

Q1. Explain StringBuffer class in Java?

Ans:

String(Immutable):

StringBuffer(Mutable): If our content are not fixed and we want to perform some modification but thread safety is required then we should go for String Buffer class

StringBuilder(Mutable): If our content are not fixed and we want to perform some modification but thread safety is not required then we should go for String Builder class

Constructor:

1. StringBuffer();

//default initial capacity is 16

New capacity =(current capacity +1)\*2

2. StringBuffer(String str):

//default capacity=default capacity

SB+length of the String

3. StringBuffer(int initialCapacity)

The screenshot shows a Windows desktop environment. At the top, a Notepad window titled 'B1 - Notepad' contains the following Java code:

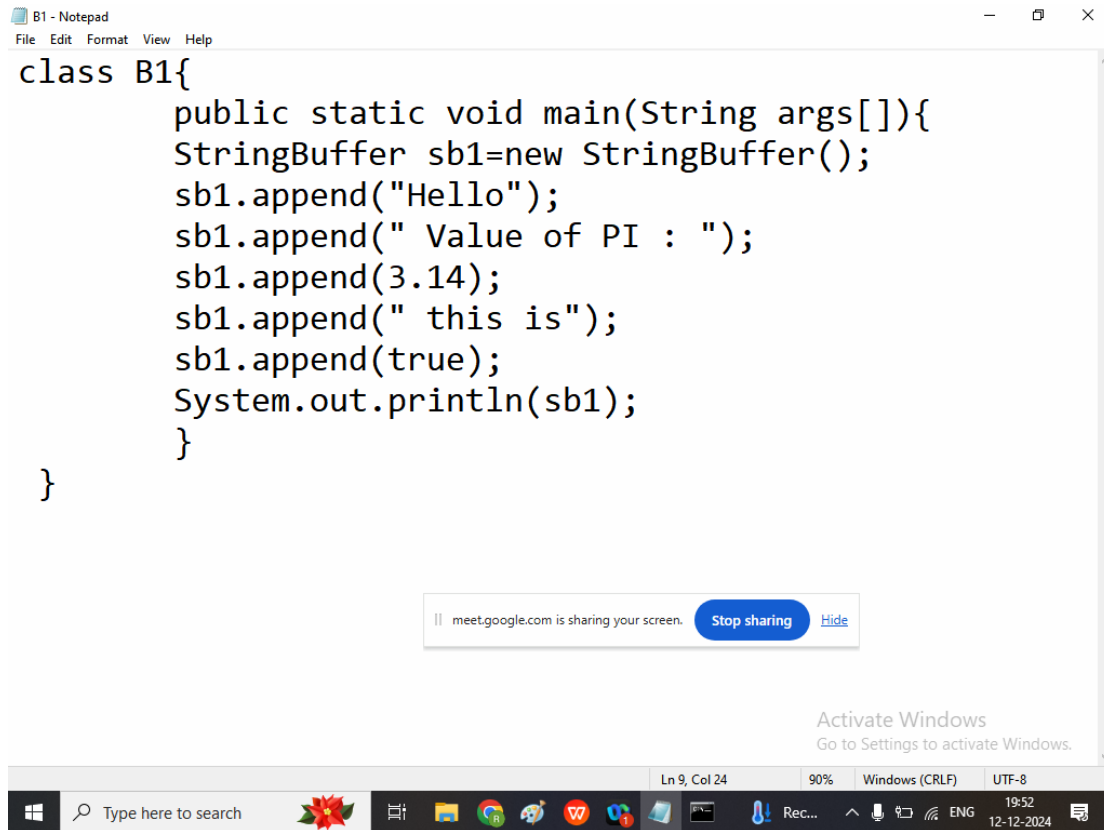
```
class B1{  
    public static void main(String args[]){  
        StringBuffer sb1=new StringBuffer();  
        System.out.println(sb1.capacity());  
        StringBuffer sb2=new StringBuffer("hello");  
        System.out.println(sb2.capacity());  
        StringBuffer sb3=new StringBuffer(1001);  
        System.out.println(sb3.capacity());  
    }  
}
```

Below the Notepad window, a notification from 'meet.google.com' states 'is sharing your screen.' with 'Stop sharing' and 'Hide' buttons. In the bottom right corner, an 'Activate Windows' watermark is visible with the text 'Go to Settings to activate Windows.'

The Windows taskbar at the bottom includes the Start button, a search bar with the text 'Type here to search', and several pinned application icons. The system tray on the right shows the date and time as '12-12-2024 19:48' and the temperature as '18°C'.

