With recursive algorithms there is always a “recursive case” and a “base case”

Recursive case – where there is recursion

Base case – no recursion

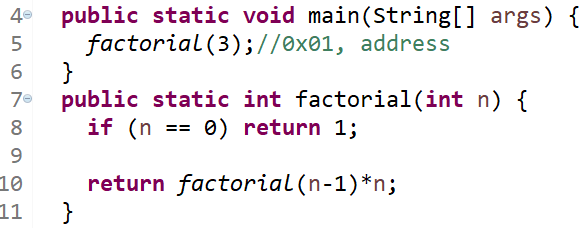
How to write a recursive method:

1. Write “if”

2. Handle the base case

3. Write the recursive call – always reduce the size of the problem!

4. Assume the recursive call works and ask “how does knowing the result of the recursive call help?”



factorial

n : 3

return address: 0x01

return 3\*factorial(3-1) //0x02

factorial

n : 2

return address: 0x02

return 2\*factorial(2-1) //0x03

factorial

n : 1

return address: 0x03

return 1\*factorial(1-1) //0x04

factorial

n : 0

return address: 0x04

return 1

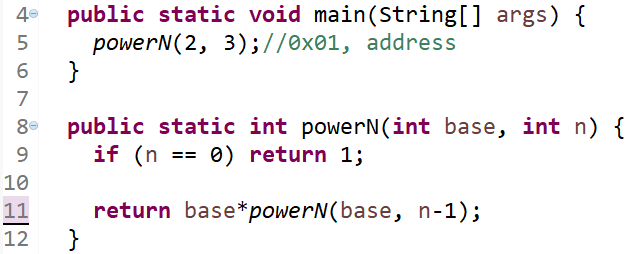
1

2

6

1

Method Stack Frames



powerN

n : 3

base: 2

return address: 0x01

return 2\*powerN(2, 3-1) //0x02

powerN

n : 2

base: 2

return address: 0x02

return 2\*powerN(2, 2-1) //0x03

4

8

2

Method Stack Frames

powerN

n : 0

base: 2

return address: 0x04

return 1

powerN

n : 1

base: 2

return address: 0x03

return 2\*powerN(2, 1-1) //0x04

1