What languages use garbage collection?

Lisp has used garbage collection since John McCarthy invented it in 1958. Java, Scala, Python, and .NET/C# are all popular GC languages. Additional garbage collection languages include the relatively young Go, Ruby, D, OCaml, and Swift, as well the older languages Eiffel, Haskell, ML, Modula-3, Perl, Prolog, Scheme, and Smalltalk.

Java, Python, and .NET/C# are some of the more popular programming languages that implement garbage collection. The Java virtual machine (JVM) actually provides four different garbage collectors: serial, parallel, concurrent mark and sweep, and G1GC, the garbage first garbage collector. G1GC is now the default in Java; it is a regionalized and generational parallel compacting collector that achieves soft real-time goals.

Python, specifically the standard CPython implementation, combines reference counting with three-level generational collection that only focuses on cleaning container objects. The .NET CLR (common language runtime) uses a three-level generational mark and compact collection algorithm. The CLR also segregates memory objects into two heaps, one for large objects (85,000 bytes or higher) and one for small objects; the large object heap usually isn't compacted, just marked and swept, but can be compacted if necessary.

Many programming languages require garbage collection, either as part of the language specification (e.g., RPL, Java, C#, D, Go, and most scripting languages) or effectively for practical implementation (e.g., formal languages like lambda calculus). These are said to be garbage-collected languages