Today's agenda

- Char and String
- ASCII
- Problems

int[]: Collection of integers

String: Collection of Characters

ex: "AlgoPrep"

Characters:

a) Alphabet

↳ a-z (lowercase)

↳ A-Z (uppercase)

b) Special characters

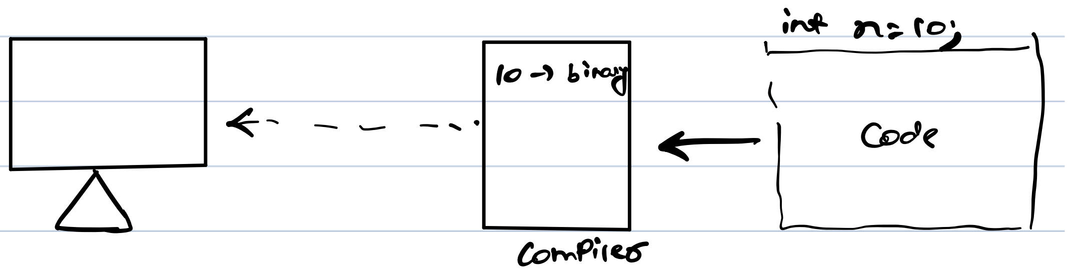
↳ @ # $? ! etc.

c) Numbers

↳ 0, 1, 2 9

Syntax:

```
 type    name   character
 char    ch  =  'A';
```



Characters $\xrightarrow{\text{ASCII values}}$ number → binary number

Predefined

Char ch = 'B'; ⇒ 'B' = 66

ASCII → 256 Characters

'A' : 65	'a' : 97	'0' : 48
'B' : 66	'b' : 98	'1' : 49
'C' : 67	'c' : 99	⋮
⋮	⋮	'9' : 57
'Z' : 90	'z' : 122	'10' : Not a Character

* char Rules

1. you can do any mathematical operation on character & the answer will be integer.

$$s.o.p(\overset{65}{\cancel{'A'}} + \overset{66}{\cancel{'B'}}); \rightarrow 131$$

typecast

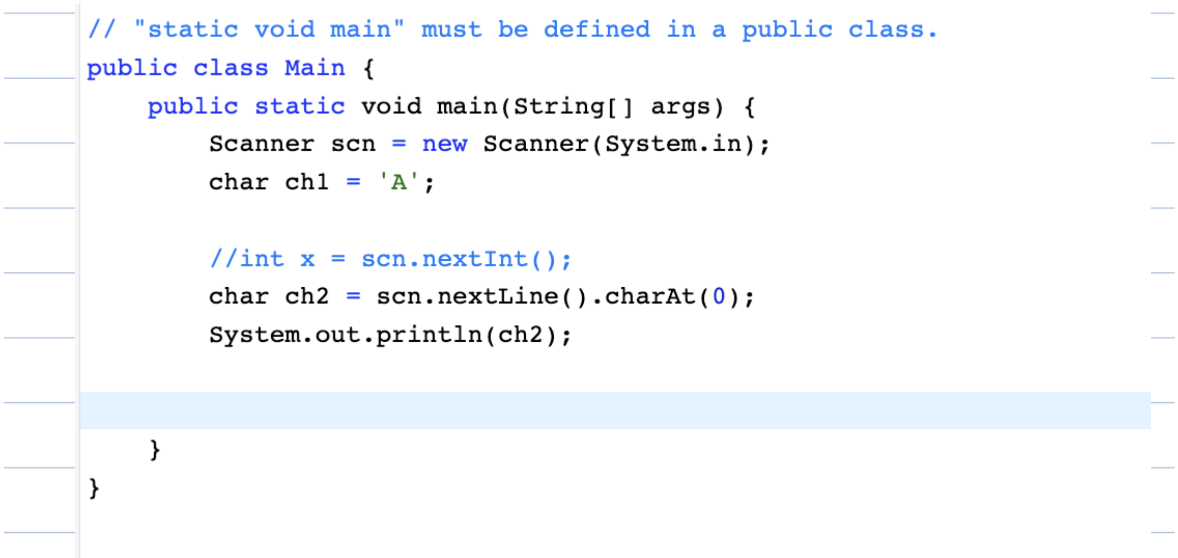
2. char to int : implicit

ex: int x = 'c';

3. int to char : complicated → in few cases it will be implicit
↳ in few cases it should be explicit

ex: char ch = 65;

|| do explicitly always.



```
// "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);
        char ch1 = 'A';

        //int x = scn.nextInt();
        char ch2 = scn.nextLine().charAt(0);
        System.out.println(ch2);


    }
}
```

Quiz 1:

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

Quiz2:

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

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

Quiz 3:

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

Quiz 4:

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

Quiz 5:

```
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");
}                                   → Smaller
else{
        s.o.p("smaller");
}
```

// Strings

↳ Collection of characters.

