Today's agenda

- Char and String
- ASCII
- Problems

Characters:

a) Alphabet

↳ a-z (lowercase)

↳ A-Z (uppercase)

b) Special Characters

↳ @, #, *, ?, ! etc.

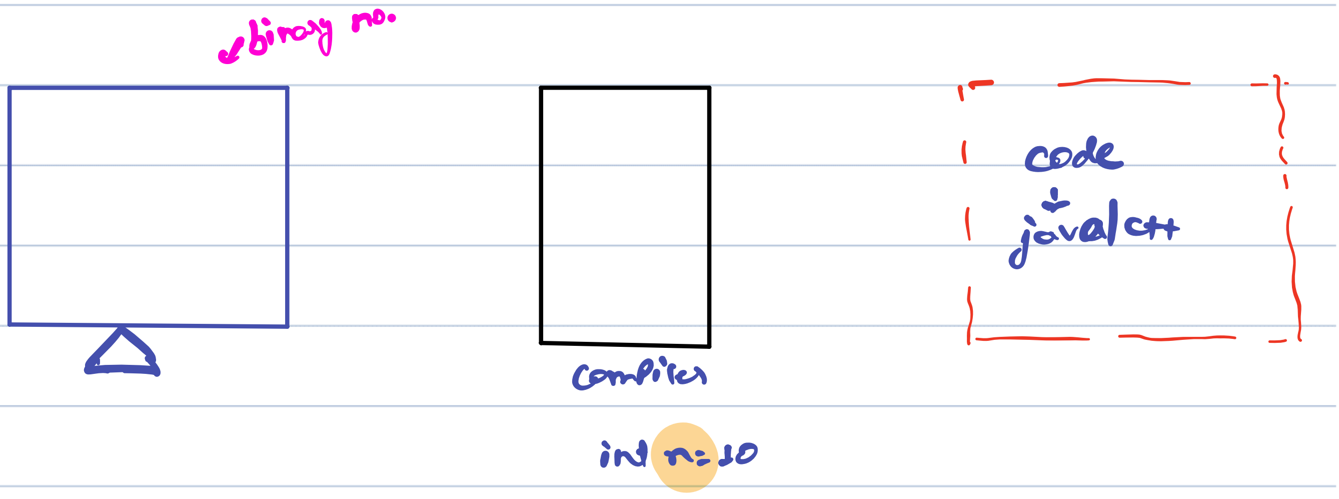
c) Number

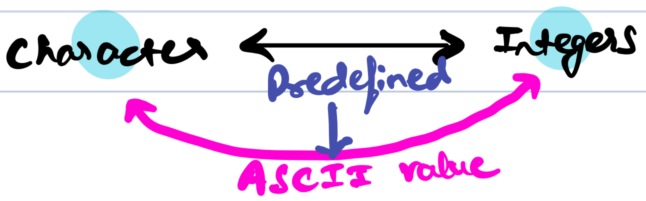
↳ 0, 1, 2. . . _ 9

Syntax:

type name character

```
char ch = 'A';
```





Char ch = 'c';

ASCII → 256

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

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

'0' == 48
'1' == 49
'2' == 50
⋮
'9' == 57

* char 8ules

↳1. you can do mathematical operations on characters, answer will be integer.

en: s.o.p('A' + 'B'); → 131

$$\underset{65}{\uparrow}\qquad\underset{66}{\uparrow}$$

Typecast

↳2. Char to int : implicit

int n= 'C';

s.o.p(n); → 67

↳3. int to char : complicated

→ few cases, implicit

↳ few cases, explicit

⇓

do explicit always

Char ch = (char)65; → 'A'

Quiz 4: → (implicit)

Char ch2 : 66;

s.o.p(ch2); → 'B'

Quiz 5: → (explicit)

Char ch4 : 'A';

ch4 : (char)(ch4 + 3); → errors

65 = 68

s.o.p(ch4);

$\boxed{'A'}$

ch

Quiz 1:

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

Quiz2:

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

67

$\boxed{\cancel{65}}$

x

Quiz 3:

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

Quiz 4:

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

Quiz 5:

```
                                             'A'
                                             ch
char ch4 = 'A';
                     65  = 68
ch4 = (char)(ch4 + 3);         → error
s.o.p(ch4);
```

