let a = 100

function func1(){

let b,c

console.log('a is',a)

}

func1()

a = 200

func1()

//returning functions - higher order function

//lexical scoping - inner scope can access parent scope variables

//A closure is the combination of a

//function bundled together (enclosed)

//with references to its surrounding state

//(the lexical environment).

//In other words, a closure gives you

//access to an outer function's scope from

//an inner function.

function outer(name){

let outerVariable = 'Bread'

function inner(){

let innerVariable = 'Butter'

console.log('inner variable',innerVariable)

console.log('outer variable',outerVariable)

console.log('a is',a)

console.log('hello',name)

}

return inner

}

let call1 = outer('Vidya')

call1()

let call2 = outer('John')

call2()

call1()

function makeAdder(x){

return function(y){

return x+y

}

}

let add5 = makeAdder(5)

console.log(add5(10))

let add100 = makeAdder(100)

console.log(add100(20))

console.log(add100(45))

console.log(add5(22))