## Today's agenda

- ↳ Char and String
- ↳ ASCII
- ↳ Problems

## Characters:

a) Alphabet

↳ lowercase (a-z)

↳ uppercase (A-Z)

b) Special Character

↳ @, #, !, ?, $, etc.

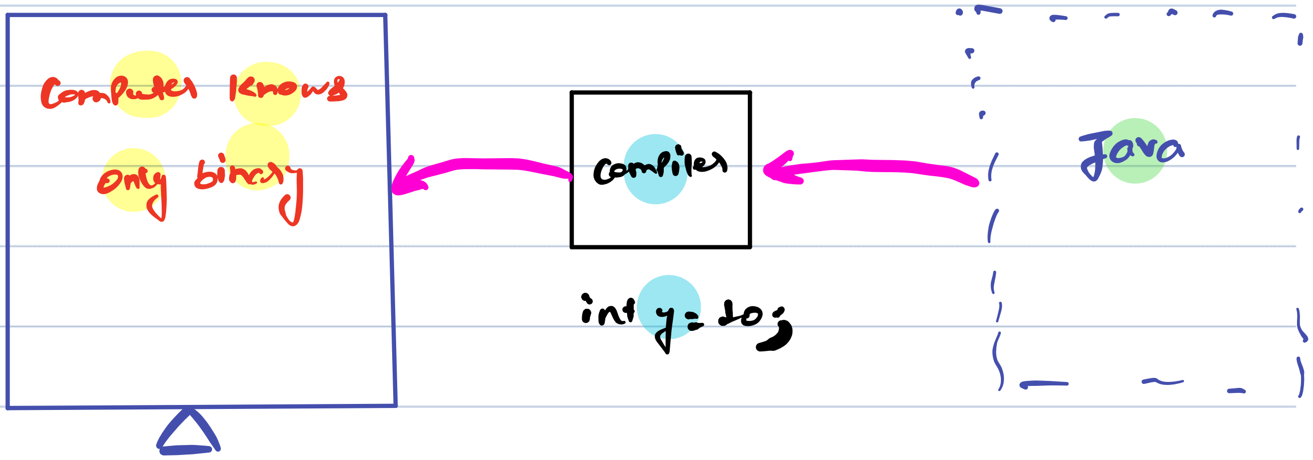
c) Number

↳ 0, 1, 2, . . — 9

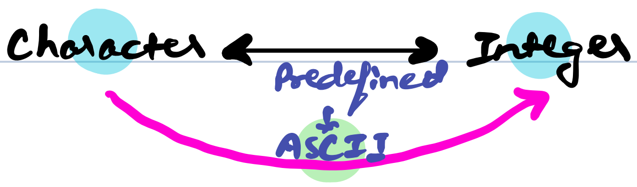
**Syntax:**

type → char, name → ch

↳ char ch = 'A';



decimal → binary ?



ASCII → 256 Chars

int n=3;
char ch= '3';

'A' = 65
'B' = 66
'C' = 67
'Z' = 90

'a' = 97
'b' = 98
'c' = 99
'z' = 122

'0' = 48
'1' = 49
'2' = 50
'3' = 51
'9' = 57

* Char Rules

↳ 1. you Can do mathematical operation on char, answer will be integer.

ex: 'A' + 'B' → 131

$\underset{65}{\downarrow}$ $\underset{66}{\downarrow}$

Typecast (betⁿ Integer and character)

↳ Char to int : implicit

```
int x = 'c';
s.o.p(x) → 99
```

↳ int to char : Complicated{explicit} → few case, implicit
↳ few cases, explicit
⇓
always do explicit

```
Char ch = (char) 65;
```

## Quiz 1:

```
Char ch1 = 'B';
 S.o.p (ch1);   → B
```

## Quiz2:

```
int x= 'A';
  x = x+2;
  S.o.p (x);   → 67
```

67
$\boxed{\cancel{65}}$
x

## Quiz 3:

```
Char ch3= 'xyz';  → error
 S.o.p (ch3);
```

## Quiz 4:

```
Char ch2 =   66;
 S.o.p (ch2);   → 'B'
```

→ even though this code is implicit, you do it explicitly.

