

\Rightarrow Revision

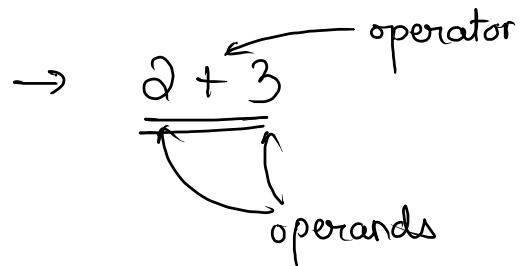
- Programming language & Operators
- Variables
- Conditions and logical operators
- if else
- nested if else
- Switch statement
- Characters and Strings
- ★ → for loops
- ★ → while & do while loop
- ★ → Patterns
- ★ → functions (return statement)
- digit traversal ($\%10$, $/10$) & number theory
- ★★ → Arrays (Printing, finding & searching, storing info, updating)
- ★ → Brute force (Permutation & Combination)
- ★ → Time & Space Complexity

\Rightarrow Operators (specific special symbols to perform some operations)

\rightarrow Types :-

1) Airthematic operators :-

(+ , - , / , * , %)

\rightarrow 
operator
 $\underline{\underline{2+3}}$
operands

2) Logical operator :-

(&& , || , !)

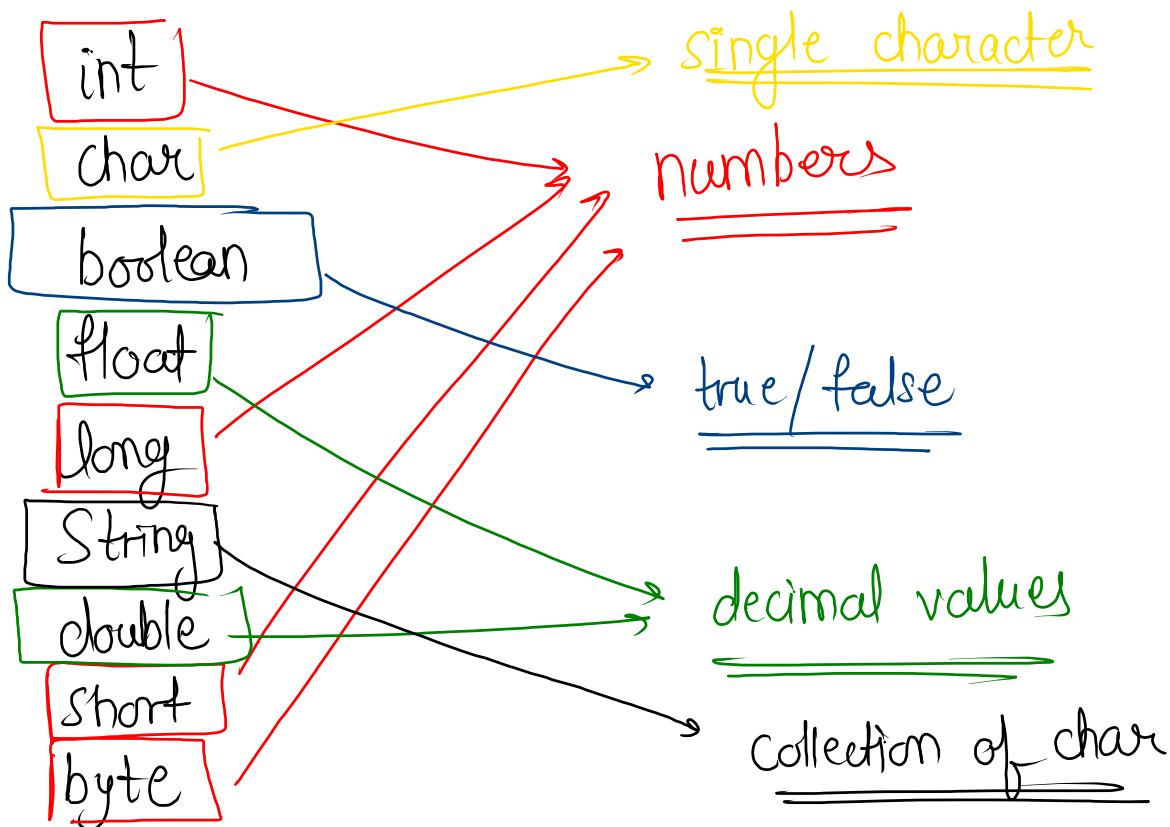
3) Assignment (int a = 5 ;)

4) Relational :- > , < , >= , <= , == , !=

5) Unary Operator :- a++ , a--
 ++a , --a

→ Variables (used to store some value)

Types :- (Primitive)



⇒ Conditions

```
if( conditions ) {  
    // statement  
}
```

```
if( conditions ) {  
    // statement 1  
} else {  
    // statement 2  
}
```

If else ladder

```
if ( c1 ) {  
    //s01  
} else if ( c2 ) {  
    //s02  
} else if ( c3 ) {  
    //s03  
} else {  
    //s04  
}
```

Note:-

- 1) it always works from top to bottom
- 2) only 1 cond" will work at a time
- 3) if cond" is mandatory & everything else is optional

⇒ Switch statement

Syntax :-

```
switch ( condition ) {  
    Case 01:  
        //statement 01;  
        break;  
    Case 02:  
        //statement 02;  
        break;  
    default :  
        //statement 03;  
        break;  
}
```

; → semi colon

: → colon

() → parenthesis

[] → sq. bracket

{ } → curly bracket

Note :-

== can be used for
a single
variable

Area = 50.0
50.00000

20.7
20.70000

Note :-

String.format ("% .5f", variable);

\Rightarrow Nested

```
if ( c1 ) {  
    if ( c2 ) {  
        // s1  
    }  
}
```

3

⇒ Characters 'c'
↳ String :-

Inbuilt functions :-

String str = "Geekster Classes";
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

1) str.length() // 16

2) str.charAt(8) // K, space ''
 ↑
 index

3) str.toUpperCase() // GEEKSTER CLASSES

4) str.toLowerCase() // geekster classes

5) str1 + str2
 ↓ ↓
 abc efg

always create a new string

(because
string is
immutable)

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    char ch = scn.next().charAt(0);

    solve(ch);
}

public static void solve(char ch) {
    if ( ch >= 'a' && ch <= 'z' ) {
        ch = (char)(ch - 32);
    } else if ( ch >= 'A' && ch <= 'Z' ) {
        ch = (char)(ch + 32);
    } else {
        System.out.println("No alphabet");
        return;
    }

    if ( ch == 'a' || ch == 'A' || ch == 'b' || ch == 'B' ) {
        System.out.println("Sorry");
    } else {
        ch = (char)(ch - 2);
        System.out.println(ch);
    }
}
```