* Given an integer value 'n' write a program to reverse the given integer using a "User-Defined Function".

int reverseInteger(int n){

}

I/p → 123
O/p → 321
I/p → 120
O/p → 21

Suppose n = 123 →

Remember: 123 = 1×100 + 2×10 + 3

We need to target the single digits of the number.

So, n = 123 ⟶ n = n%10 = 123 % 10
digit = 3

Let's take ans = 0
Now, ans = ans×10+digit
= 0×10 + 3
= 3

ans = ans×10 + digit
= 3×10 + 2
= 32

32×10 + 1 = 321

Now, n = n/10
= 123/10
= 12

Repeat: n % 10 = 12 % 10
= 2

12/10 = 1

1/10 = 0 (stop)

ans = 0
ans = ans×10 + digit
= 0×10 + 1
= 1

ans = ans×10 + digit
= 1×10 + 3
= 13

13×10+6

$\boxed{136}$ ✓

n = 63①

digit = n%10 = ①

n = n/10
= 631/10 int
= 63

digit = n % 10
= 63 % 10 = 3

n = n/10 = 6

6 % 10 = 6

n/10 = 6/10 0

* Easiest but most important topic in C & C++ :→

Pointers :→

int a = 10;
↓
variable ⟶ CPU ⟶ Memory | Address | 1000

The address can be accessed in two ways:

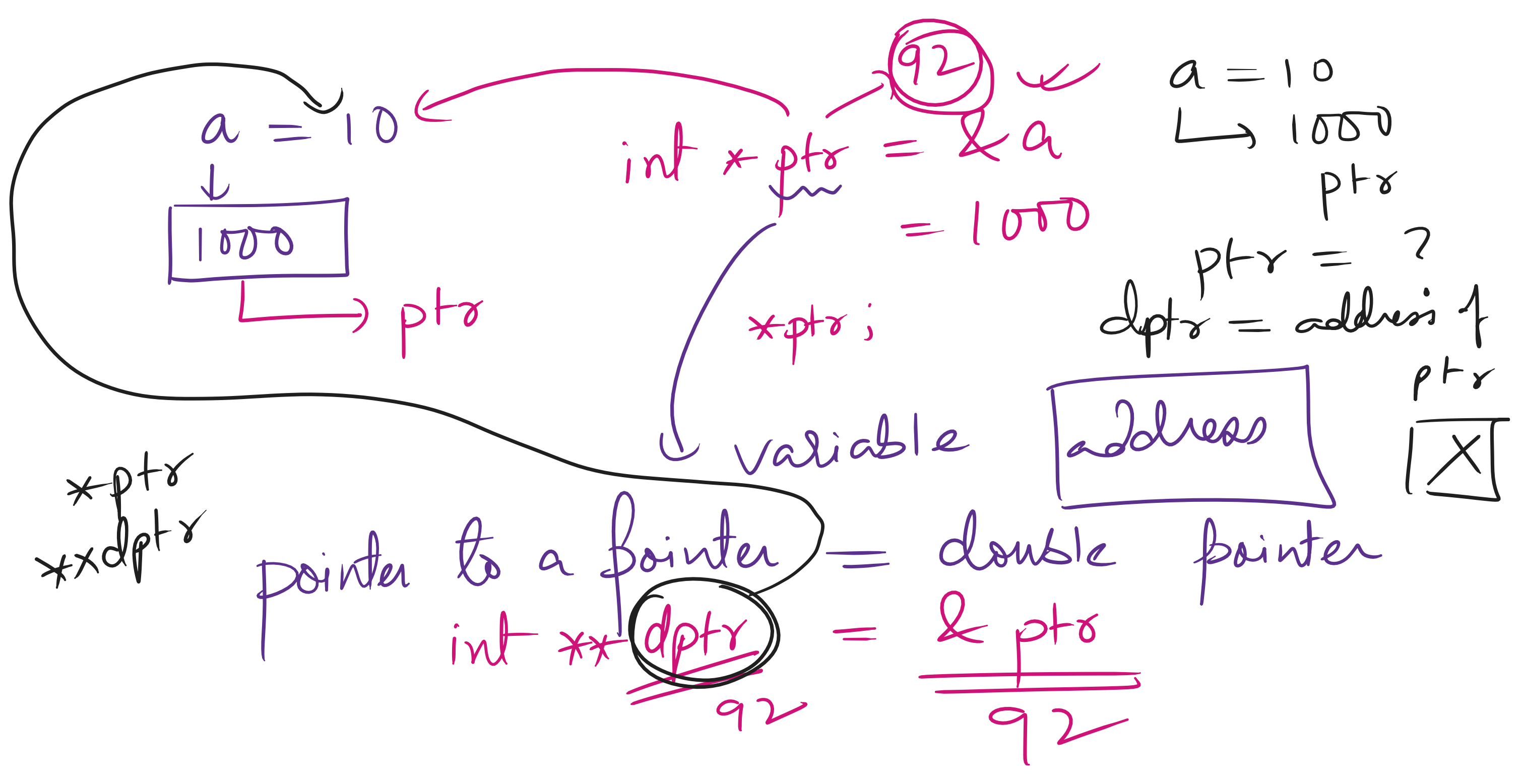
① Address Operator ⟶ &a

(ii) Pointer or Reference

int * ptr = &a; ← 1000

a | 1000 | → ptr
↓
10

How to get value of a using pointer?



pointer to a pointer = double pointer

int ** dptr = & ptr

Swapping two numbers :→

int a = 10;
int b = 20;

int temp = a;
a = b;
b = temp;

10 20 copy

swap(int a, int b)

10, 20
20, 10 ✗

CBV

main( ) {
a = 10 [1000]
b = 20 [2000]
}

pointers

Arrays :→ Collection of Homogeneous Data
(C, C++, Java)
↓
Same data type

Python : List → [1, 1.2, "a" , True]
(Heterogeneous)

int arr[ ] = { 3, 8, 6, 9, 4 }

1 2 3 4 5 → Positions

0 1 2 3 4 → Indexes

index = pos − 1 | pos = index + 1

What's the formula for calculating the dynamic size of an array?

[3, 6, 9, 8, 1, 0, 4]

$\frac{28}{4} = 7$

$\frac{7}{1}$

7 elements = 7×4 = 28 = ④

$= \frac{28}{7}$

sizeof(arr) / sizeof(arr[0]) = ⑦

28 / 4

arr[0] − i, c, f, d,

sizeof (int)

int arr[ ] = {1, 2, 3, 4, 5};   4
char arr[ ] = {'a', 'b', 'c'};   1
float arr[ ] = {1.2f, 4.5f, 6.8f};   4
double arr[ ] = {2.45, 9.26, 8.46, 3.99};   8

int size = sizeof(arr) / sizeof(arr[0]);

generalized.