Syntax:

↳ String st = "AlgoPrep";

String st = "A"; → this works

st

0	1	2	3	4	5	6	7
A	L	G	O	P	R	E	P

→ S.o.p (st.charAt(3)); → O

→ st.~~ch~~arAt(3) = ~~'x'~~'z'; → error

↳ In String you can't change characters directly. (no update).

↳ Strings are immutable.

→ **SubString** → Any continuous part of String

String st = "AlgoPrep";

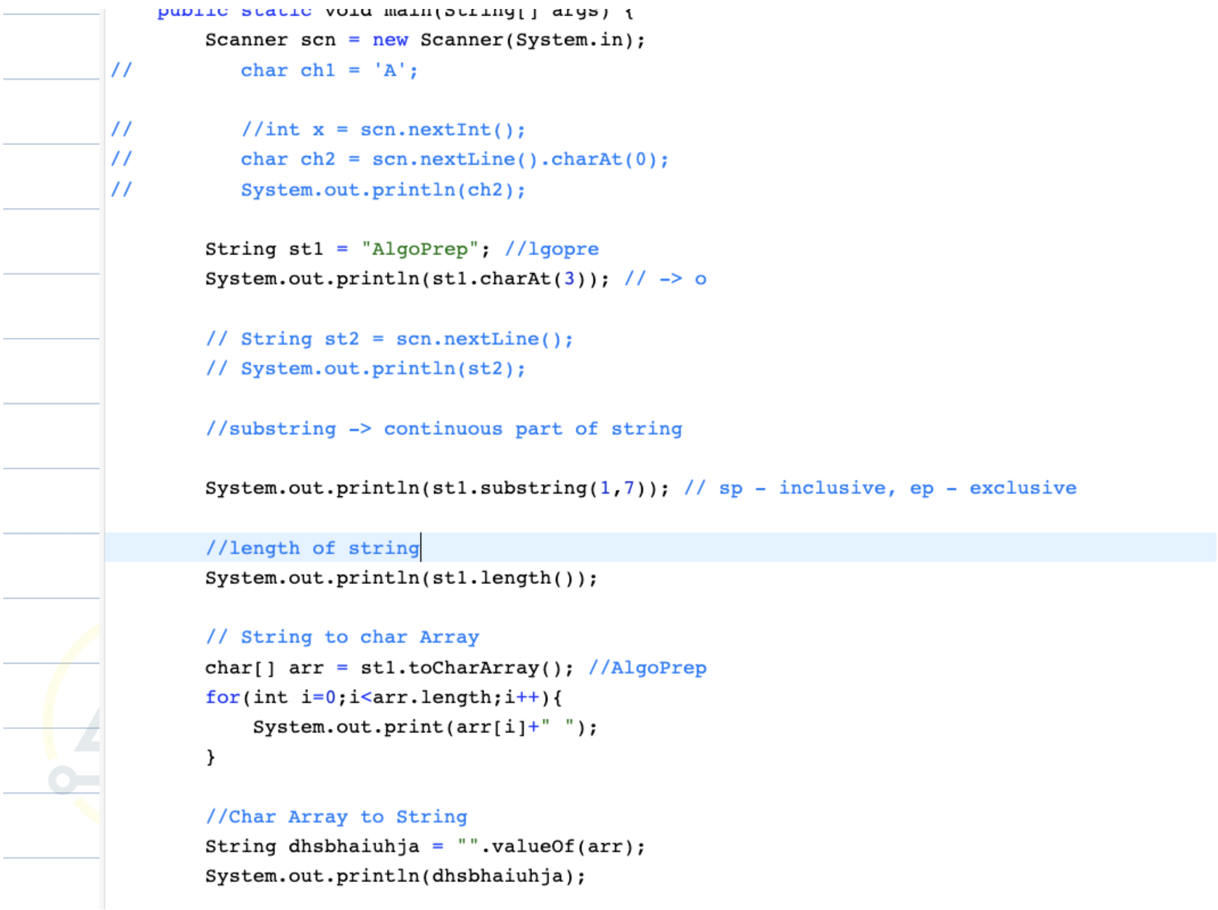
↳ Algo ✓ ↳ goPrepd ✗

↳ A ✓

↳ Prep ✓

↳ AlgP ✗✗

↳ AlgoPrep ✓



```
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
//         char ch1 = 'A';

//         //int x = scn.nextInt();
//         char ch2 = scn.nextLine().charAt(0);
//         System.out.println(ch2);

        String st1 = "AlgoPrep"; //lgopre
        System.out.println(st1.charAt(3)); // -> o

        // String st2 = scn.nextLine();
        // System.out.println(st2);

        //substring -> continuous part of string

        System.out.println(st1.substring(1,7)); // sp - inclusive, ep - exclusive

        //length of string
        System.out.println(st1.length());

        // String to char Array
        char[] arr = st1.toCharArray(); //AlgoPrep
        for(int i=0;i<arr.length;i++){
            System.out.print(arr[i]+" ");
        }

        //Char Array to String
        String dhsbhaiuhja = "".valueOf(arr);
        System.out.println(dhsbhaiuhja);
