Revision

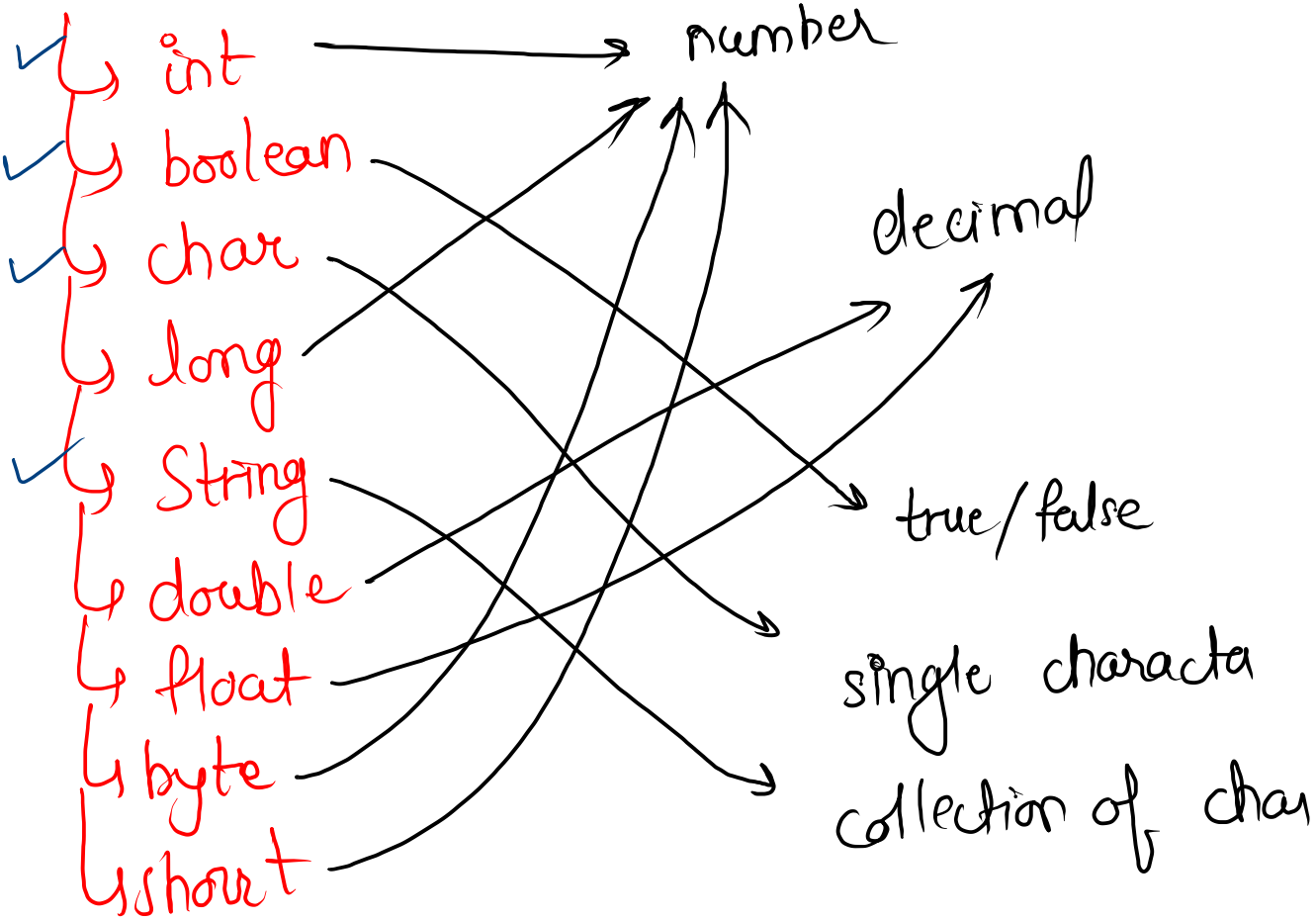
- Programming Language & Operators
- variables
- Conditions and logical operators
- if else cond. (ladder)
- nested if else
- switch statement
- ★ characters and strings
- ★ for loops
- while and do while loop
- ★ Patterns (Template)
- functions (return statement)
- ★ digit traversal (/10, %10) & number theory
- ★★★ array (Printing, searching, storing and upgradation)
- ★ Brute force approach (Permutation & Combination)
- ★ Time Complexity & Space Complexity