Quiz 6:

```
char ch5 = 'A';

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

Break till 10:20 Pm

int[] arr = {1,2,3] → Collection of variable

// Strings

↳ Collection of Characters

Syntax:

↳ String st = "AlgoPreP";

st

0	1	2	3	4	5	6	7
A	l	g	o	P	r	e	P

↳ s.o.p (st.charAt(2)); → g

↳ st.charAt(2) = 'z';

↳ In String you can't change Characters directly. (update not allowed)

0	1	2	3	4	5	6	7

→ char[] st = {'A', 'L', 'G', 'O', 'P', 'R', 'E', 'P'};

↳ st[2] = 'z';

* SubString → Any Continuous Part of String.

↳ String st = "AlgoPreP";

↳ Al ✓ ↳ goP ✓

↳ A ✓ ↳ erP ✗✗

↳ AlP ✗✗

0 1 2 3 4 5 6 7

String st = "AlgoPreP";

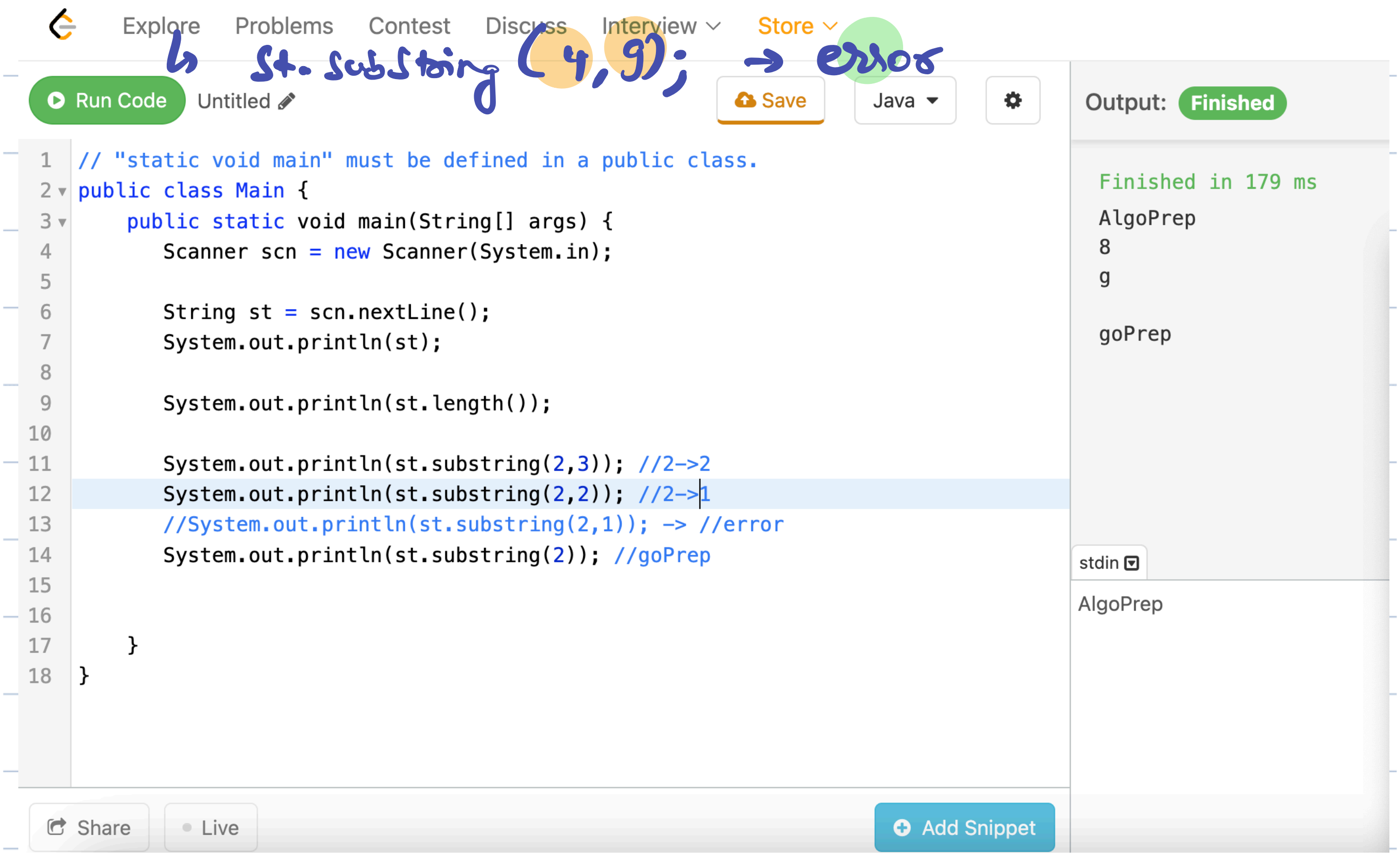
inc enc

↳ st.substring(1, 4);

↳ st.substring(0,6); → AlgoPr

↳ st.substring(5,8); → reP

↳ st.substring(4,9); → error



```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.

