Subsequences. (Recursion)

Memory Allocation

Compile Time

→ Static Memory

→ Small.

→ Self Destroy

→ Program fixed-size

eg

```
int a;
int b[100];
```

Run time

→ Heap memory.

→ Big

→ Programmer has to destroy

→ Run time allocation. / 'on the fly'

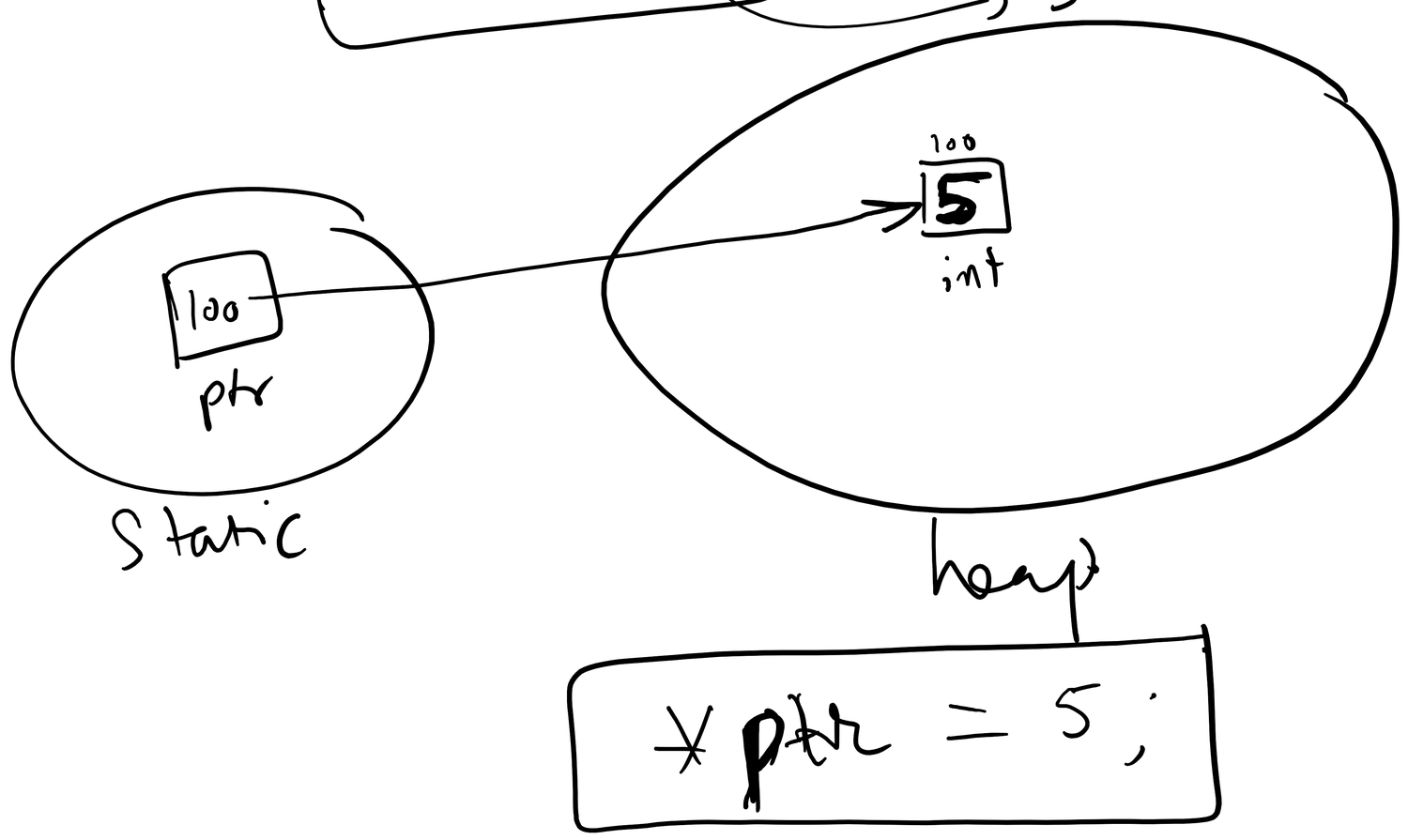
→ 'new' → Allocation

'delete' → Deallocation

int *ptr = new int ;

Allocates memory.

