Day-4 How re-rendering works with Context When a component consumes some contest value and the value of , that component this context changes MI-HINDLIS When it comes to the default behavior of React rendering, if a component renders, React will e cursively me-runder all its children regardless of props or context Top livel component injects
Context provider at the t
function Appl [return < Applantext. Provider> < Camp A /> const comp A = () => < Comp B />
Const comp B = () => < Comp C function.) => null same const Comp C as func CompA(){ New, if App re-renders, A, B, C thru will re-render as well. Appl Context Provider) -> A -> B -> C



	Date: / /20
	If the components are complex in nature, this could result in performance hit.
	this could result in performance hit.
\rightarrow	How to solve this issue?
	his has bained it six at it begins
	React. mino () & use Memo hook
- N	+ Fru mana no vo
X	React. meno ()
	noal higher who are itolasi good in
-	If your component renders the same
	risalt given the same props, you can wrop it in a call to React. memo().
	can wrap it in a call to keact, mimil).
	the a pertormance about by mimbizing.
	the millt.
	Margination is a sylvantamina technique
7	Mindization is a programming technique. That accelerates performance by caching the return values of expensive func.
	Har Maline Maline March Control
	calls.
	CUUS,
	This means that react will skip
7	rendering the component, and rules
	the last rendered Hibelt.
	Harry Truem To an although I'm c-
X	usiMino()
1	Lawrence skip kolis . Litaria shi di .
->	Used for caching a specific calculation
SHAN	or value de acceps 2 arguments.
	emilian instructo without sal six
7	const minoised = use Memo (compute, dependen cies)
	a la table at the decrease the second of the
	(More about hooks in next module)
Marin Control of the	