Name: Heet Barot

Class: MI-349-730

Advanced JavaScript

The paper acknowledges advanced JavaScript as a mode of programming devoid of a critical concept in it called the closures. Since it is a concept that programmers need to know, a broader way of looking at points to extension of variable scope through closures. Despite being a straightforward concept, it is fairly complex since they allow variable scopes to be extended beyond the restrictions that exist within the functions. To establish closures, one needs to nest a precise function within a function. In such a scenario, the inner function will have the ability to gain access to each variable existing within the scope of a parent function. Therefore, this concept is readily available whenever there is need to create properties and methods in scripts that exhibit object orientation.

A good example below such as the one below is a demonstration of closure.

```
function myObject() {

	this.property1 = "value1";

	var newValue = this.property1;

	this.performMethod = function() {
		myMethodValue = newValue;
		return myMethodValue;

	};

}

var myObjectInstance = myObject();

alert(myObjectInstance.performMethod());
```

Critical portions in this script represent the nested functions which remain anonymous and last line represent the *alert* function. Since this method call describes a nested function, then the alert portion presents the ability to learn and read the *newValue* which represents the value in this given variable. Despite the absence of such a variable in the defined scope of this function – which is basically anonymous, the description still meets the standards of variable reading.

Whenever developers utilize these closures frequently – without their knowledge, a closure is established whenever an anonymous function gets nested within another function. Such happens readily while the same closure can use variables made from the scope of the parent function.

Lastly, the potent of closure becomes eminent whenever an inner function is pronounced. When this happens, values that are rarely accessible within a prescribed scope become accessible in a regional scope hence their utilization for any other value.