Size

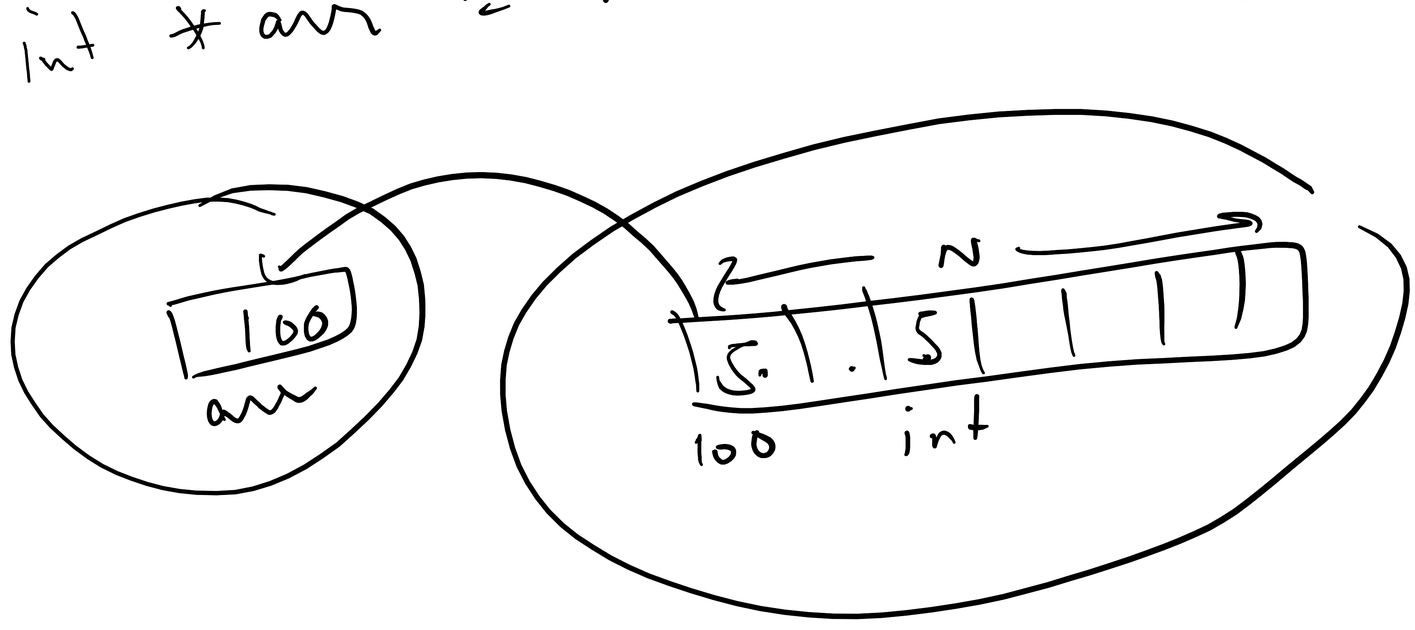


*ptr = 5;

```
int n;
cin >> n;

int *arr = new int [n];
```

Greek



arr(2) = 5

*(arr + 2) = 5

arr[0] = 5; ✓

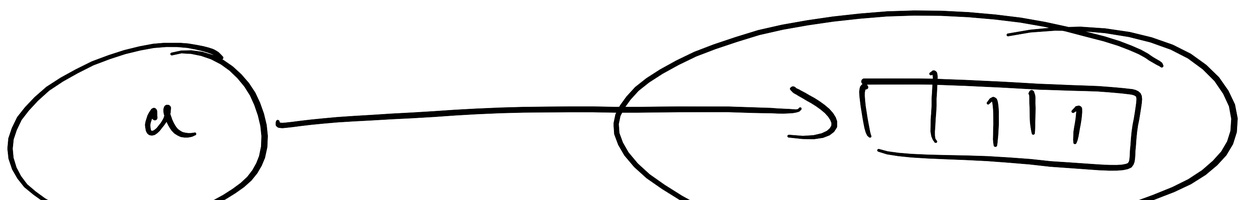
*(arr + 0)

