

Strings

▼ Intro

Strings can be defined as group of characters

- In JAVA Strings are immutable
- `String str = "abcd"`
- `.length()` is used to find the length of a string . It is used as a function in case of strings
- Concatenation : can be done using `+`
- `.charAt(0)` : is used to find the character at any index value. To print the complete string we can use `.charAt(i)` in a loop

▼ Palindrome

Check if the string is a palindrome

- A string which is same to read no matter if it is started reading from the front or from the back e.g. noon , madam
 - Logic : Compare First character to the last , then the second char to the second last char and similarly all char upto the middle of the string
 - Loop would be performed upto half of the length of the loop
 - comparing `str(i) = str (n - i - 1)`
 - n is the length of the string starting form 0
 - Time Complexity : Linear $O (n)$

▼ Shortest Path

Given a route containing 4 directions (E, W, N, S), find the shortest path to reach destination.

"WNEENESENNN"

Question 2

Given a route containing 4 directions (E, W, N, S), find the **shortest** path to reach destination.

"WNEENESENNN"

Formula for distance:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Calculation:

$$\sqrt{(3-0)^2 + (4-0)^2} = \sqrt{9+16} = \sqrt{25} = 5$$

Direction mapping:

N	↑	y+1
S	↓	y-1
W	←	x-1
E	→	x+1

APNA COLLEGE

- We move one unit based on the Letter of the String denoting the direction
 - N = y + 1
 - S = y - 1
 - W = x - 1
 - E = x + 1
- Formula to find the Shortest Path :

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Time Complexity : Linear O (n)

▼ Comparing Strings

- Concept of Interning in JAVA :
 - if a string "hello" already stored in for e.g. by a name Str1 , then
 - if we create another string "hello" as Str2 ,

- the new string would not take up new space rather it would just point towards Str1
- To create a new string use new key word as Str3 = new string ("Hello")

```
if(s1.equals(s3)) {
    System.out.println("Strings are equal");
} else {
    System.out.println("Strings are not equal");
}
```

.equal function only checks the values

▼ Sub Strings

```
public static void main(String args[])
// Substring
System.out.println(str.substring (0,5))
```

▼ Largest String

Lexographically arranging order

Question 3

For a given set of Strings, print the **largest** string.

"apple", "mango", "banana"

str1.compareTo(str2)

0 : equal

< 0 : -ve st1 < st2


> 0 : +ve st1 > st2

}

compare to ignore case ← A'a'

α

APNA
COLLEGE



```
public static void main(String args[]) {
    String fruits [] = {"apple", "mango", "banana"};\
```

```
String largest = fruits [0] ;
for(int i=1; i<fruits.length; i++) {
    if (largest.compareTo(fruits[i] < 0) {
        largest = fruits [i];
    }
}
System.out.println(largest);
```

