|  |  |  |
| --- | --- | --- |
| Category | Stack Memory | Heap Memory |
| What is Stack & Heap? | It is an array of memory.  It is a LIFO (Last In First Out) data structure.  In it data can be added to and deleted only from the top of it. | It is an area of memory where chunks are allocated to store certain kinds of data objects.  In it data can be stored and removed in any order. |
| Practical Scenario | https://csharpcorner-mindcrackerinc.netdna-ssl.com/article/stack-vs-heap-memory-c-sharp/Images/ex11.gif  Value of variable storing in stack | https://csharpcorner-mindcrackerinc.netdna-ssl.com/article/stack-vs-heap-memory-c-sharp/Images/ex122.gif  Value of variable storing in heap |
| What goes on Stack & Heap? | "Things" declared with the following list of type declarations are Value Types  (because they are from System.ValueType):  bool, byte, char, decimal, double, enum, float, int, long, sbyte, short, struct, uint, ulong, ushort | "Things" declared with following list of type declarations are Reference Types  (and inherit from System.Object... except, of course, for object which is the System.Object object):  class, interface, delegate, object, string |
| Memory Allocation | Memory allocation is Static | Memory allocation is Dynamic |
| How is it Stored? | It is stored Directly | It is stored indirectly |
| Is Variable Resized? | Variables can’t be Resized | Variables can be Resized |
| Access Speed | Its access is fast | Its access is Slow |
| How is Block Allocated? | Its block allocation is reserved in LIFO.  Most recently reserved block is always the next block to be freed. | Its block allocation is free and done at any time |
| Visibility or Accessibility | It can be visible/accessible only to the Owner Thread | It can be visible/accessible to all the threads |
| In Recursion Calls? | In recursion calls memory filled up quickly | In recursion calls memory filled up slowly |
| Used By? | It can be used by one thread of execution | It can be used by all the parts of the application |
| StackOverflowException | .NET Runtime throws exception “StackOverflowException” when stack space is exhausted | - |
| When wiped off? | Local variables get wiped off once they lose the scope | - |
| Contains | It contains values for Integral Types, Primitive Types and References to the Objects | - |
| Garbage Collector | - | It is a special thread created by .NET runtime to monitor allocations of heap space.  It only collects heap memory since objects are only created in heap |