## Methods of StringBuffer and StringBuilder

1. `int capacity()`: it will return current capacity of the SB Object
  2. `append()`: It is used to insert data at the end of the StringBuffer object
- `append(String)`
  - `append(int)`
  - `append(float)`
  - `append(String)`
  - `append(char)`
  - `append(boolean)`



The screenshot shows a Windows 10 desktop environment. A Notepad window titled 'B1 - Notepad' is open, displaying the following Java code:

```
class B1{  
    public static void main(String args[]){  
        StringBuffer sb1=new StringBuffer();  
        sb1.append("Hello");  
        sb1.append(" Value of PI : ");  
        sb1.append(3.14);  
        sb1.append(" this is");  
        sb1.append(true);  
        System.out.println(sb1);  
    }  
}
```

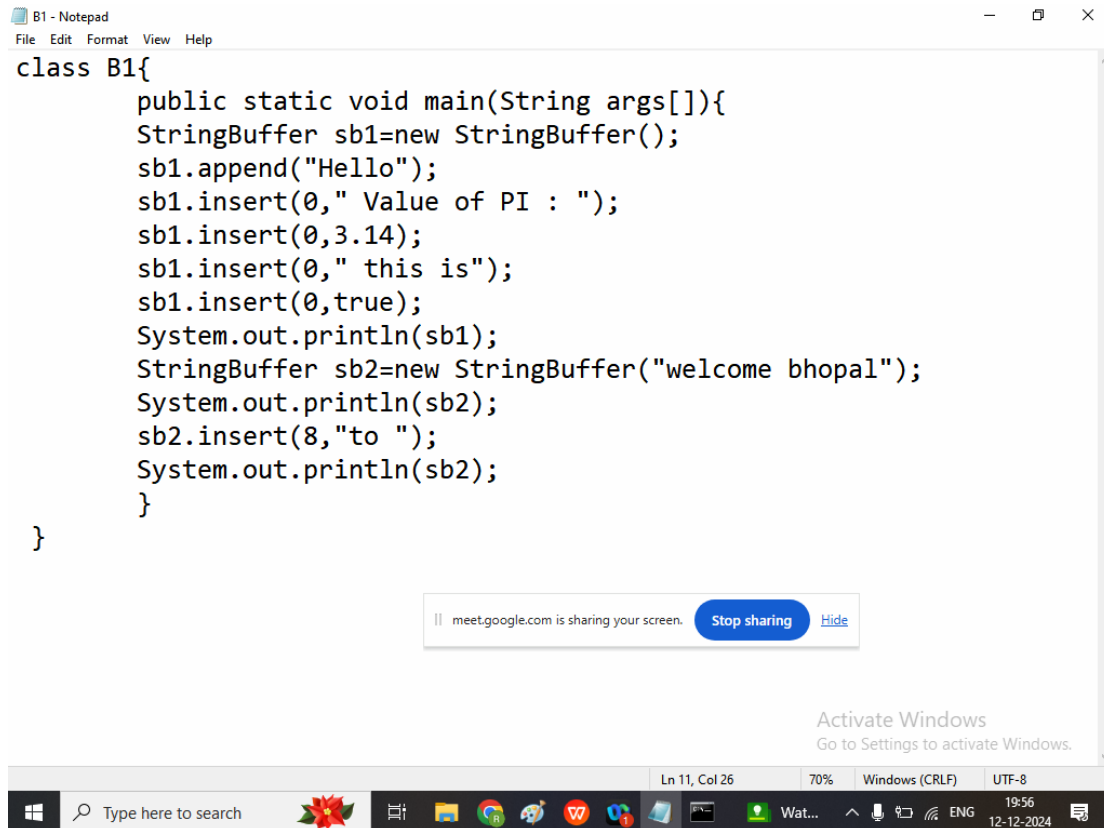
Below the code, there is a notification from 'meet.google.com' stating 'is sharing your screen.' with 'Stop sharing' and 'Hide' buttons. At the bottom right of the Notepad window, there is a 'Activate Windows' watermark with the text 'Go to Settings to activate Windows.'

The Windows taskbar is visible at the bottom, showing the Start button, a search bar, and several application icons including File Explorer, Google Chrome, and Microsoft Word. The system tray on the right shows the date and time as '19:52 12-12-2024'.

3. Insert(): It is used to add data into SB object in any position

Syntax:

insert(position,data);



```
class B1{
    public static void main(String args[]){
        StringBuffer sb1=new StringBuffer();
        sb1.append("Hello");
        sb1.insert(0," Value of PI : ");
        sb1.insert(0,3.14);
        sb1.insert(0," this is");
        sb1.insert(0,true);
        System.out.println(sb1);
        StringBuffer sb2=new StringBuffer("welcome bhopal");
        System.out.println(sb2);
        sb2.insert(8,"to ");
        System.out.println(sb2);
    }
}
```

Q1.Explain StringBuffer class  
deleteCharAt() method?

Ans: It is used to delete character by  
character from the given SB object?

Example:

Write a java program to delete first and  
middle and last character of String

Enter Any String: welcome

Result After Delete: elom

```
class B1{
    public static void main(String args[]){
        StringBuffer sb1=new
StringBuffer("welcome");
        System.out.println(sb1);
        sb1.deleteCharAt(sb1.length()/2);
        sb1.deleteCharAt(0);
        sb1.deleteCharAt(sb1.length()-1);
        System.out.println(sb1.toString());

    }
}
```

---