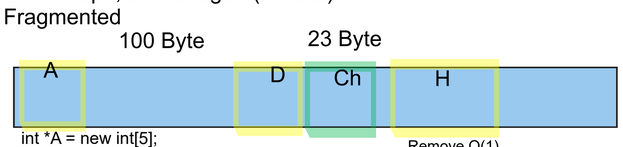
Garbage Collector
part of CLR , for managing Data @ Heap

C , C++ , .Net
Stack
Add O(1) , Remove (1)

C++ Heap , UnManaged (No GC)
Fragmented




```
int *A = new int[5];
B= new ()
C = new ();
D = new();
Delete C;
Delete B
.....
.....
char *Ch = new Char[20];
```

Remove O(1)
Add O(N)**
Search Best Fit Location

.Net Heap
Managed by GC

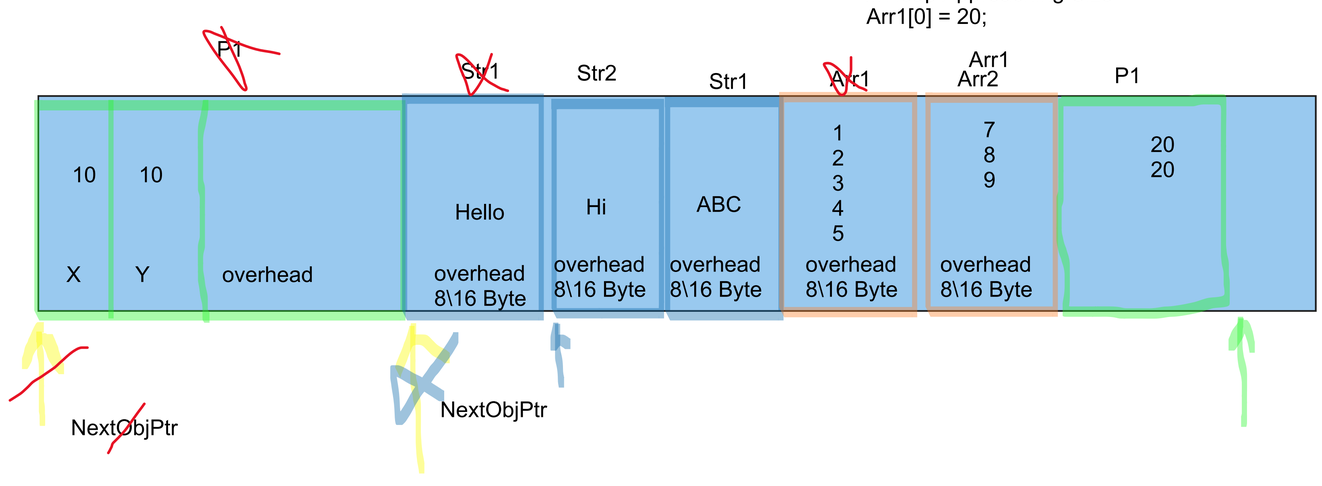
Adding into .Net Heap as fast as Stack

Dynamic Size ,
Initial Budget : 12.256MB

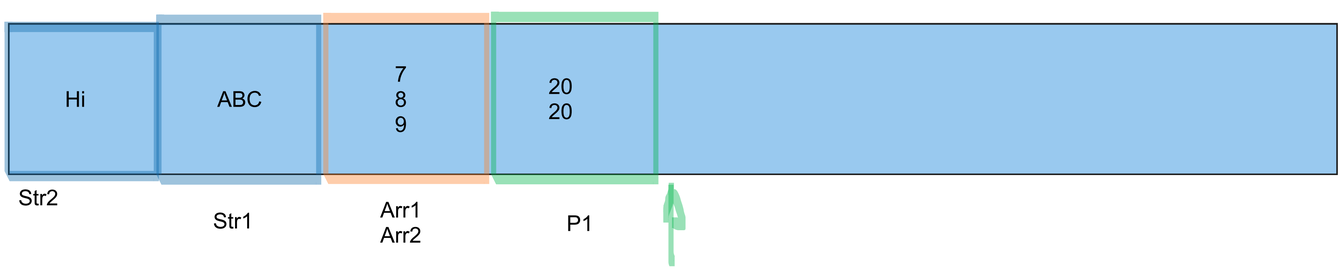
1. new Allocate Required # of Bytes in Heap (Object Size + Overhead Variables )
( 8 Bytes + 2 * 32\64 Bit ) = 24 Byte after NextObjPtr
2. crossout (Initialize) allocated Bytes
3.Call UserDefined Ctor if Exists Ctor(this:NextObjPtr)
4. Assign Reference
5.Advance NextObjPtr

```
Point P1 = new Point(10, 10);
String Str1 = "Hello";
String Str2 = "Hi";
Str1 = "ABC";
int[] Arr1 = { 1, 2, 3, 4, 5 };
int[] Arr2 = { 7, 8, 9 };
P1 = new Point(20, 20);
Arr1 = Arr2;
///Heap Approaching Size
Arr1[0] = 20;
```

GC Run
1.Heap Gen 0 Approach Initial Budget
2.User Call GC.Collect();
3. OS Enforce CLR to Run GC for All App , when Low Memory Detected by OS
4. Application Exit (*******)-Finalizer

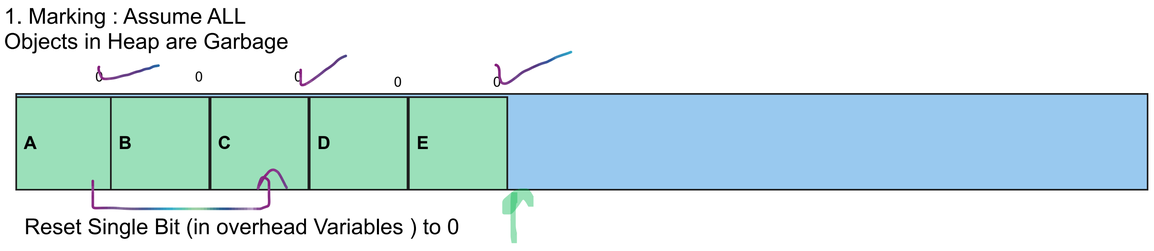



Heap After Complete
First Run to GC




Mark &
Collect (Compact)
Reference Correction

1. Marking : Assume ALL
Objects in Heap are Garbage




Reset Single Bit (in overhead Variables ) to 0

Starting Application Roots, try to vist ALL Objects , Flip Bit to 1
in any Reatchable Objects ,

Upon Visting Object , try to vist all Reachable Objects from
this Object ( visit Assosiated\Aggregated\Composited Objects)

2. Compacting : Shift All Surviving Objects Together

Reference Correction in
Application Roots

End of GC Run , Restart Application normal path of Execution

Heap Generations

Gen 0 : any newly Allocated Object , from Gen 0
Allocation to new Objects in Gen0 , Till Gen 0 Approach Initial Budget (256 KB)
Aproach Initial Budget for Gen 0 Fire Collection Phase of GC
Any Surviving Object from Gen 0 will be Promoted to Gen 1
Mark & Compact from Gen 0 , Upgrade to Gen 1 Till Gen 1 Approach Initial Budget (2 MB)
GC will Mark & Compact Both Generation
Any Surviving Object from Gen 01 will be Promoted to Gen 2 (Highest Generation)
Gen 2 Initial Budget (10 MB)