LU1 – styles in react

External stylesheets

Src>Styles.**css**

.primary{

    background-color: black;

    color:green

}

**Import style.css in app.jsx**

import './Styles.css'

    <h2 className='primary'>styles type 1 in react</h2>

Type2- inline styles

    <h2 style={{color:'red',backgroundColor:"black"}}>styles type 2 in react</h2>

Use camel case attributes like backgroundColor

{} – for jsx

Other is an object of attributes combination

**Lu2: hooks**

-limitations of javascript classes (class based components)

This keyword usage is complex in class based componetns

Component lifecycle methods must be used while creating components for complex schenarios like data fetching , subscribing the events

Data fetching done in componentdidmount() to make api calls,fetch and sometimes in componentdidupdate() based on scenario – so this confusion will be there.

For event listeners u set events in componentdidmount() and unsubscribe in componentwillunmount()

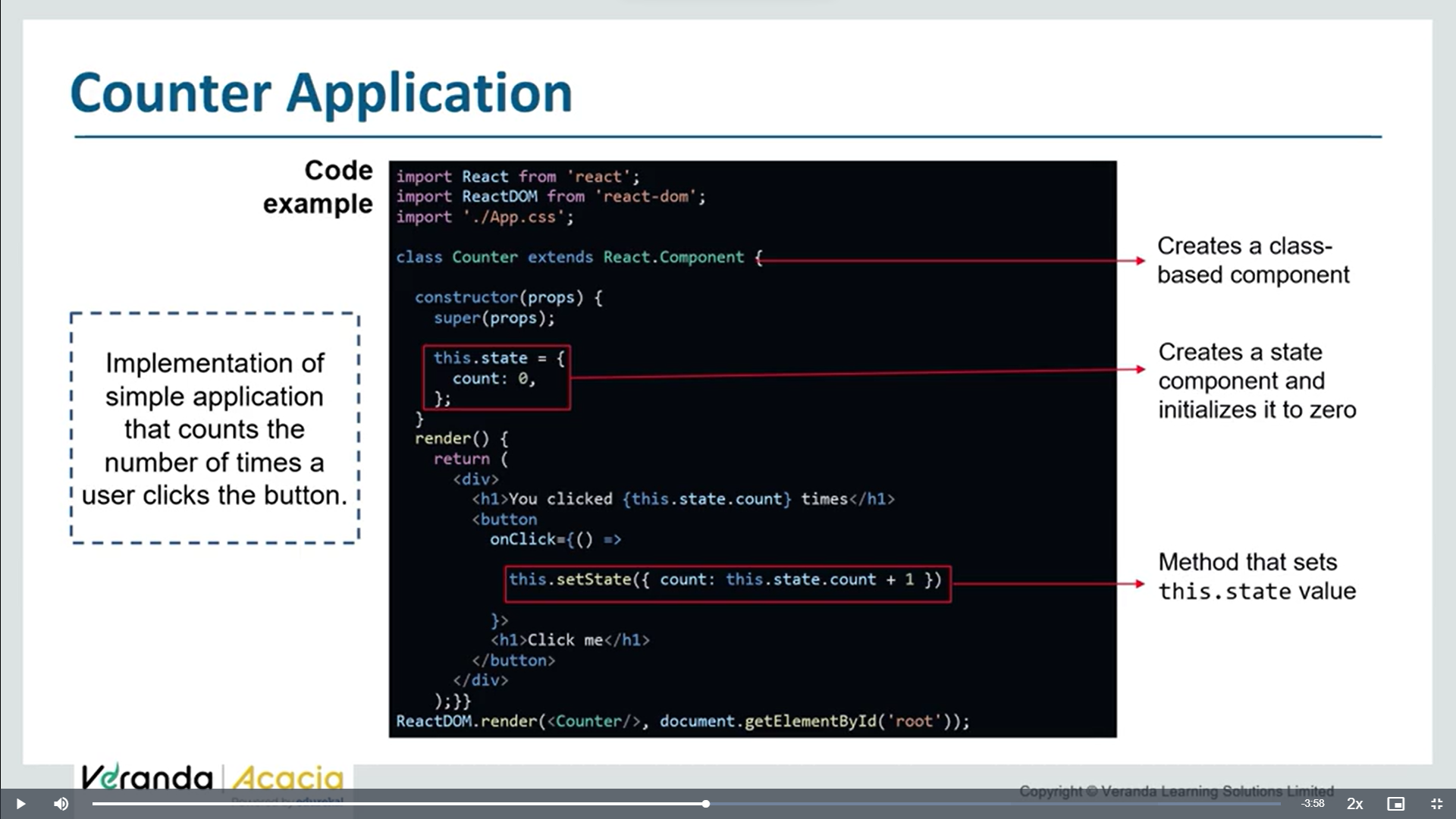
This leads to splitting of code as per component lifecyclemethods and not as per components functional use

Through use of multiple components may not matter interms of view structure but can cause wrapperhell

Wrapper hell refers to the the chaotic hierarchy of components that can happen in a complex React project. In React we learn about patterns such as “Render Props” and “Higher-Order Components”. Render props refers to the share of data between two components by using a prop that has a function as a value.24 Apr 2021

A simple application like counter app- even such simple app becomes complex in classbased component.

We initialize state in constructor



**Q. Is it possible to work with states, without javascript classes to simplify the application code?**

**Yes u can write functional components with react hooks.**

**Earlier drawbacks of functional components:**

1. **We cannot work with states in functional components**
2. **We cannot work with lifecycle methods in func comp**

**Now react came with new concept hooks**

**React hooks:**

React hooks are basically functions that let us include react state and lifecycle features without js classes.

They allow us to use features of statefull components in functional components

They mainly deal with state,effect,context in react app

We need to ensure that order of hooks always stays same

We cannot use hooks in loops /conditionals

**Functional components with react hooks:**

Func comps with hooks are simpler,as there is no need to define constructors,this keyword,lifecyclemethods

Hooks don’t split components as per lifecycle methods, rather split into smaller funcs based on related pieces

They organise logic inside a component into reusable isolated units

Easier to test and refractor, allow developers to write clear and concise code

Diy 77

Normal inline and external style