**Horse Race Simulator Report**

This report’s purpose is to document improvements made to the **Horse** and **Race** class of the simulator. It documents identified issues alongside a thorough explanation, providing possible solutions and testing evidence. Furthermore, it also explains key programming concepts used.

**Horse Class:**

The horse class makes use of encapsulation, a fundamental concept in Object Oriented Programming. This involved bundling related data alongside methods that utilise that data into a single unit (i.e., Class). This is done by setting the access modifiers for the class fields to **private**, which prevents access to the fields outside the Class, only allowing them to be manipulated via accessor (**getter**) and mutator (**setter**) methods.

The following are the mutator methods:

* getConfidence(): Returns the confidence level of the horse
* getDistanceTravelled(): Returns the total distance travelled by the horse
* getName(): Returns the horse’s name
* getSymbol(): Returns the character used to represent the horse
* hasFallen(): Returns a Boolean value whether the horse has fallen

The following are the accessor methods:

* fall(): Sets the horse as fallen
* moveForward(): Increments the distance travelled variable by one
* setConfidence(): Sets the confidence rating of the horse to the given value
* setSymbol(): Sets the horse’s symbol to the character given

The following is a private method I have made:

* validateConfidence():
  + This method takes one param: “double confidence”. Uses an if statement to check if the param exceeds the bounds. If the confidence is >= 1.0, it returns 0.99, else if the confidence is <= 0.0, it returns 0.01, else it returns the param as it was.
  + This method has been used within the constructor and setConfidence() to make sure the confidence value does not exceeds the bounds (0 and 1)