

StringBuilder

- As we know that String is an Immutable class and the aim of the String class is Memory Efficiency.
- But StringBuilder is Mutable and it does not provide memory efficiency like String class. If we create the same named string, it always creates a new object of the StringBuilder class inside the heap memory.
- The aim of using StringBuilder is that if there is a situation where we have to do a lot of manipulation in the string then String class will create the new String object each and every time. But on the other side, StringBuilder is mutable. So it doesn't create a new StringBuilder Object, but it can mutate or can make changes inside an already created StringBuilder Object. So It will only occupy memory once.
- Initially the constructor of the StringBuilder object will create a specific length byte array. And if we made any changes inside the StringBuilder object then behind the scene the changes will be made inside the origin byte array. Whereas in the String class, changes will be made inside the new byte array. So it creates a new object while modifying String.
- If StringBuilder object's initial byte array gets full then it will create a new byte array, then it will copy old characters inside it and then will assign reference of that newly created byte array inside the old one. Thus the old byte array will be removed by the Garbage collector.
- **Example with Important Methods...**

```
StringBuilder s1= new StringBuilder("Rahul");

//      StringBuilder s2= new StringBuilder("Rahul");
//      System.out.println(s1==s2); //false

//      System.out.println(s1.length()); //5

//      System.out.println(s1.charAt(2)); //h
//
//      s1.setCharAt(1, 'o');
//      System.out.println(s1); //Rohul

//      s1.insert(0, "P");
//      System.out.println(s1); //PRahul
```

```
//      s1.delete(0,1);  
//      System.out.println(s1); //ahul  
//      s1.delete(0,3);  
//      System.out.println(s1); //ul  
  
//      s1.append(" Bhutaiya");  
//      System.out.println(s1); //Rahul Bhutaiya  
  
//      System.out.println(s1.toString()); //Rahul //converts StringBuilder into  
String
```