▼ String Builders

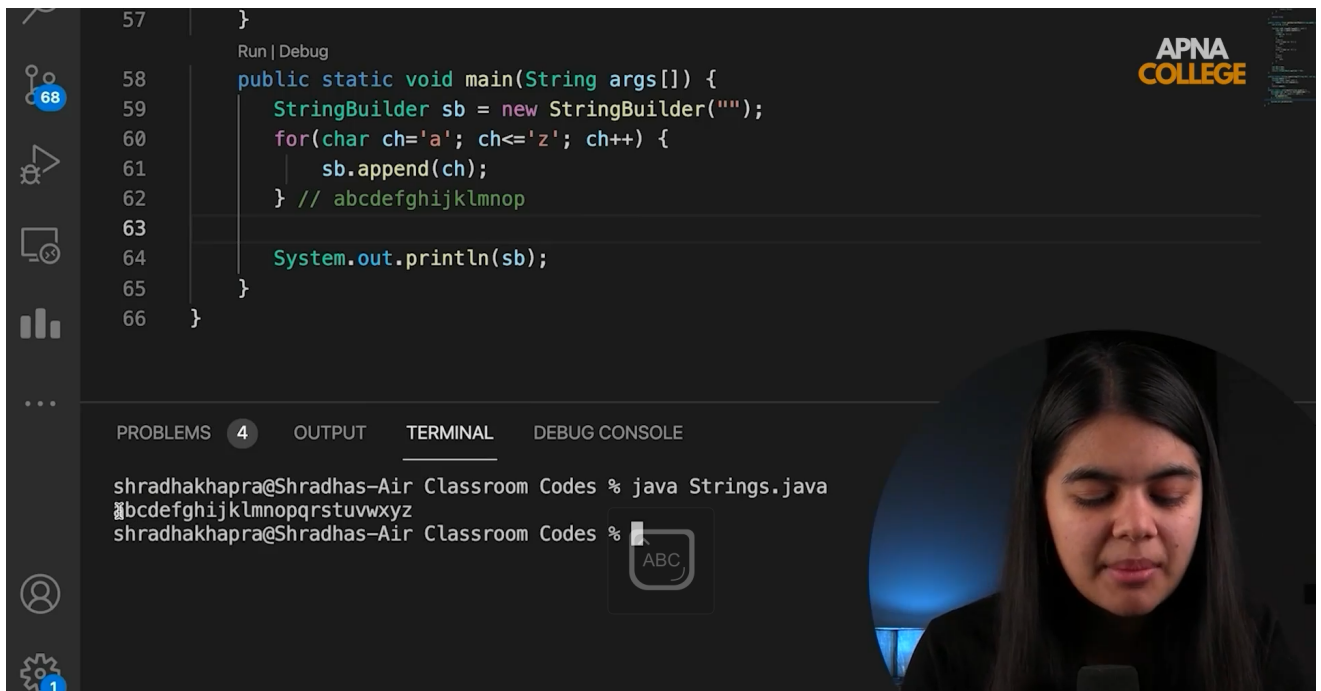
String Builders are used because the traditional way of adding some chars to an already existing String is a very inefficient way which could even add up to a time complexity of $O(n^2)$

- Therefore we use string builders to add char by using .append(Ch)
- $O(n)$

```
public static void main(String args[]) {

    StringBuilder sb = new StringBuilder("");
    for(char ch='a'; ch <='z' ; ch++ ){
        sb.append(ch);
        // abcdefghijklmnop

    System.out.println(sb) ;
}
```



The screenshot shows an IDE with a Java file named `Strings.java`. The code defines a `main` method that uses a `StringBuilder` to generate the alphabet string `abcdefghijklmnopqrstuvwxyz`. The terminal output shows the command `java Strings.java` being executed, resulting in the printed string `abcdefghijklmnopqrstuvwxyz`. A circular inset in the bottom right corner shows a person looking at the screen.

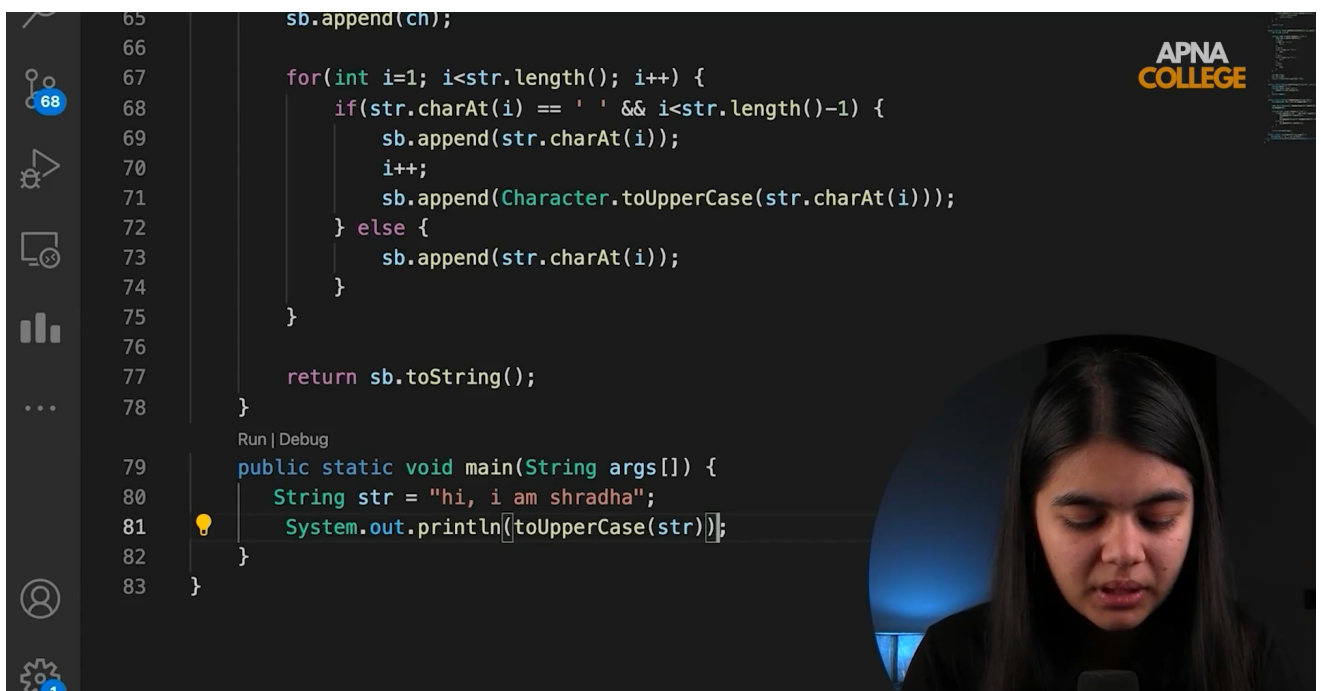
```
57 }
58 Run | Debug
59 public static void main(String args[]) {
60     StringBuilder sb = new StringBuilder("");
61     for(char ch='a'; ch<='z'; ch++) {
62         sb.append(ch);
63     } // abcdefghijklmnop
64
65     System.out.println(sb);
66 }
```