## Quiz 5:

$\boxed{'A'}$

ch

```
Char ch4 = 'A';
                 → 68
ch4 = ch4 + 3;  → explicit conversion
                 Error
s.o.p(ch4);
```

## Quiz 6:

```
Char ch5 = 'A';
```

$\boxed{'A'}$

Ch

```
     65
if (ch5 >= 90){
      s.o.p("greater");
}
else{
      s.o.p("smaller");      → Smaller
}
```

## //Strings

↳ Collection of Characters

Syntax:                         ↙type     ↙name                      ↙int[] arr = {1,2,3}

↳ String st = "Algo Prep"

St

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| A | l | g | o | _ | P | r | e | P |

↳ S.o.p (st.charAt(6)); → r

↳ St.charAt(6) = 'z' → error

↳ In String, you can never change a character. {atleast directly}

## //idea

Step1: Convert the String into char[] → syntax??

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|

char[] s = {'A', 'l', 'g', 'o', _, 'P', 'r', 'e', P}

Step2: ↳ s[6] = 'z';

Step3: Convert array back to String → syntax??

* Substring: Any continuous part of String.

ex: String st = "AlgoPrep";

↳ "Al" ✓✓ ↳ "go Pre" ✓✓

↳ "A" ✓✓ ↳ "perP" ××

↳ "Alo" ××

0 1 2 3 4 5 6 7

String st = "A l g [o P r] e p"

↓

↳ St. SubString (3, 6); → "oPr"

↳ Substring from sp to ep-1

St.substring (3,1) → error

St.Substring (1,2) → l

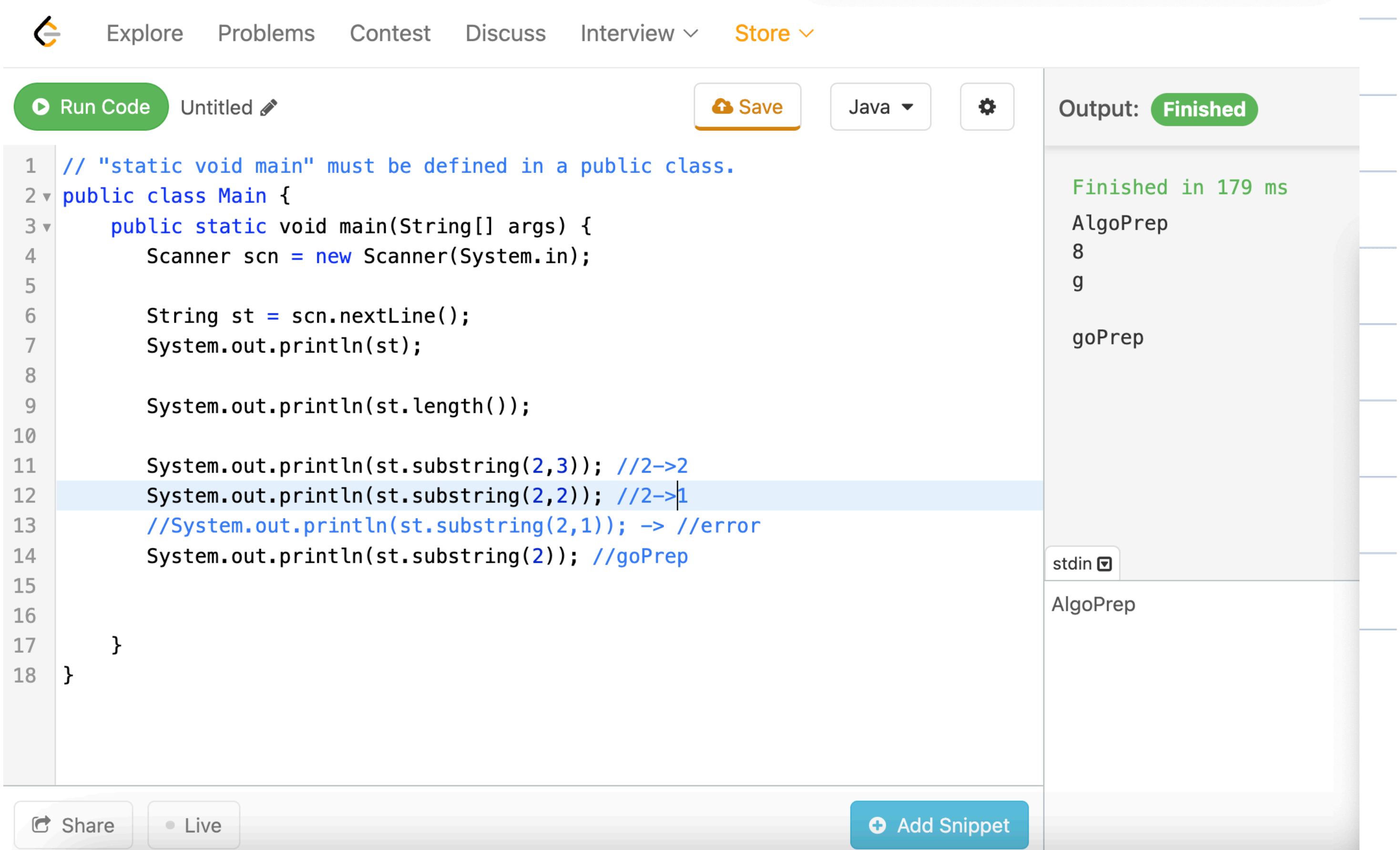
St.substring (1,1) → ""

