





1. Create a Folder named Stylesheets and add a file named “mystyle.css” in order to add some styles to the components:



1. Edit the App.js to invoke the CalculateScore functional component as follows:



1. In command Prompt, navigate into scorecalculatorapp and execute the code by typing the following command:



1. Open browser and type “localhost:3000” in the address bar:



**Output :**



