**Q: Why are we using map?**

<ul className="transaction-list">

            {transactions.map((transObj, ind)=>{

            return(<li>

              <span>{transObj.desc}</span>

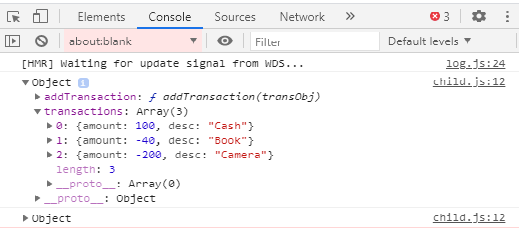
            <span>{transObj.amount}</span>

          </li>)

            } )}

        </ul>

**Ans:** console.log(transactions)



Index

* map will print all the iterations one by one i.e “Cash”,”Book”,”Camera”. These are show due to map function.
* It has best practice, when you run map prototype or object or function, so give unique “key” to your element i.e. <li>.
* That “key” is return in second argument i.e. ind.
* When you pass the “key” then every element (array elements) has send a unique key.
* When the array elements are print via map with unique key or different id.
* “ind” is a iterator and transObj is an object.
* When we run iterators and run dynamic, so we must provide key.
* map second index gives us the id which is the array index i.e. 0, 1, 2.

<ul className="transaction-list">

            {transactions.map((transObj, ind)=>{

            return(<li key={ind}>

              <span>{transObj.desc}</span>

            <span>{transObj.amount}</span>

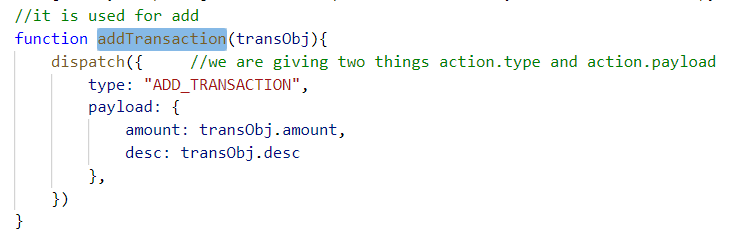
          </li>)

            } )}

        </ul>

Q2:

Ans:

****

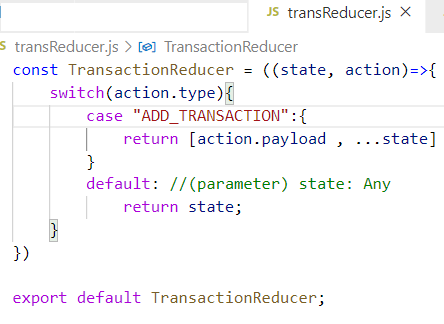
transContext.js

* The above will dispatch in reducer
* When **addTransaction** is called, it will take and pass the transaction object i.e. transObj, then it will call the **dispatch()**, the dispatch() function information is save or store in reducer.
* We will import in child.js

function Child() {

  let {transactions, addTransaction} = useContext(TransactionContext);

* not in component state, we will do in Reducer function.

****

* **action.payload** is the updated value
* **…state** is the three operator, it is obtained by old destructing, it is also called spread operator
* An object has so many keys/values pair.
* So one method is, we refer one object into another object like b=a, then a and b refer to one object technically
* After that you want to merge these values of the object with other values and you want to create a new object.
* The dots mean you want to spread what is state object contains and make a new object.
* **action.payload** and **state** spread(combine) and make a new object or new array will made.
* spread operator (**…**) is used in both array and object.
* so spread operator will break array reference