**What is the purpose of access modifiers?**

Access modifiers are keywords which protect information and logic by restricting access to it. If something is made private, the information or method will only be accessible by fellow class members. If something is made public, it will be accessible by both fellow class members and code outside of the class (Gaddis, 2016). JavaScript does not have access modifiers, so their functionality must be implemented through the use of closures (Osmani, 2012).

**What are the most commonly used access modifiers in a classical programming language?**

There are three main access modifiers used in classical programming including private, public, and protected.

Private: private members can only be accessed by other members inside the same class.

Public: public members can be accessed by any member inside or out of the class.

Protected: protected members can be accessed by sibling members inside the same class or child members which exist inside of subclasses.

(Gaddis, 2016)

**How are closures used to enforce “private and public” access?**

Closures encapsulate data inside of a specific lexical scope, so that they can only be accessed inside of that scope or by using methods which can reach into that scope. These accessor methods are returned from the scope so that they can be used elsewhere, but can still access data inside of the private scope of a function (“Closures - JavaScript | MDN,” n.d.).

Closures - JavaScript | MDN. (n.d.). Retrieved February 26, 2018, from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>

Gaddis, T. (2016). *Starting Out With Java, From Control Structures Through Objects* (6th ed.). Pearson Education.

Osmani, A. (2012). *Learning JavaScript Design Patterns*. O’Reily Media Inc.