**Practice 6**

Aim: Explain Function closure with practical

* Closure means that an inner function always has access to the vars and parameters of its outer function, even after the outer function has returned.
* The self-invoking function only runs once. It sets the count to zero (0), and returns a function expression.
* This way counter becomes a function. The "wonderful" part is that it can access the count in the parent scope.
* This is called a JavaScript closure. It makes it possible for a function to have "private" variables.
* A closure is a function having access to the parent scope, even after the parent function has closed.

Flowchart:

**Start**

**Display Result of noClosure function Variable**

**Display Result of withClosure function Variable**

**withClosure ()**

**Function with closure variables**

**noClosure ()**

**Function without closure variables**

**Define variable for the closure**

**End**