StringBuilder objects are like string objects, except that they can be modified or changed. Hence it is also used to create mutable (modifiable) string object. StringBuilder is also same as StringBuffer except for one important difference that StringBuilder is not synchronized that means it is not thread safe whereas StringBuffer is synchronized that means it is thread safe i.e., multiple threads cannot access it simultaneously. It is available since Java 1.5.

StringBuilder class provides an API compatible with StringBuffer but there is no such guarantee of synchronization. This class is designed for use as the replacement of StringBuffer class where the StringBuffer was being used by a single thread. It is recommended to use this class rather than StringBuffer due to its speed under most implementations.

Instances of StringBuilder are not safe for use by multiple threads.

Constructors of StringBuilder class –

1. **StringBuilder ( ) -** Creates an empty StringBuilder and reserves room for 16 characters.
2. **StringBuilder (int size) -** Create an empty string and takes an integer argument to set capacity of the buffer.
3. **StringBuilder (String str) -** Create a StringBuilder object and initialize it with string str.

Important methods of StringBuilder class –

- **public StringBuilder append(String s) –**

It is used to append the specified string with this string. The append() method is overloaded like append(char), append(boolean), append(int), append(float), append(double) etc.

- **public StringBuilder insert(int offset, String s) –**

It is used to insert the specified string with this string at the specified position. The insert() method is overloaded like insert(int, char), insert(int, boolean), insert(int, int), insert(int, float), insert(int, double) etc.

- **public StringBuilder replace(int startIndex, int endIndex, String str) –**

It is used to replace the string from specified startIndex and endIndex.

- **public StringBuilder delete(int startIndex, int endIndex) –**

It is used to delete the string from specified startIndex and endIndex.

- **public StringBuilder reverse() –**

It is used to reverse the string.

- **public int capacity() –**

It is used to return the current capacity.

- **public void ensureCapacity(int minimumCapacity) –**

It is used to ensure the capacity at least equal to the given minimum .

- **public char charAt(int index) –**

It is used to return the character at the specified position.

- **public int length() –**

It is used to return the length of the string i.e. total number of characters.

- **public String substring(int beginIndex) –**

It is used to return the substring from the specified beginIndex.

- **public String substring(int beginIndex, int endIndex) –**

It is used to return the substring from the specified beginIndex and endIndex.

Different Methods Details of StringBuilder Class –

- **StringBuilder append() method -**

The StringBuilder append() method concatenates the given argument with this string.

```
class StringBuilderExample
{
        public static void main(String args[])
        {
                StringBuilder sb=new StringBuilder("Hello ");
                sb.append("Java");              //now original string is changed
                System.out.println(sb);         //prints Hello Java
        }
}
```

- **StringBuilder insert() method -**

The StringBuilder insert() method inserts the given string with this string at the given position.

```
class StringBuilderExample2
{
        public static void main(String args[])
        {
                StringBuilder sb=new StringBuilder("Hello ");
                sb.insert(1,"Java");            //now original string is changed
                System.out.println(sb);         //prints HJavaello
        }
    }
```

- **StringBuilder replace() method –**

The StringBuilder replace() method replaces the given string from the specified beginIndex and endIndex.

```
class StringBuilderExample3
{
        public static void main(String args[])
        {
                StringBuilder sb=new StringBuilder("Hello");
                sb.replace(1,3,"Java");
                System.out.println(sb);                    //prints HJavalo
        }
}
```

- **StringBuilder delete() method -**

The delete() method of StringBuilder class deletes the string from the specified beginIndex to endIndex.

```
class StringBuilderExample4
{
        public static void main(String args[])
        {
                StringBuilder sb=new StringBuilder("Hello");
                sb.delete(1,3);
                System.out.println(sb);                    //prints Hlo
        }
}
```

- **StringBuilder reverse() method –**

The reverse() method of StringBuilder class reverses the current string.

```
class StringBuilderExample5
{
    public static void main(String args[])
    {
            StringBuilder sb=new StringBuilder("Hello");
            sb.reverse();
            System.out.println(sb);//prints olleH
    }
}
```

- **StringBuilder capacity() method -**

The capacity() method of StringBuilder class returns the current capacity of the Builder. The default capacity of the Builder is 16. If the number of character increases from its current capacity, it increases the capacity by (oldcapacity*2)+2. For example if your current capacity is 16, it will be (16*2)+2=34.

```
class StringBuilderExample6
{
    public static void main(String args[])
    {
            StringBuilder sb=new StringBuilder();
            System.out.println(sb.capacity());              //default 16
            sb.append("Hello");
            System.out.println(sb.capacity());              //now 16
            sb.append("java is my favourite language");
            System.out.println(sb.capacity());
            //now (16*2)+2=34 i.e (oldcapacity*2)+2
    }
}
```

===