⇒ Operator (special symbols to perform same operat^{n})

- arithematic ope. (+, −, *, /, %)
- logical ope. (&&, ||, !)
- assignment (=)
- relational (>, <, >=, <=, !=, ==)
- Unicary (a++, ++a, a--, --a)

⇒ Variables (used to store value)

Premitive Types :-



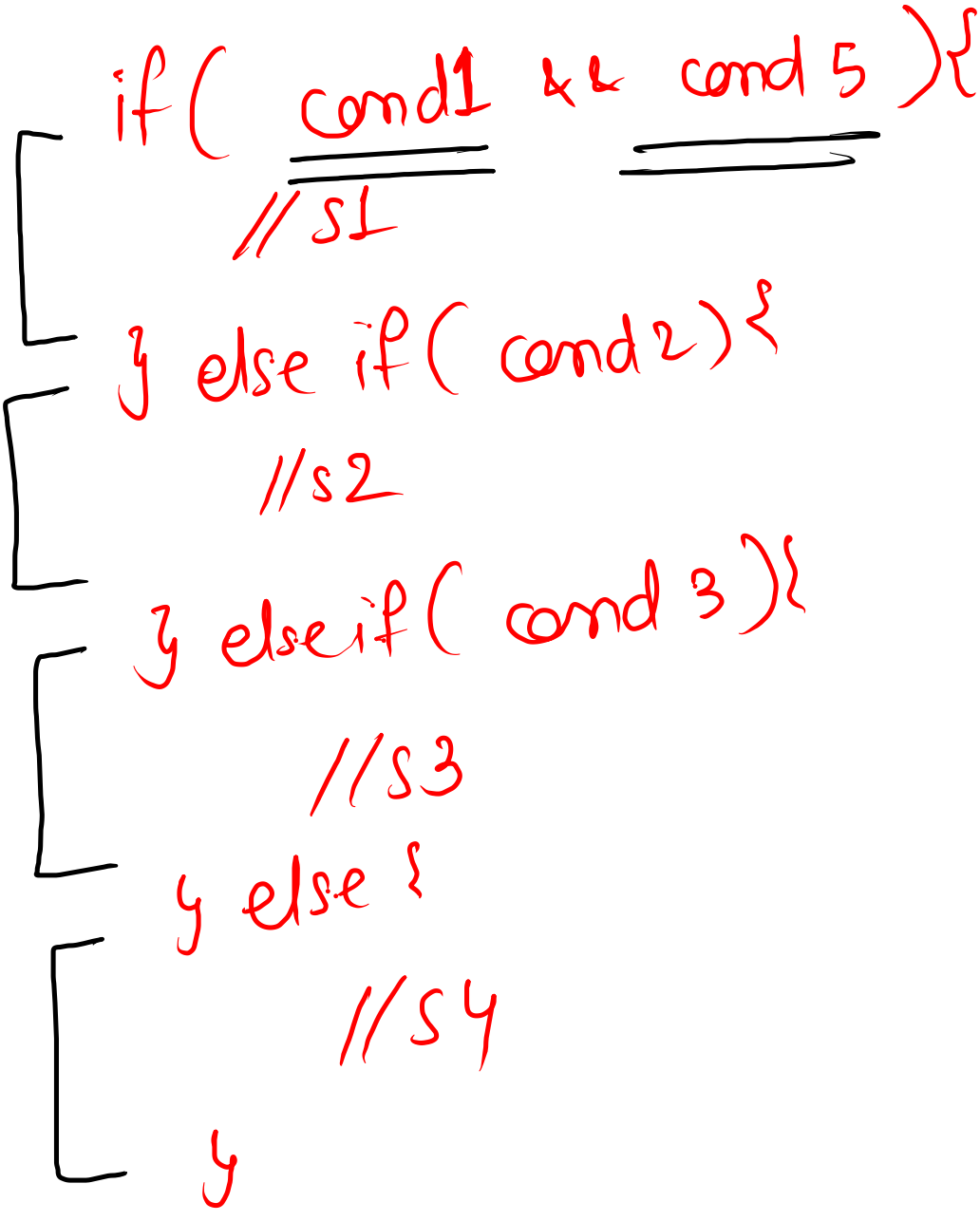
⇒ Conditions

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

