**1.** **What is Redux.**

* -Redux is a predictable state container for javascript applications.
* -Redux is a state management library.
* -Redux stores the state of our application.
* -With Redux the state is maintained outside the application not in a particular component.
* -if a component wants to update the state, it communicates with the state container,
* the state container updates the state in predictable manner.
* -Redux 1.0 August 2015
* -React-Redux is the official UI Binding Library for React.
* -React-Redux is the library that provides binding to use React and Redux together in an application.
* -In a typical Redux app there is just a single store with a single Root Reducing function.
* -As Your app grows, you split the Root Reducer in to smaller reducers independently operating on the different parts of the state tree.
* -this is exactly like how there is just one root component in a react app, but is composed out of many small components.

**2.** **What is Core Concepts in Redux.**

* Store : Holds the state(data) of our application. (shop)
* Action : Describes the changes in the state of the application.
* (what happened)(Intention to Buy Cake)
* Reducer : Ties the store and actions together. (shopKeeper) the state container.

**3.** **What are the Principles Of Redux.**

* The state of our whole application is stored in an object tree within a single store. Maintain your whole application state in a single object which would be managed by the Redux store.
* The only way to change the state is to dispatch action, an object that describe what happened. To update the state of your app , you need to let Redux know about that with an action. Not allowed to directly update the state object.
* To specify how the state tree is transformed/updated by actions, we need to write pure Reducers. These ensures that neither the views nor the network callbacks will ever write directly to the state

**4.** **What is Action in Redux.**

* Actions are a plain JavaScript object that contains information. Actions are the only source of information for the store. Actions have a type field that tells what kind of action to perform and all other fields contain information or data. And there is one other term called Action Creators, these are the function that creates actions. So actions are the information (Objects) and action creator are functions that return these actions.

**5.** **What is Reducer in Redux.**

* Actions only tell what to do, but they don’t tell how to do, so reducers are the pure functions that take the current state and action and return the new state and tell the store how to do.
* An action is dispatched with an intention to cause change. This change is performed by the reducer. Reducer is the only way to change states in Redux, making it more predictable, centralised and debuggable.