↳ St. SubString (0, 5) → "AlgoP"

↳ St.SubString (5, 8) → "rep"

↳ St. SubString (5,9) → "error"

Break till 10:40 pm



```java
// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);

       String st = scn.nextLine();
       System.out.println(st);

       System.out.println(st.length());

       System.out.println(st.substring(2,3)); //2->2
       System.out.println(st.substring(2,2)); //2->1
       //System.out.println(st.substring(2,1)); -> //error
       System.out.println(st.substring(2)); //goPrep


    }
}
```

```
Finished in 179 ms
AlgoPrep
8
g

goPrep
```

stdin

```
AlgoPrep
```

## Q) Toggle Characters

↳ Given a Char[] which Contains only small and Capital letters, toggle them.

       0 1 2 3 4 5 6 7
ex: AlGoPreP → aLgOpREp

'A' : 65 $\underset{-32}{\overset{+32}{\rightleftarrows}}$ 'a' : 97

'B' : 66 $\underset{-32}{\overset{+32}{\rightleftarrows}}$ 'b' : 98

⋮

'Z' : 90 $\underset{-32}{\overset{+32}{\rightleftarrows}}$ 'z' : 122

↳ uppercase to lowercase : +32

↳ lowercase to uppercase : -32

//Psuedo Code

inplace
(without using
space.

T.C: O(N)

S.C: O(1)

```
void toggle ( char [] ch) {
        int n: ch.length;
    for(int i=0; i< ch.length ; i++) {
        if (ch[i] >= 'A' && ch[i] <= 'Z'){
            ch[i] = (char)ch[i] +32;
        }
        else { //lowercase
            ch[i] = (char)ch[i] - 32;
        }
    }
}
```

## Q) Reverse the given String

↳ Given a String str, reverse the String.

ex: algoprep → perpogla

T.C: O(N)