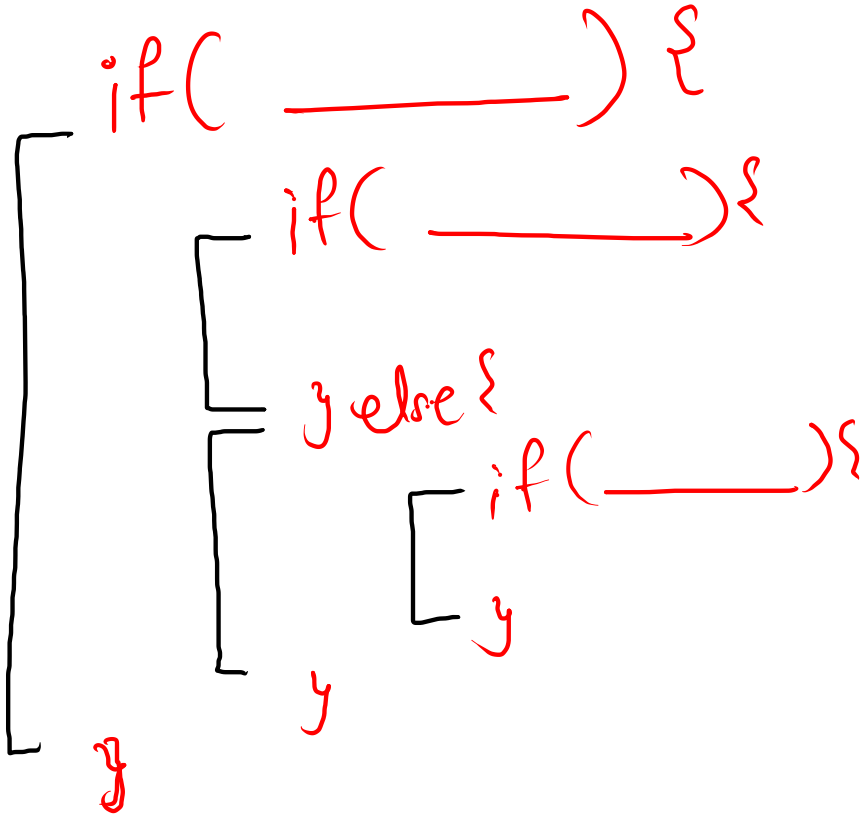
```
if (                    ){

} else if(              ){

} else {

}
```

Notes:-

1) always checks from top to bottom
2) only 1 cond^n can be executed at a time
3) if cond^n is mandatory & all other are optional
4) if else can handle more than 1 cond^n at a time


if(cond1 && cond 5){
//s1
} else if(cond 2){
//s2
} elseif(cond 3){
//s3
} else {
//s4
}

