1. Variables that exist only in a method or another small section of code.
2. A command that performs an action
3. Identifiers are the names that you use to identify the elements in your programs, such as namespaces, classes, methods, and variables.
4. An identity reserved by C# for its own purposes.
5. A bucket to store a value
6. Assign a data type and variable name: int myVar
   1. myVar = 10; – use the assignment operator
   2. We can use a variable without a value
   3. Yes. Int myVar = 10;
7. Visual studio automatically creates a new solution file for the project resulting in multiple solution files.
8. You cannot use all arithmetic operators on all variable types. For example, you cannot subtract two strings.
9. int intVar = 42; string stringVar = intVar.ToString();
10. string stringVar = "42"; int intVar =  System.Int32.Parse(stringVar);
11. Precedence controls the order in which expressions are evaluated. \* / % operators have precedence over + -. So 2 + 6 \* 2 = 14.
    1. Associativity comes into play when using operators that have the same precedence. Left and right associativity is which operand is evaluated first. Arithmetic operators read right to left.
12. The definite assignment rule is that you must assign a value to a variable before you can use it.
13. Postfix evaluates the variable then modifies it. The Prefix modifies the variable then modifies it.
14. Instead of using the + to append to a string we can use a C# feature called string interpolation. If we start the string with a dollar sign we can add to that string with a value inside curly braces. {}
15. Var tells C sharp to imply the variable type rather than you explicitly declare it.