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

```
void   toggle ( char[] ch){

    for (int i=0; i<ch.length; i++){

        if ( ch[i] >=65 && ch[i]<=90){

            ch[i] = (char)(ch[i] + 32)
        }
        else{

            ch[i] =(char)(ch[i] -32)

        }
    }
}
```

T.C: O(N)

S.C: O(1)

Q) Reverse the given String

↳ Given a String str, reverse the String.

```
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);
}
```

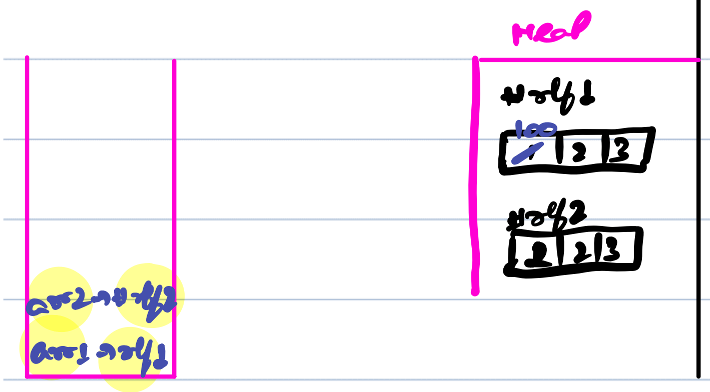
T.C: O(N)

S.C: O(N)

arr

int [] arr1 = {1, 2, 3}

int [] arr2 = {1, 2, 3}

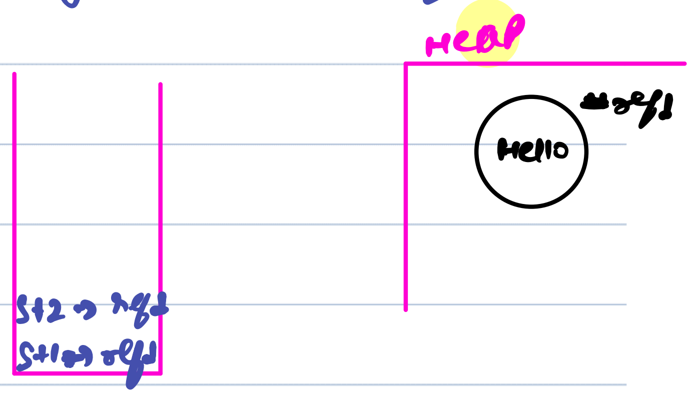


arr1[0] = 100;

String

String str1 = "Hello";

String str2 = "Hello";

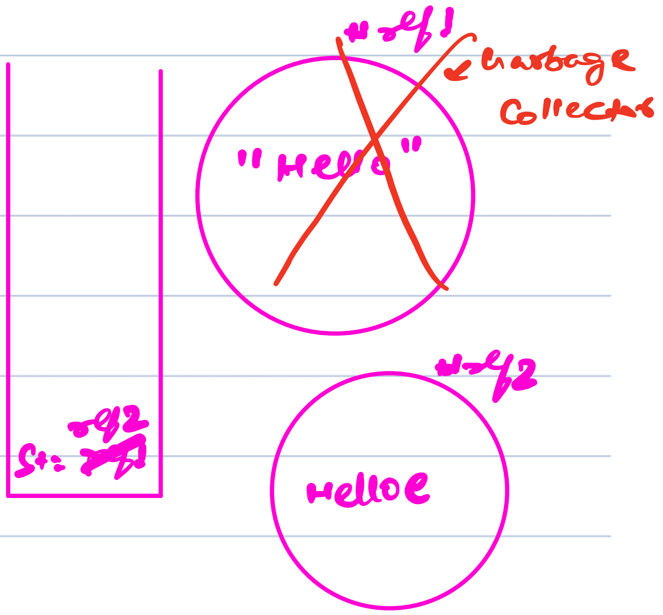


str1.~~charAt~~(0) = ~~'x'~~;

↳ strings are immutable

String st = "Hello";

→ st = st + "e"; → O(n)



String st = "Hello";

```
→ for (int i=0; i<N; i++) {     → st.length()
      st = st + "e"; → O(N)
  }
```

↳ $O(N^2)$ → T.C

→ Arraylist → dynamic array

↳ ArrayList <Integer> al = new ArrayList<>();

al

0	1	2	3
10	20	30	10

add at last

```
al.add(10);
al.add(20);
al.add(30);
al.add(10);
```

```
s.o.p(al.size());
```

↓

3

```
s.o.p(al.get(2))
```

```
al.set(2, 100);
```