Delete

int *xptr = new int;

delete xptr;

int *a = new int[10];



For arrays

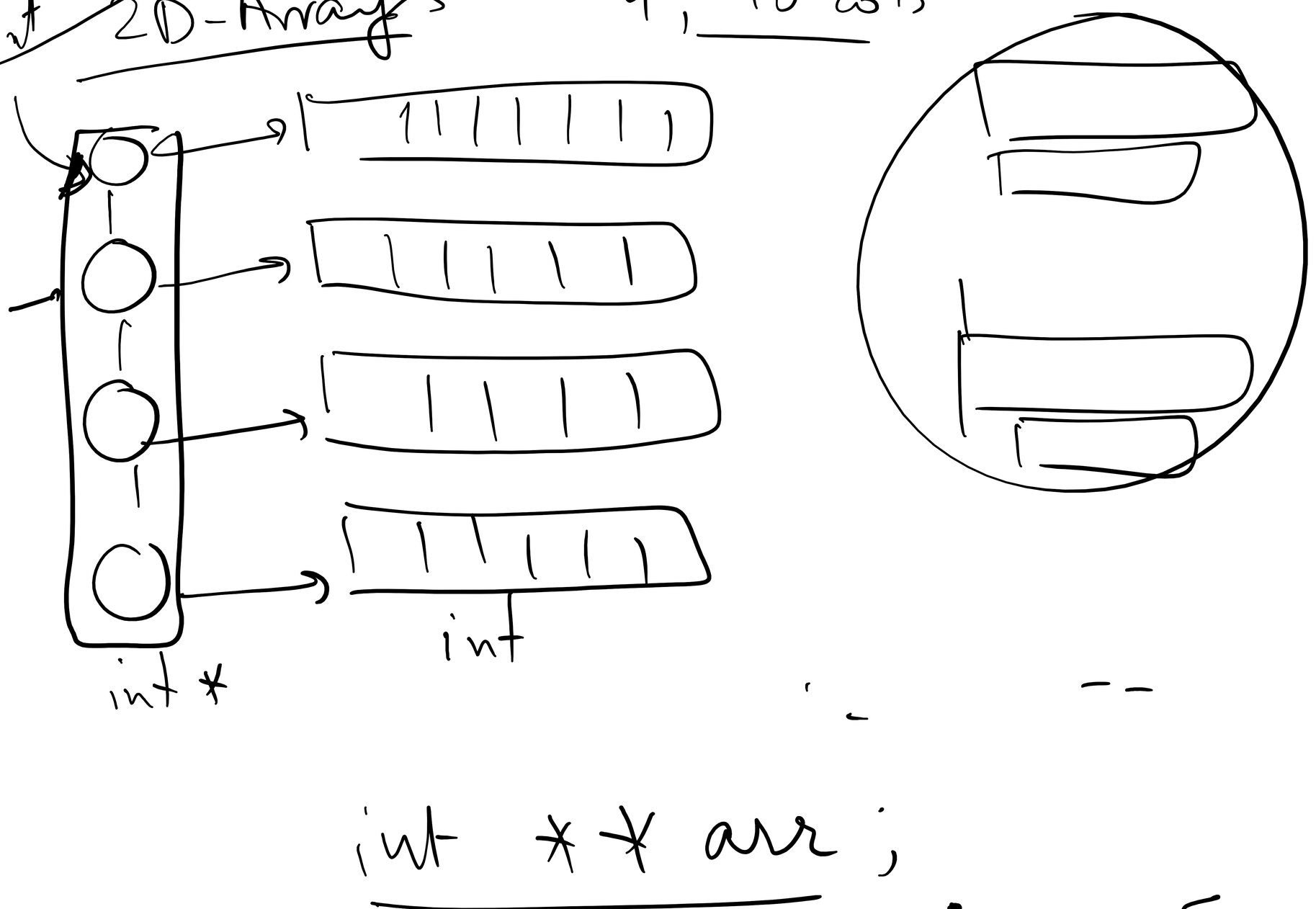
delete [] a;