```

Break till 9:48 PM

Q) Toggle Characters

↳ Given a char[] which contains only small and capital letters, toggle them.

↳ lowercase → uppercase

↳ uppercase → lowercase

ex: AlgoPrEP → aLGOpReP

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

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

'C' : 67 $\overset{+32}{\underset{-32}{\rightleftarrows}}$ 'c' : 99

'Z' : 90

uppercase to lowercase : +32

lowercase to uppercase : -32

→ whenever you want to change char, you think in terms of ascii.

//Psuedo code

```
p s 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{ //lowercase

            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.

//Psuedo code

```
Public static String reverseString (String str ){

    Char[] ch = Str.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)

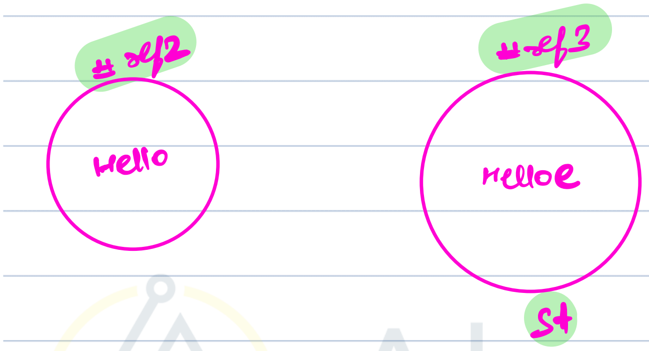
S.C: O(N)

→ String St= "Hello"; → immutable

St= St +"e"; → T.C: O(n)

S.o.p (St); → Helloe

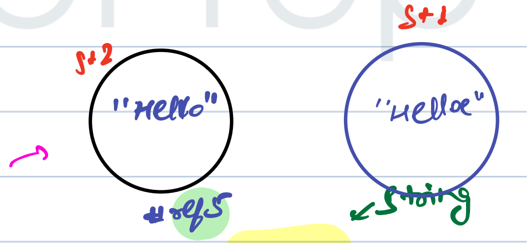
↳ concatenation can happen



→ do not concatenate in String.

→ String St1 = "Hello";

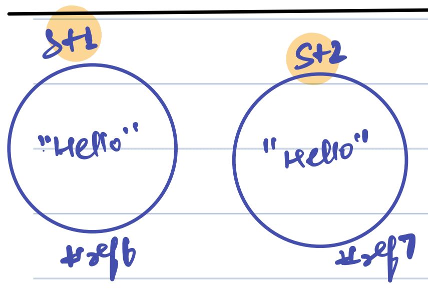
String St2 = "Hello";



idea2

↑

St1 = St1 + "e";



idea1

↗ array

Save Space

↓

Point multiple Same String to same address

↓

What if i change 1 String?

↓

Strings are immutable