Reading Quiz 1

* Computer Programming
  + The process of designing an implementing an executable computer program for accomplishing a specific computing task.
* C#
  + An easy to learn, modern, general-purpose, object-oriented programming language developed by Microsoft.
* Visual Studio
  + The professional integrated development environment most commonly used to program in C# or other .Net languages.
* Console Application
  + A .NET application that runs from the command line (rather than with a graphical user interface)
* Main
  + The entry point for every C# application.  This is the method that executes when you run your program.
* Console
  + A software component that represents the command line.  A programmer can use it to write to the command line or read from the command line.
* WriteLine
  + A method of the Console class that displays a message to the user at the command line.
* ;
  + Terminates every statement in a C# program.
* { and }
  + Create a block of code in a C# program.  These surround the body of the Main method in all of the Hello World examples.
* “
  + Delimit a string, or set of characters, in a C# program.

Reading Quiz #2

* Variable
  + A named identifier use to store a value in memory while your application is running.  The value can change as your application is running.
* scope of a variable
  + The area of a program in which a variable is known and can be used.  Local and global are examples.
* weakly typed (implicitly typed) programming language
  + JavaScript is this type of programming language.  It determines the data type of a variable based on the context in which it is used.  In fact, you can use a variable without declaring it AND you never declare a datatype.
* assignment statement
  + A programming statement that gives a variable a value.  The variable name goes on the left hand side and the value or expression that the variable should have goes on the right hand side.
* strongly and explicitly typed programming language
  + C# is this type of programming language.  You must declare a variable and its datatype before using the variable.  This allows the compiler to ensure that a programmer is manipulating data in a program in a reasonable way.
* Int
  + One of several data types that can be used to store whole numbers.  In the current version it is stored in 32 bits in memory.
* double
  + One of two data types that can be used to store real numbers.  It is the default data type for a floating point literal in C#.
* Decimal
  + The C# data type most typically used to store monetary values.  When you intend for a literal value to have this datatype you MUST add m (or M) to the end of the value.
* Casting
  + The process of temporarily converting the value in a variable to another data type for the purposes of a calculation.  In the expression below the variable miles is converted to a double for the purposes of the calculation.  It is an example of this process.
  + double mph = (double) miles / gallons;
* IPO Chart
  + A program design tool that helps outline a very general solution to a programming problem.  It focusses on identifying the Input and Output for a problem as well as the general Process steps required to solve the problem.

Reading Quiz #3

* Algorithm
  + A step by step set of instructions that describe how to perform a specific task.  You use one of these when you follow a recipe, when you tie your shoes and when you do long division.
* Pseudocode
  + An English language outline like tool for expressing an algorithm.
* !=
  + The inequality operator.
* switch statement
  + An example of a selection statement.  It matches a control variable or expression to one of a set of "cases".  The statements in the block following the case will execute when the control variable and the case value match.
* Selection
  + One of three control structures available in all programming languages.  It allows programs to branch or make choices.  The others are sequence and repetition.
* if statement
  + An example of a selection statement.  It executes one (or more) blocks of code depending on the truth of one (or more) conditions.
* ==
  + The equality operator.
* =
  + The assignment operator.
* relational operator
  + An operator that is used for comparison.  >, <, >=, <=, ==, != are examples
* logical operator
  + An operator that allows you to combine simple conditions to make more complicated conditions.  And (&&), Or (||), Not (!) are examples.

Reading Quiz 4

* While loop
  + A pre-test condition that executes as long as a condition is true.
* Random
  + The class that can be used to generate random numbers
* Next
  + The method in the Random class that generates a random integer between 2 values
* ++
  + The increment operator in C#. It adds one to the variable that it is used with.
* --
  + The decrement operator in C#. It subtracts one from the variable it is used with.
* Selection
  + One of the 3 control structures (building blocks that control the flow of execution of a program) available in all programming languages. It allows programs to branch or make choices.
* Do While loop
  + A post test loop. Because the condition is at the bottom of the loop, the body of the loop will always execute at least once.
* For Loop
  + One of the repetition statements available in C#. The heading for the block has 3 parts (initialization; condition; update)
* Sequence
  + One of the 3 control structures (building blocks that control the flow of execution of a program) available in all programming languages. In absence of anything else, program statements are executed in the order that they are written.
* Repetition
  + One of the 3 control structures (building blocks that control the flow of execution of a program) available in all programming languages. It allows programmers to execute a block of code more than once without having to copy and paste.

Quiz #5

* Parameter
  + An extra piece of information passed to a method to customize it for a specific need.  We added two of these to the method getInt, min and max, to allow the method to make sure that the integer value is within a range.
* pass by value or copy
  + The default parameter passing mechanism for simple data in C#.  A copy of each parameter value is passed to the method.  Because a copy is passed, the original data in the calling method remains unchanged.
* pass by reference
  + The parameter passing mechanism used when you want to indirectly return more than one value from a method.  A reference to the data is passed to the method.  Because a reference is passed, changing the data in the method causes the data to be changed in the calling method.
* Return
  + This keyword ends a method.  When it is followed by a value, it send the value back to the calling method.
* method heading
  + The first line of a method. It starts with the return data type.  It includes the method name and ends parentheses.  Sometimes a list of parameters is included in the parens.
* method
  + A named group of programming instructions that perform a specific task.  A reusable abstraction that reduces the complexity of writing and maintaining programs.
* method body
  + The block of code that executes when the method is called.  The "guts" of the method.
* Cohesion
  + A module design principle that refers to the degree to which the elements inside a module belong together.  As a guideline, each method should do ONE thing.
* Coupling
  + A module design principle that measures the degree of interdependence between software modules.  As guideline, programs should pass data between methods using parameters and return values.  Global variables should be avoided.
* local variable
  + A variable that is declared inside a method.  It is known and can be used only inside the method.

Reading Quiz #6

* {}
  + The delimiters used when initializing an array with a set of literal values.
* []
  + The delimiters used with an array index.
* Value type
  + int, double and bool (not a complete list) are this type of datatype in C#.  The memory location for an int variable holds a value and the default parameter passing mechanism for these datatypes is pass by value.
* 2-D Array
  + A data structure used to store a table of values.  Each individual value is referenced using 2 indices, one for the row and another for the column.  You might use this data structure when implementing a game of TicTacToe.
* Array
  + A data structure used to store a list of values.  Each individual value is referenced using a position or index.
* Index
  + The "position" or "location" of an individual element in an array.  The first element in the array is at 0.
* Length
  + The array property that tells you how many elements are in the array.
* Int[]
  + The datatype of an array of integers.  It is used at the beginning of the declaration of an array of integers as well as before a parameter that is an array of integers.
* Reference type
  + An array is this kind of datatype in C#.  The memory location for an array variable holds the address of the array rather than the elements in the array.   Parameters of this type are passed by reference.
* 0
  + The first element in an array has this index.