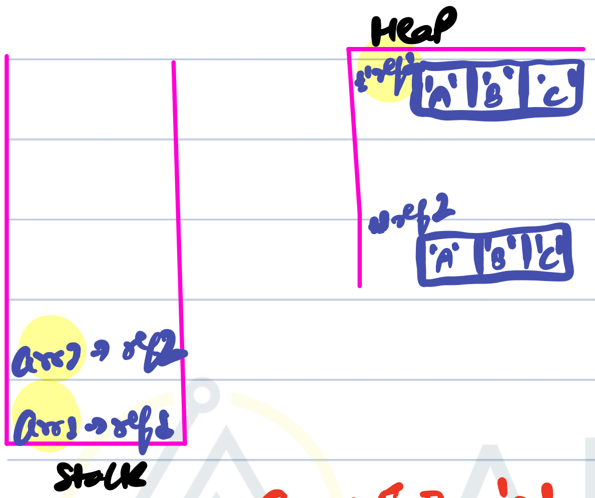
S.C: O(N)

```
String reverseString (String St){

    Char[] ch = St.toCharArray();

        int sP=0;
        int eP= ch.length-1;
        while (sP < eP){
            Char temP= ch[sP];
                ch[sP] = ch[eP];
                  ch[eP] = temP;
        }     sP++; eP--;

        return "".valueOf(Ch);
}
```

→ Character is Stored in Stack.

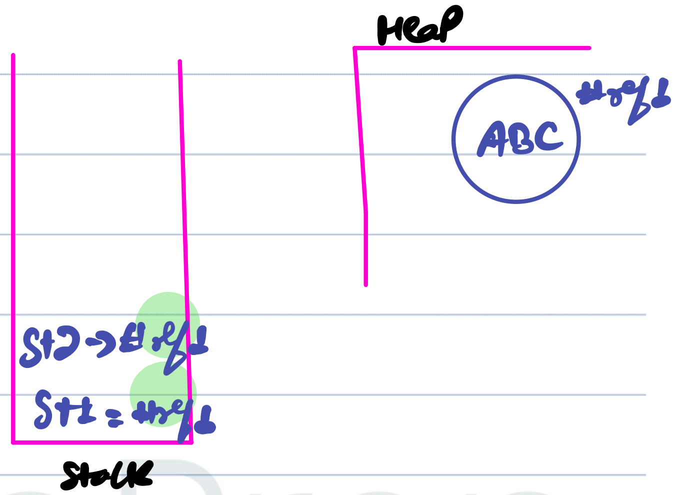
Internal working

```
                    0    1    2
char[] arr1 = {'A', 'B', 'C'}
char [] arr2 = {'A','B','C'}
```



```
arr1[0] = '2';
S.o.p (arr2[0]);
   ↳ 'A'
```

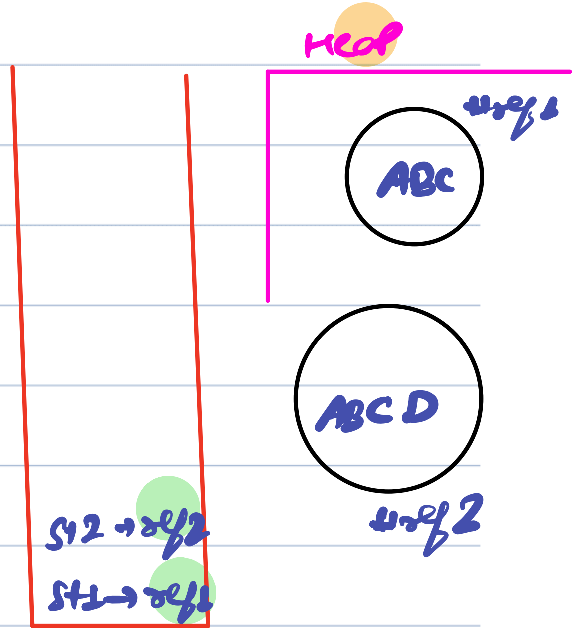
```
                      0 1 2
String St1 = "ABC";
String St2 = "ABC";
```



```
St1.~~charAt(0)~~ = '2'
S.o.p (St2.charAt(0)); → 2
```

↳ Strings are immutable

```
String St1= "ABC";                → O(N)
String St2 = St1 + "D";   → valid

      String St= "Hello";
      int n = St.length();
   for (int i=0; i<n; i++) {
         St = St + "e"; →O(n)
   }

      ↳ T.C : O(n²)
```



Tried solving space problem with String

↓

String must be immutable

↓

if you do any change in String that will create a new String with changes. {which consumes a lot space }

→ To add char to String we use stringbuilder.