Glossary

https://csci-1301.github.io/about#authors

September 5, 2023 (05:12:31 PM)

Contents

Keywords/Reserved Words[1]: words defined by the C# language and used for one thing
Datatypes[2]: categories in C# used to define types of values, such as strings
Variable[3]: values that be changed
Constant[4]: values that can not be changed
Identifier[5]: words defined by the programmer to refer to an object or variable.
Operations:
Operators[6]: symbols used to perform operations
Modulo[7]: the % used to divide two numbers and return the remainder
Escape Sequence[8]: used to represent a non-printable character
Reference Types (Objects and Strings)[9]: a variable of a class object holds a reference to the address of the object on the managed heap.
Value Types (all other reserved words)[10]: a variable of a class object stores the exact data value held by the variable
Numeric Types[11]:
Booleans[12]: a binary datatype that can only be true or false

Decision Structures (`if`/`else`/`switch`)[13]:

Control Structures (loops)[14]:

Instantiation (instance of a class)[15]: the act of creating a object, an instance of a C# class
Initialization[16]: the act of both declaring a variable's datatype and identifier and assigning it value
Declaration[17]: the act of creating a variable's datatype and identifier
Assignment[18]: the act of giving a value to an identifier
Implicit Conversion[19]: the act of automatically storing the value of one identifier into another identifier that differs from its own
Explicit Conversion (casting)[20]: the act of storing the value of one identifier into another identifier that differs from its own using additional syntax
Rules[21]: are required syntactical ways to write a program for it to function Conventions[22]:
are not required for the program to function, but are heavily encouraged for the readability and comprehension of other programmers
Format Specifiers (C, N, P, E)[23]: added to variable calls in strings to format the numeric variable in various ways (see link for specifics)
Constructor[24]: a method used to instantiate an object and assign it's attributes
Parameter[25]: any variable declared within a method
Argument[26]: any value that must passed to a method in order for it to be called

---

[13]https://csci-1301.github.io/book.html#decisions-and-decision-structures
[14]https://csci-1301.github.io/book.html#loops-increment-operators-and-input-validation
[22]https://csci-1301.github.io/book.html#conventions-of-c-programs

Attribute[27]: the variables declared within a class to act as the characteristics of any of its instantiated object

Method[28]: a code block that contains a series of statements

Class Member (attributes and methods)[29]: anything defined within a class that can be access within and outside of the class

Scope[30]: Time and place in program where the variable exists

Iterator: an object that traverses an array or list

Sentinel Value[31]: a special value in the context of an algorithm which uses its presence as a condition of termination, typically in a loop or recursive algorithm.

Guard Condition: boolean expressions (predicates) found at the top of a method or function that determine whether the function should continue to execute.

Accumulator[32]:

Counter[33]: a variable used to count the number of times a certain condition is met

Complex Condition[34]: a condition consisting of multiple conditions

Method Signature[35]: the way a computer reads a method by its name and the datatype of its parameters

Method Overloading[36]: the act of creating multiple methods with the same signature

Return Type[37]: the datatype of any value returned from a called method

UML Diagram[38]: a written diagram used to display a class and all of its members

Input Validation[39]: whenever a program checks if the user gave a usable input and responds accordingly to avoid errors implementation