a = null;

free/delete

int ** 2D-Arrays

4, 10 cols



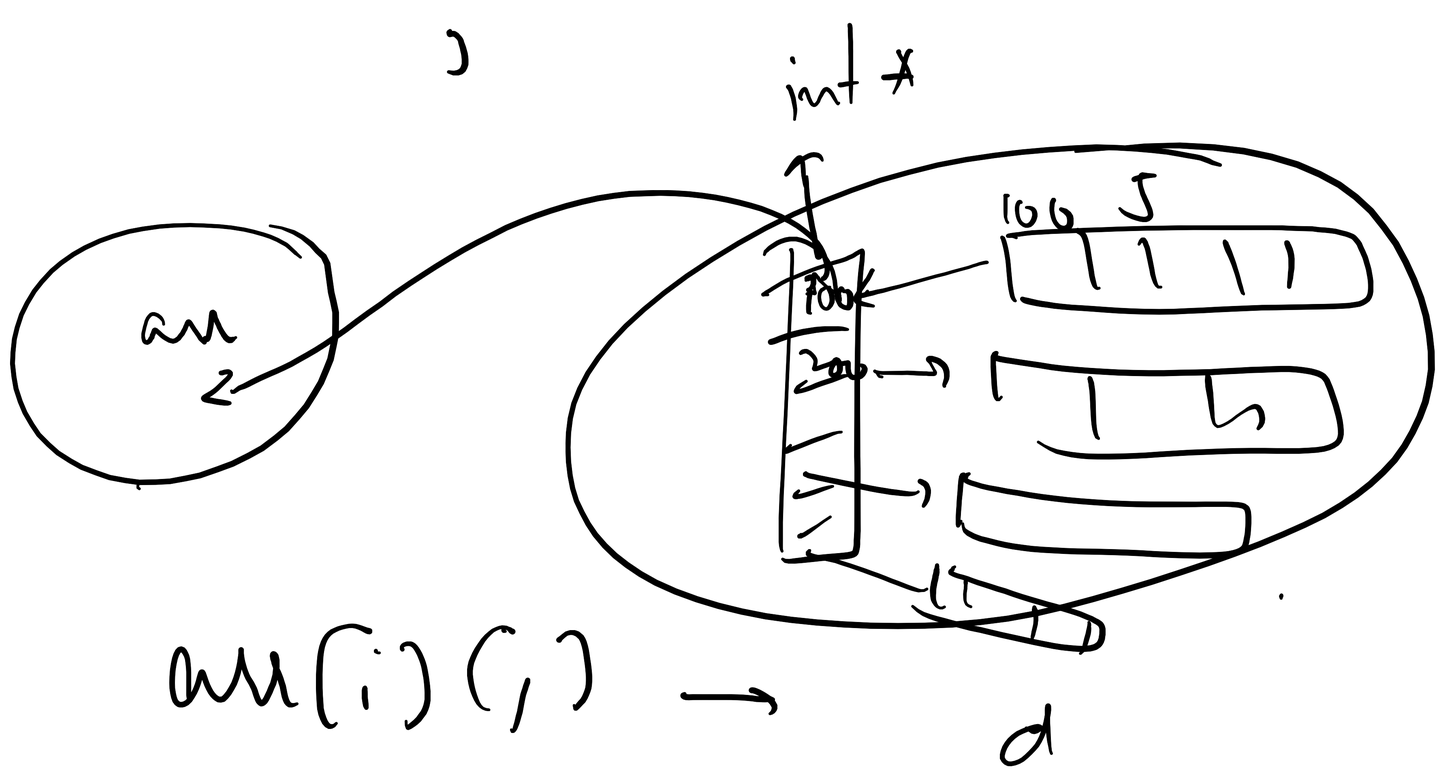
int ** arr;

arr = new int * [Rows];

for (i=0 ——— < Rows, i++)
{

arr [i] = new int [cols];