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

shradhakhapra@Shradhas-Air Classroom Codes % java Strings.java
abcdefghijklmnopqrstuvwxyz
shradhakhapra@Shradhas-Air Classroom Codes %

$O(n^2)$

▼ Convert each first letter to Uppercase



The screenshot shows the same IDE with updated Java code. The `main` method now uses the `toUpperCase` method to convert the first letter of each word in the string `"hi, i am shradha"` to uppercase. The terminal output shows the command `java Strings.java` being executed, resulting in the printed string `Hi, I Am Shradha`. A circular inset in the bottom right corner shows a person looking at the screen.

```
65 sb.append(ch);
66
67 for(int i=1; i<str.length(); i++) {
68     if(str.charAt(i) == ' ' && i<str.length()-1) {
69         sb.append(str.charAt(i));
70         i++;
71         sb.append(Character.toUpperCase(str.charAt(i)));
72     } else {
73         sb.append(str.charAt(i));
74     }
75 }
76
77 return sb.toString();
78 }
79 Run | Debug
80 public static void main(String args[]) {
81     String str = "hi, i am shradha";
82     System.out.println(toUpperCase(str));
83 }
```

▼ String Compression

Question 5



String Compression

"aaabbccdd" "a3b2c3d2"

"aaaaabbbdd" → "a4b3d2"

"abc" → "abc"
→ "a1b1c1"

- Initializing a count variable

```
80 public static String compress(String str) {
81     String newStr = "";
82     //aaabc
83     for(int i=0; i<str.length(); i++) {
84         Integer count = 1;
85         while(i<str.length()-1 && str.charAt(i) == str.charAt(i+1)) {
86             count++;
87             i++;
88         }
89         newStr += str.charAt(i);
90         if(count > 1) {
91             newStr += count.toString();
92         }
93     }
94
95     return newStr;
96 }
```

PROBLEMS 5 OUTPUT TERMINAL DEBUG CONSOLE

```
shradhakhapra@Shradhas-Air Classroom Codes % java Strings.java
a3b2c3d2
shradhakhapra@Shradhas-Air Classroom Codes %
```

```
86         count++;
87         i++;
88     }
89     newStr += str.charAt(i);
90     if(count > 1) {
91         newStr += count.toString();
92     }
93 }
94
95 return newStr;
96 }
```

Run | Debug

```
97 public static void main(String args[]) {
```

PROBLEMS 5 OUTPUT TERMINAL DEBUG CONSOLE

```
shradhakhapra@Shradhas-Air Classroom Codes % java Strings.java
a3b2c3d2
shradhakhapra@Shradhas-Air Classroom Codes %
```