Wednesday, 19 May 2021 8:43 PM

Functions

→ Restuarent
→ 10 coffees / tea

Black coffee — large — initial process();
5 Add tea leaves
⋮

Green tea — large — initial process()

Lemon tea — large — initial process()

```
initial process () {
    1  Take a large pan
    2  Boil water
    ⋮
}
```

→ A function is a block of code or collection of statements grouped together to perform a certain task or operation.

→ reusability of code
→ easy modification of code
→ reduce redundancy of code
→ reability of code.

Method declaration

public (access modifier) int (return type) sum (method name) (int a, int b) (list of parameters) {

// logic

}

add (a, b) → add(int a, int b)
add (a, b, c) → add (a, b, c)
add (a, b, c, d) → add (a, b, c, d)

→ Function overloading based on list of parameters

add (______)

int add (a, b) — int int
string add(a, b) — int int
boolean add (a, b) — int int

based on return type ?? — no

Functions

→ pass by value → Creating a copy in the memory of the actual parameters' value that is passed.

→ pass by reference → a copy of the address of the actual parameters is stored.

add(2, 3) } → actual parameters

int add (int a, int b) { → formal parameters

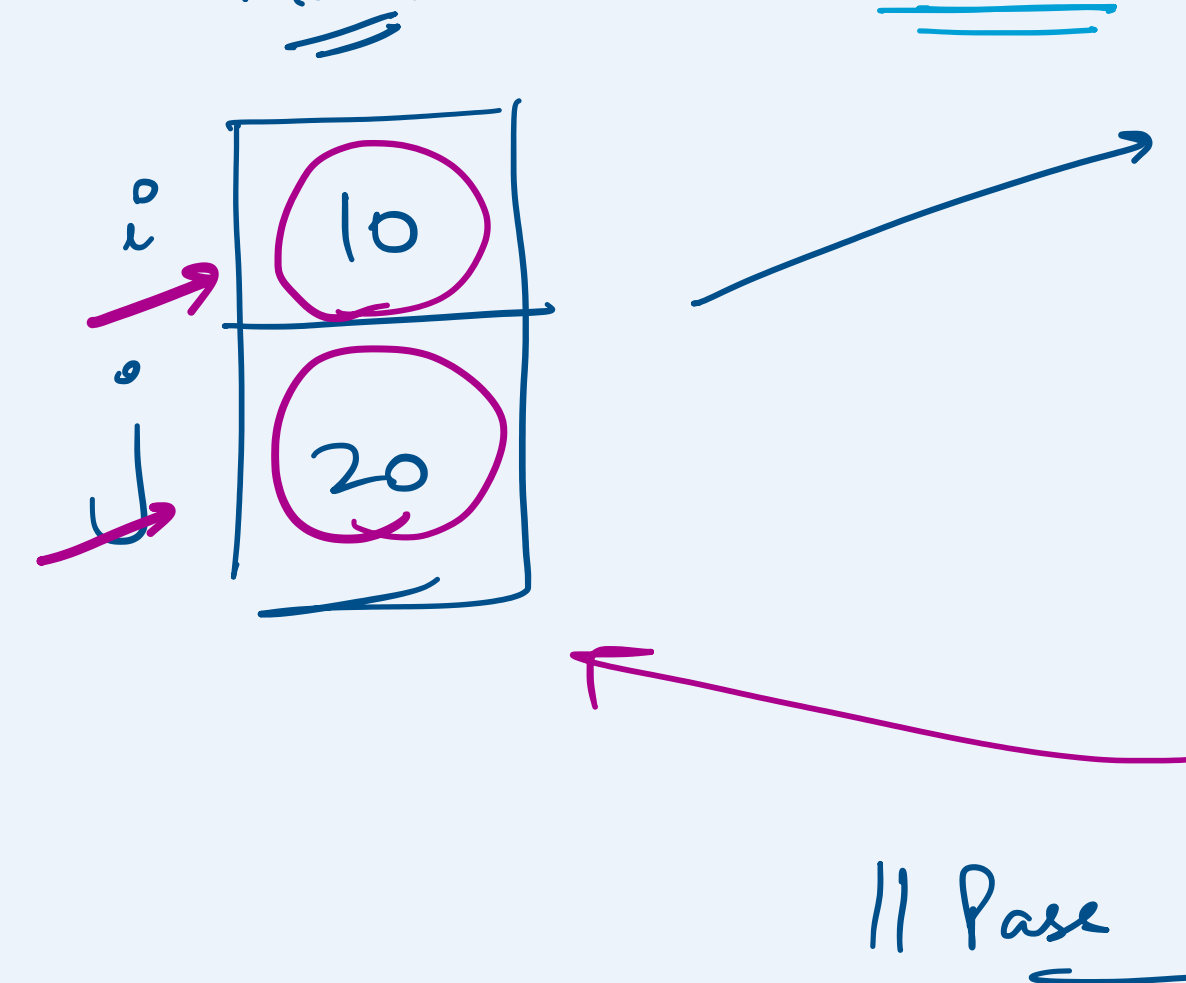
// logic

}

// Pass by value

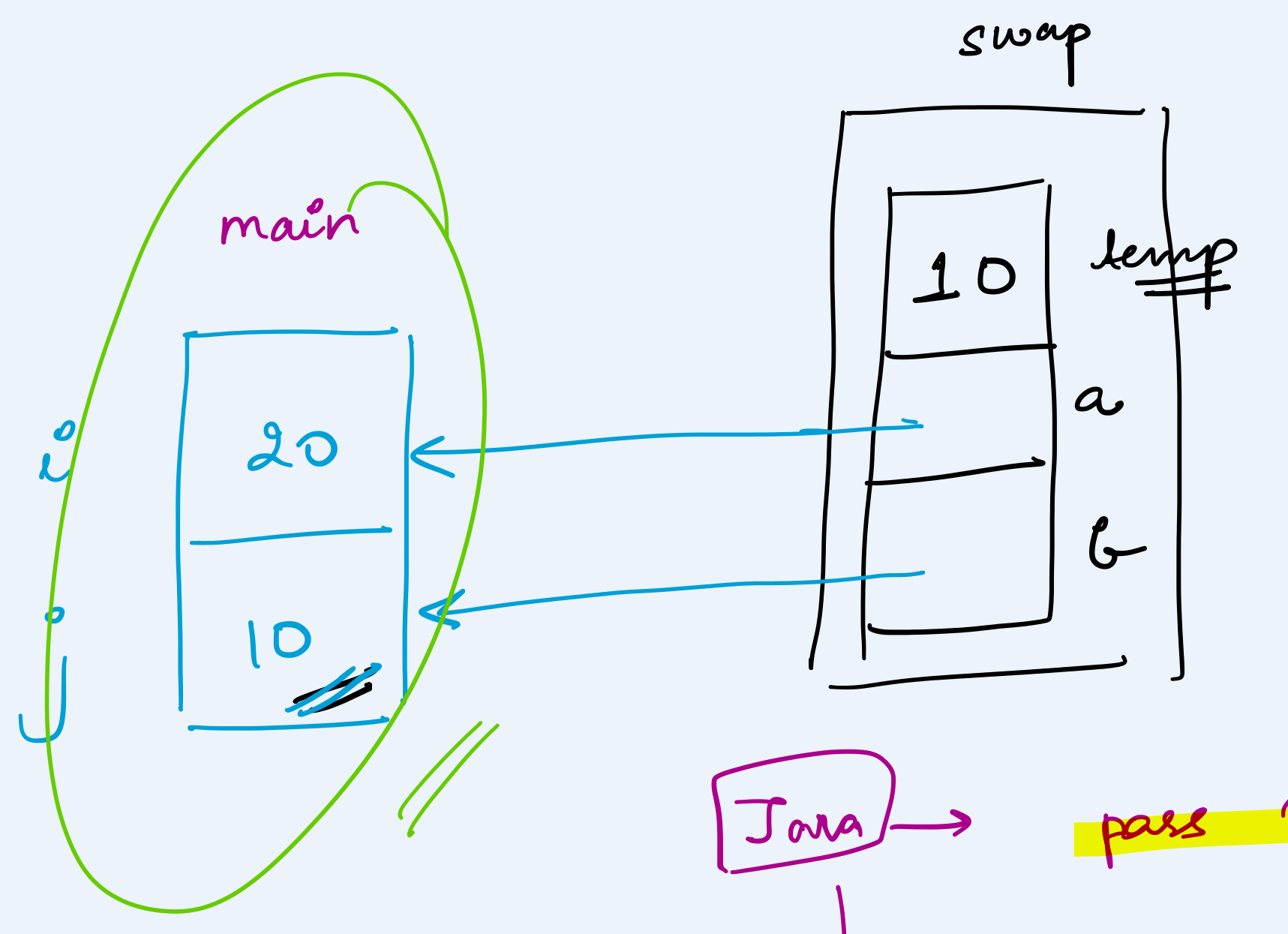
```
main () {
    int i = 10;
    int j = 20;
    swap (i, j) ;
    print ( i, j)          →  10, 20   non-swap ??
}
swap ( int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
```

pass by value



// Pass by reference

```
int main() {
    int i=10;
    int j = 20;
    swap ( i, j) ;
    print (a, b)      →   20   10
}
swap ( int &a, int &b) {
    int temp = a;
    a = b
    b = temp;
}
```



& *

Java → pass by reference ✗✗
Java → pass by value ✓✓

C++ → pass by reference ✓
C++ → pass by value ✓