

(int / void)      function-name ( ) { → C/C++

↗ return-type      method-name ( ) { → Java

// function body      ↳ (parameters)  
 ↳ (function/method)  
 ↳ variables

↳ public / private / protected / (default) → when nothing is written

↳ static (Objects) + ↳ CBT      ↳ f(2, 3) (actual values)  
 ↳ new      ↳ arguments

(Reverse Integer) : → n = 123 → of P 321

int reverseInteger (int n) {  
 digit = n % 10  
 ans = 0  
 }      ↳ r = 0      ↳ n / 10  
 ans = ans \* 10 + digit      ↳ rev = rev \* 10 + 3  
 n = n / 10      ↳ 32      ↳ rev = 0 + 3  
 ↳ 32 / 10      ↳ n / 10      ↳ = 3  
 while ( )      ↳ -3      ↳ 3 / 10 = 0  
 ↳ stop

int x = 10;  
 ↓ keyword      ↓ val      ↓ value  
 ↓ identifier

[0x1000]      &x  
 0x1000

int \* ptr = &x;  
 (0x1000)

[xptr = 10]

{ value (out of scope)  
 address (↑ ) }  
 dangling point

Storage Classes:		Scope	Lifetime
* static		global	file
* extern		global	multiple files inside same folder
* auto		local	{ }
* register		RAM	RAM
f1.c	f2.c		→ (No pointers)
[static ✓] x	[extern ✓] y		[SNPSU-CS-IS] folder

### Baker's Partition Problem

$$f(s) = 16$$

$$\frac{n \times n+1}{2} = \frac{s \times s}{2} = 7^2$$

No of Cuts	Max Pieces
S(n) + 1	1
$\frac{n \times n+1}{2} + 1$	2
2	4
3	7
4	11
5	16

### Max Regions in a Plane

Using N straight Lines

int max-pieces (int n)

int max-regions (int n)

No of Straight Lines	Max Regions
0	1
1	2
2	4
3	7
4	11
5	16

int arr[] = {1, 2, 3, 4, 1, 2}; (xor)

static void findTwoUniques (int arr[]) {

[0/1] → 3, 4 (print) [O(n)] ≈

{ n1 = 0

n2 = 0

\* \* [xor b] = xorAll & -xorAll |

u1 u2

1 1 x 2 x 2

3 3 4 4

1 1 x 2 x 2

= 3 = 4

(n1 = 0) { 1, 2, 3, 4, 2, 1 } 0011 0100 ↳ 1 arr[0] ↳ rsb n2 = 0

xorAll = 1 & 2 & 3 & 4 & 2 & 1 = 7

rsb = xorAll & -xorAll = 7 & -7

= 0001 or 1

for (i = 0; i < arr.length; i++) {

if (arr[i] & xorAll != 0)

↳ n1 & arr[i]

0 & 1 & 3 & 1 = 3

n2 & arr[i] 0 & 2 & 2 & 4 0010 0001

(4) 0

arr [1, 2]

swap (arr)

2, 1