**CHAPTER 6 [1]**

* conditional statement - a statement that you can use to execute code based on a condition or group of conditions
* It allows a program to make decisions based on these conditions
* The basic conditional logic structures use the if/else and switch statements
* The if/else statement block consists of 4 parts: the if statement, the comparison, the code to execute if the comparison returns true, and an optional else block containing code to be executed if the comparison returns false
* Indentation has no effect on how your program executes
* Nesting is the process of putting one structure inside another structure
* Code can make decisions based on multiple conditions by nesting if/else statements
* Complex comparisons use logical operators.
* Complex comparisons can be used to check for multiple conditions is a single comparison statement.
* Chaining if/else statements is another variation for checking multiple conditions
* After one condition is checked and determined to be false, then the other condition is checked. This can be chained through several iterations.
* Some languages hand an elseif statement. Javascript does not, but can achieve the same results.
* Switch statements control program flow by executing a specific set of statements, depending on the value of an expression.
* Compares value of an expression to a value contained in a case.
* Frequently used instead of chained if-else statements.
* Case – in a switch statement, each *case* represents a specific value
* Each case contains one or more statements that execute if the value of the case matches the value of the switch statement’s expression
* Consists of the keyword case, followed by a literal value or variable name, followed by a colon.
* Default – used within switch statements
* It contains statements that execute when the value returned by the switch statement expression does not match any case
* Consists of the keyword default followed by a colon
* Break – used to exit out of the switch block
* Conditional operator – can be used to shorten the amount of code needed for simple if-else statements
* Often called a ternary operator as it is the only Javascript operator that requires three operants
* Syntax: condition ? value1 : value2

**CHAPTER 6 [2]**

* Loop – a block of code that can be repeated a certain number of times
* allows code to be made shorter and more efficient
* For Loop – loops through a block of code a given number of times
* Works well with arrays
* As with the if/else blocks, you can also nest for loops
* Ensure when you nest for loops, you use different counter variables for each loop
* With nested loops, the inner loop will complete on each iteration of the outer loop
* While loop – tests a comparison and loops through a block of code until the comparison is no longer true
* in a while loop, you must ensure that you change the value of the comparison variable within the loop or you will be stuck in an infinite loop
* a while loop is a pre-test loop which means the comparison test is done before the loop is executed
* Do while loop – tests a comparison and loops through a block code until the comparison is no longer true
* A do while loop is a post-test loop which means the comparison test is done after the loop is executed the first time
* Therefore, a do while loop always executes at least one time
* The break statement can be used to break out of a loop
* Should be used sparingly. Results in multiple exit points from a loop
* The continue statement is used to stop the current iteration of a loop and restart the loop with the next iteration

**CHAPTER 7**

* Array – a collection of data values stored in a single variable
* Each data value is identified by an index number
* The index of arrays is zero based
* Javascript arrays are actually a type of object
* Javascript arrays can contain any variable type, objects, functions, and even other arrays
* Arrays are identified starting at [0]
* An array element’s index is its numbered position in the array
* There are two ways to declare an array: using the array constructor and using an array literal
* The naming rules for arrays are the same as for variables
* Array names should normally be a plural name
* The array length can be optionally specified [constructor]
* The array element values can be optionally specified [constructor/literal]
* To store values in an array, use the array name, followed by an index number in brackets and an assignment expression
* Its just like assigning values to variables, except for the brackets containing an index number
* To access values in an array, use the array name followed by an index number in brackets
* Its just like accessing values of variables, except for the brackets containing an index number
* Every array object has a property named length
* The length property contains the length of the array
* Using the length property of an array allows the use of loops to either populate or access an array
* The length property can be accessed using standard dot notation which has the name of the object followed by the name of the property

CHAPTER 8

CHAPTER 10