***Lists and Hooks***

**Explain Life cycle in Class Component and functional component with Hooks.**

**Class Components:**

* **Phase of Mounting:**  
  constructor(): This function is triggered during the creation of a component instance.  
  componentWillMount() is no longer supported and should not be used. carried out immediately prior to the component mounting to the DOM.   
    
  Render the component with render().   
  After the component is mounted to the DOM, componentDidMount() is called. Perfect for creating subscriptions or making API calls.
* **Update Stage:**  
  componentWillReceiveProps(nextProps): This function is deprecated and should not be used. called whenever a new set of props is given to the component.  
  ought toComponentUpdate(nextProps, nextState): This function decides whether or not to render the component again.   
    
  componentWillUpdate(nextProps, nextState): This function is deprecated and should not be used. called immediately before rendering in the event that new props or states arrive.   
  render(): Renders the element once more.   
  componentDidUpdate(prevProps, prevState): Called right away following an update. Excellent for responding to prop or state changes by updating the DOM.
* **Phase of Unmounting:**  
  componentWillUnmount(): This function is called just before a component is destroyed and unmounted. Perfect for clearing out network requests and getting rid of event listeners.

**Functional Components with Hooks:**

* **Phase of Mounting:**  
  hook for adding state to functional components is useState().  
  useEffect(() => {}, []): Hook that combines the functionality of componentDidUpdate, componentDidMount, and componentWillUnmount. replaces componentDidMount and componentDidUpdate and runs after each render. To ensure that the effect only occurs when those dependencies change, dependencies can be specified using the second argument.
* **Update Stage:**  
  Updates in the functional component are brought about by re-renders due to updates in the props or state.  
  Similar to the updating stage in class components, useState() and useEffect() can be used to handle state updates and side effects, respectively.
* **Phase of Unmounting:**   
    
  The cleanup function that is supplied to useEffect() can be used to carry out cleanup operations. Prior to the component being taken out of the user interface, this function executes.