→ Nested if else



Note:-

if cond. is compulsory

⇒ Switch statement

```
switch ( condition ) {
    case val1 :
        //statement 1
        break;
    case val2 :
        //statement 2
        break;
    default :
        // statement 3
        break;
}
```

disadvantage:

only check for one variable only

Note:-

```
int x = 5
int y = 7

int sum = x + y ;
Syso(sum);
```

Ans :- 12.0

12.0000

Inbuilt function

```
public static void main(String[] args) {
    int x = 17;
    int y = 5;
    double divisible = x / y;
    System.out.println( String.format("%.5f", divisible) );
}
```

⇒ Characters and String (gmp)

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(3) ⟶ 'k'
3) str.toUpperCase() ⟶ "GEEKSTER CLASSES"
4) str.toLowerCase() ⟶ "geekster classes"
5) Syso(str + str) ⟶ "Geekster classesGeekster classes"

Gmp

(abc) (efg)

will always create a new string

str1 = str1 + str2

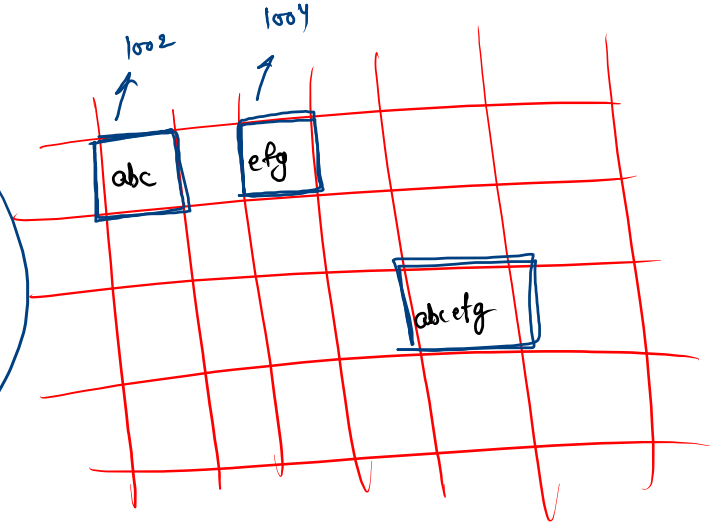
(1002) (1004)

string is immutable

a) 1002

b) 1004

c) random



loops

all are optional

```
for( initialisation ; condition ; upgradation ){
    |
    |
    |
}
```

- initialisation → `int i=0`
- condition → `i<n`
- upgradation → `i++`

initialization

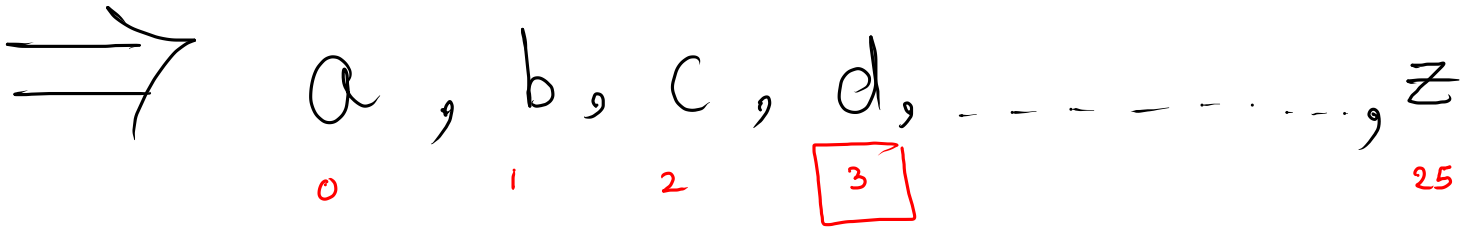
```
while ( condition ) {
        //statement

        upgradation

}
```

⇒

```
char ch = 'a'
Syso ( (char)(ch - 32)) ;   // 65
                            // 'A'
// implict typecasting
```

⇒

```
char ch = 'B'
Syso ( (char)(ch + 32)) ;   // 98
                            // 'b'
// implict typecasting
```




```
char ch = 'd'
int idx = ch - 'a';
```

= 'z' - 'a'

= 25