}



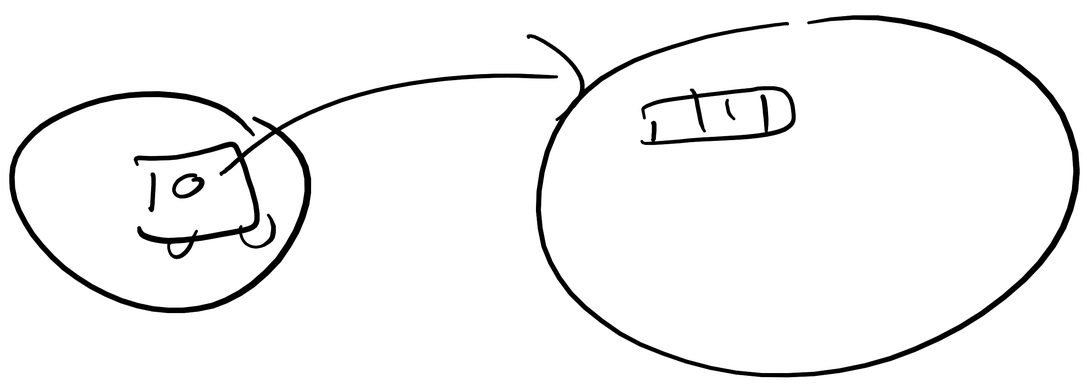
d2 = c ;

d2. ~~equal~~ (c);

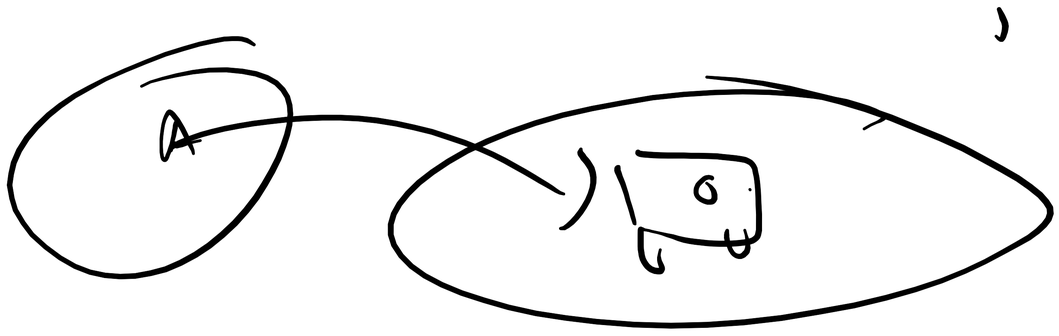
d2 . == (c) ;

d2 . = (c) ;

Car c ;

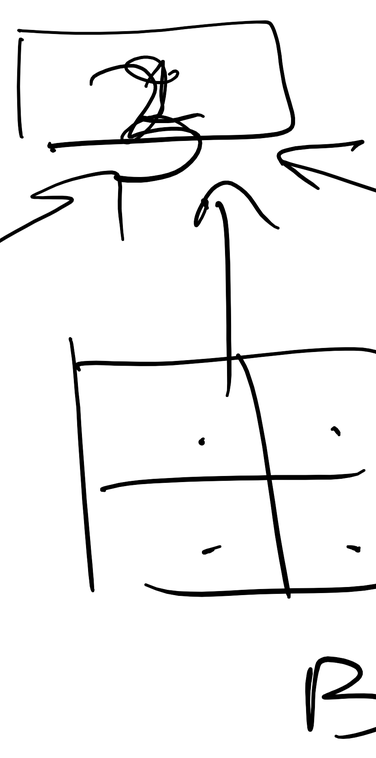


Car ↑ A = new Car ;



DM

cont = 0



Common for all

A B Static