ASSIGNMENT 3

Hooking is a concept that allows modifying the behavior of a program. It's the chance that code gives you to change the original behavior of something without changing your code of the corresponding class. This is done by overwriting the hook methods

There are seven types of Advanced built-in Hooks.

- Use Reducer Hook = It's a hook that allows you to manage the state of your component. The way the hook works is quite different from use state though.

- Use Callback Hook = A callback is a function (or delegate) that you register with the API to be called at the appropriate time in the flow of processing .

- Use Memo Hook = The useMemo is a hook used in the functional component of react that returns a memoized value.

- Use Ref Hook = useRef hook is part of the React Hooks API.

- Use Imperative Handle Hook = useImperativeHandle React Hook accepts a ref object and a function whose return value replaces the stored value in the ref object and the whole function will be updated when the deps array changes.

- Layout Effect Hook = useEffect hook is called after the screen is painted. Therefore mutating the DOM again immediately after the screen has been painted, will cause a flickering effect if the mutation is visible to the client.

- DebugValue Hook = It enables you to be able to log information in the dev tool in an easier format than just seeing the values of the hook.