* NORMAL USING OF STRING CAN LEAD US TO THE LAST VARIABLE THAT IS DECLARED:
* NO GARBAGE OR ORPHAN (HAS NO POINTER TO POINT) is generated.
* No loss of memory

using System;

using System.Collections.Generic;

namespace ConsoleApp3

{

class Program

{

public static void Main(string[] args)

{

string ob = "Soham";

ob = "and";

ob = "Ivy";

ob = "gonna";

ob = "marry";

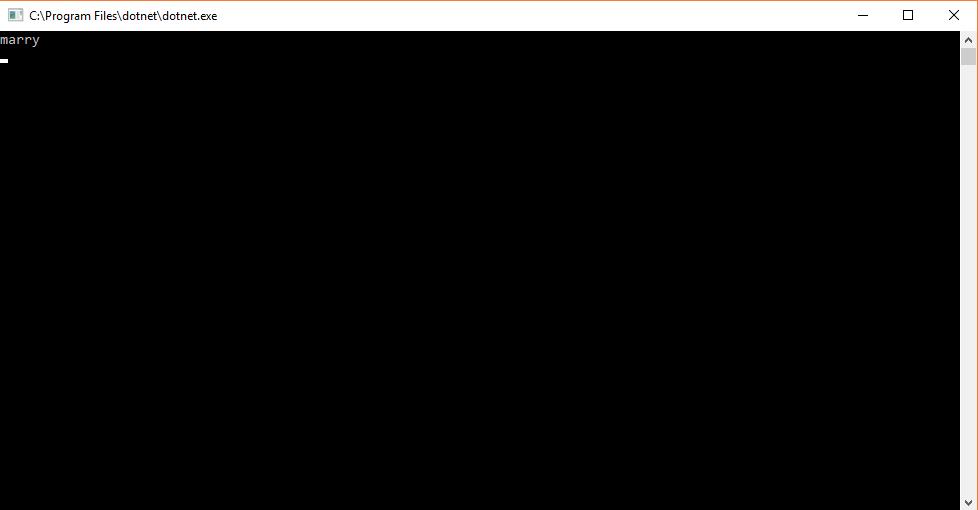
Console.WriteLine(ob);

Console.ReadKey();

}

}

}



USING STRING (RECIDES: System.String namespace)

* += (concat) USING OF STRING CAN LEAD US TO concatenation of strings
* GARBAGE OR ORPHAN (HAS NO POINTER TO POINT) is generated.
* loss of memory

using System;

using System.Collections.Generic;

namespace ConsoleApp3

{

class Program

{

public static void Main(string[] args)

{

string ob = "Soham";

ob += " and";

ob += " Ivy";

ob += " gonna";

ob += " marry";

Console.WriteLine(ob);

Console.ReadKey();

}

}

}

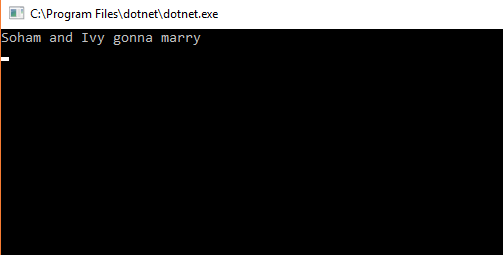
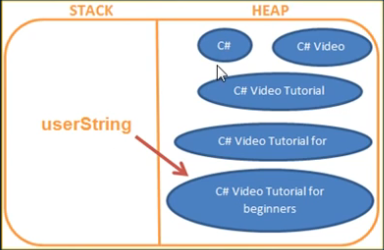


Diagram:



THIS IS IMMUTABLE;

WHEN WE GIVE FIRST VAL IS IS CONCATINATED WITH PREVIOUS. AND LEFT A COPY AS ORPHAN,

THE ORPHANS BECOME THERE UNTILL A GARBAGE COLLECTOR RUNS AND DESTROY THEM ALL, WASTES A LOT OF MEMORY.

USING STRING BUILDER CLASS (RECIDES: System.Text)

using System;

using System.Text;

using System.Collections.Generic;

namespace ConsoleApp3

{

class Program

{

public static void Main(string[] args)

{

StringBuilder sb = new StringBuilder();

sb.Append("Soham");

sb.Append(" gonna");

sb.Append(" marry");

sb.Append(" Ivy");

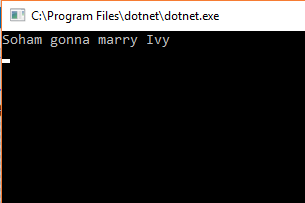
Console.WriteLine(sb);

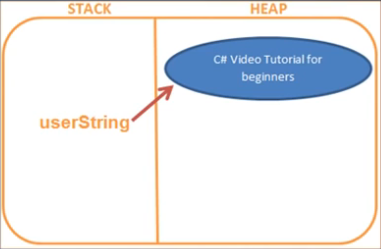
Console.ReadKey();

}

}

}





This is MUTABLE

1. mutability
2. residence
3. performance