ans will always be in high data type.

2 + 2.0 = 4.0

2 + 2.0 = 4.0

'a' + 2 = 97+2 ⇒ 99

'm'

'm' >= 'a' && 'm' <= 'w'

T

result = (char) ('m' + 3)

109 + 3 = 112

char(112)

P

jumping character

```
Scanner s = new Scanner(System.in);
char ch = s.next().charAt(0);

if(ch >= 'a' && ch <= 'w'){
    char result = (char) (ch + 3);
    System.out.println(result);
}
else if(ch >= 'D' && ch <= 'Z'){
    char result = (char) (ch - 3);
    System.out.println(result);
}
else{
    System.out.println("Can't jump");
}
```

"a" "b" "c"

a|P → P

'a' + 'b' + 'c' = abc

97 + 98 + 99

Add if a digit

'0' → 48
'1' → 49
'2' → 50
'3' → 51
'4' → 52
'5' - 53
'6' - 54
'7' - 55
'8' - 56
'9' - 57

char digit → int digit

'3' → 3

51 - 48 → 3

'ch' - '0' = int digit

'9' → 9

'9' - '0' → 57 - 48 = 9 → int

ch - '0'
ch - 48

[char + 100] = int

Explicit
(int) (ch)
(int) (d)

$D'

char ch = 'a'
int i = ('a')
97

```java
Scanner s = new Scanner(System.in);
int ch = s.next().charAt(0);
if(ch >= '0' && ch <= '9'){
    int ans = ch - '0' + 100;
    System.out.println(ans);
}
else{
    System.out.println("This is not a digit");
}
```

char

ch = '5'

ans = '5' - '0' + 100

53 - 48 + 100

5 + 100 = 105 O/P

in-built function

Character.getNumericValue(ch);

int ans = [Character.getNumericValue(ch)] + 100;

[ch - '0']

('5')

5 + 100 = 105

Uppercase.

A → 65

⋮

Z → 90

Lowercase

a → 97

⋮

z → 122

'a' - 32 = 65

97 - 32

A

'A' + 32

65 + 32 = 97

a

66 + 32 = 98

B → b

[Small to captical = -32

Capital to Small = +32]

```
Scanner s =
char ch =
    if(ch >= 'a' && ch <= 'z'){
        char ans = (char) (ch-32);
        Syso(ans);
        //syso((char)(ch - 32));
    }
else if(ch >= 'A' && ch <= 'Z'){
    char ans = (char) (ch+32);
        Syso(ans);
}
```

```java
Scanner s =
    String str = s.nextLine();
    if(str.length() >= 4){
        char ch = str.charAt(3);
        Syso(ch);
    }
    else{
        Syso("Small string");
    }
}
```

str = "Raja"; (0 1 2 3)

4 >= 4 T

ch = a → o/p

str = "abc"

3 >= 4 F

o/p → Small string.

String Concatination

str1 = "abc"

str2 = "def"

str1 + str2

"abc" + "def" ⇒ "abcdef"

str1 = "mansi"

str2 = "sharma"

str1 + str2 + str1

"mansisharmamansi"

str1 = "ab"]+

str2 = "ab"

o/p → abab

Concatenate_Two_Strings

```
Scanner s = new Scanner(System.in);
String str1 = s.nextLine();
String str2 = s.nextLine();
System.out.println(str1 + str2);
```

string concatenate 2

```
Scanner s =
    String s1
    String s2

    if(s1.length() > s2.length()){
        Syso(s2 + s1 + s2);
    }
else{
    Syso(s1 + s2 + s1);
}
```

s1 = " abc "

s2 = " a "

o/p → aabca

Unary Operator.

+ - % a+b

a b

++ --

Increment Decrement

Post a++ a-- a=a+1;

Pre ++a --a

Post → value will get printed first then increment

a=7 a++ 7 → 8

Pre → first increment then printed

a=7; ++a = 8

```java
public static void main(String[] args) {
    int a = 5;
    int b = a++;
    int c = ++a;

    System.out.println(b);
    System.out.println(a);
    System.out.println(c);
    System.out.println(++b);
}
```

```
Output

java -cp /tmp/dFjsu
5
7
7
6

=== Code Execution
```

	a = 5	b	c
b = a++	6	5	-
c = b++	6	6	5
a = ++b	7	7	5
b = ++c	7	6	6
c = ++a	8	6	8
a = b++	6	7	8
b = ++c	6	9	9
c = a++	7	9	6
a = ++c	7	9	7

Loops → used to perform a operation / task repeatedly.

Types. → 4 types

→ for loop
→ while loop
→ do - while loop.
→ for - each

loop → ∞

1 — 5 min 10 min
∞

1 — 10 : +1

play 2 ✓
3 ✓
4 ✓

for loop-

all 3 are optional

```
for(;;){
}
```
⇒ ∞

Syntax

```
for(initialization ; condition ; update){
    // statement
}
```

5 times "Hello"

```
for(int i=1 ; i≤5 ; i++){
    Sysout("Hello");
}
```

o/p →
Hello
Hello
Hello
Hello
Hello

i = 1 ≤ 5 T
i = 2 ≤ 5 T
i = 3 ≤ 5 T
i = 4 ≤ 5 T
i = 5 ≤ 5 T
i = 6 ≤ 5 F

initialization → from where to start 2

condition → where to stop 10

update → how many steps you are ↑/↓ ~ +2
index.