

**COMPLEXITY FROM
CODE**

REMEMBER

- Think about complexity at algorithm stage
- Should know complexity of approach before you start coding

CODE COMPLEXITY

- Important to know how to determine in code
- Some code lines execute only a constant number of times, some dependent on n , etc.
- Typically this focuses on analyzing loop execution
 - How many times does loop execute
 - What is complexity of operations inside the loop?
- Most detailed level: count how many times each line executes
- Becomes natural enough to skip exact counts

EXAMPLES OF DIFFERENT LEVELS

- Constant - $O(1)$
 - A single operation (initialization, addition, comparison)
 - Multiple (constant number) of constant amount of work is constant
- Methods
 - Overhead of method doesn't affect complexity
 - Have to go into method to know method complexity

EXAMPLES OF DIFFERENT LEVELS

- $O(n)$
 - Typically single (not nested loops)
 - Inner operations combined are $O(1)$
- $O(n^2)$
 - Typically doubly nested loops
 - Inner loop is $O(n)$ and outer loop executes approximately n times

for (int j=0; j<10; j++) {

```
int sum = 0; 10
for (int i=0; i<n; i++) {
    sum += i; 10*n
}
```

}

```
public void foo(int n) {  
    int i = 1;  
    int sum = 0;  
    while (i < n) {  
        sum += i;  
        System.out.println(sum);  
        i *= 2;  
    }  
}
```

```
public void bar(int n) {  
    int sum = 0;  
    for (int i=0; i<n; i++) {  
        for (int j=1; j<n; j++) {  
            sum += i*j;  
        }  
    }  
    System.out.println(sum);  
}
```