

## \* Functions:

(1)

### - Syntax:

```
access modifiers    return-type    name (parameters) {  
    //body  
    return statement;  
}
```

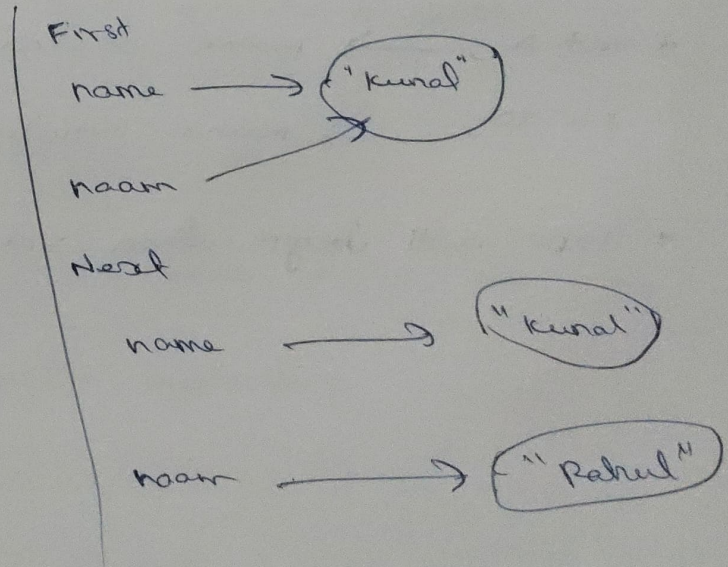
\* add static before return-type.

\* To call function syntax: function\_name ();

\* In java there is only pass by value. (means both value points to same object).

workflow

```
perm () {  
    name = "Kunal"  
  
    change (name)  
    print (name)  
}  
  
change (naam) {  
    naam = "Rahul"  
}
```





\* Primitives = int, short, char, byte, ...

↓  
just pass by value

objects and others : passing value of the reference variable.

Eg:-

```
main() {  
    a = 10, b = 20  
    swap(a, b)  
}  
swap(a, b) {  
    temp = a  
    a = b  
    b = temp  
}
```

a → 10  
a1 → 10  
b → 20  
b1 → 20

\* If you're using variable as global scope make it

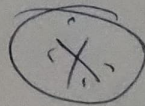
static.

11:04:00

\* Shadowing means if there is a variable declared as global scope but when a variable with same name as global variable is created inside block, inside variable will shadow the global variable.

\* int x; → means declaring

x = 30; → means initializing



\* Scope will begin when value is initialized.