

Structs / Structures in C++ → custom user defined

datatypes . Combination of multiple primitive and non primitive datatypes

product {

string name;

string desc;

float cost;

}

Types of pointers

① Dangling pointer \rightarrow A pointer pointing to a memory loc, that has been deleted.



② NULL pointer \rightarrow A pointer which points to nothing or NULL.
 $\text{int}^* \text{ptr} = \underline{\text{NULL}};$ or $\text{int}^* \text{ptr};$

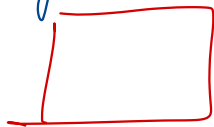
3 void pointer \rightarrow generic pointer

int * pr
float * pr

for pointer arithmetic \rightarrow (pr+1) pr+1
pr-1
:

void pointer, points to any datatype address & can
be also changed later to other datatype.

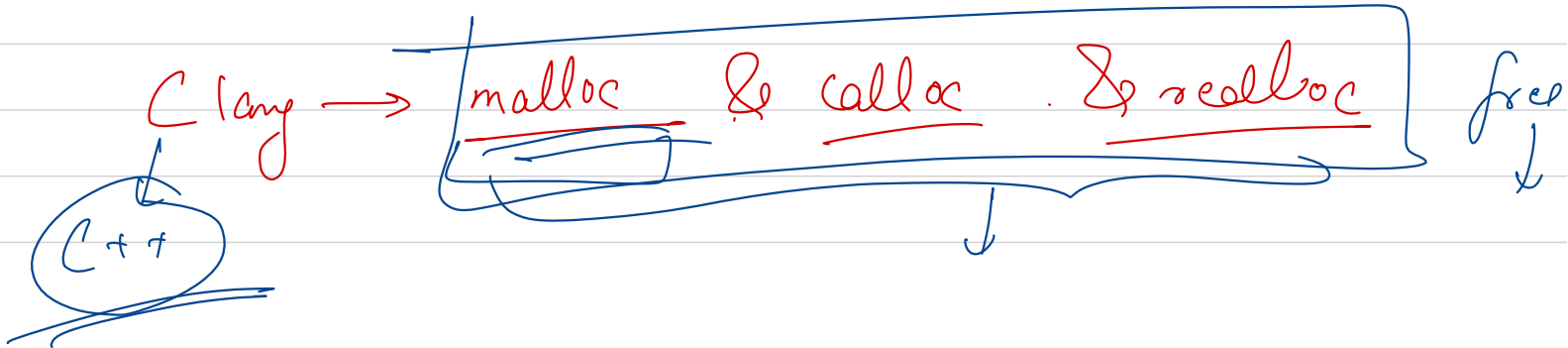
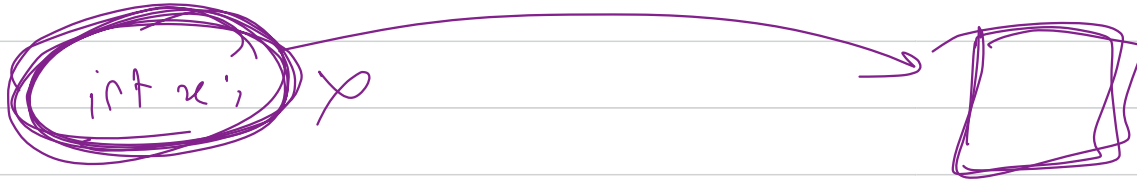
we can't dereference a void ptr & no arithmetic can
be done

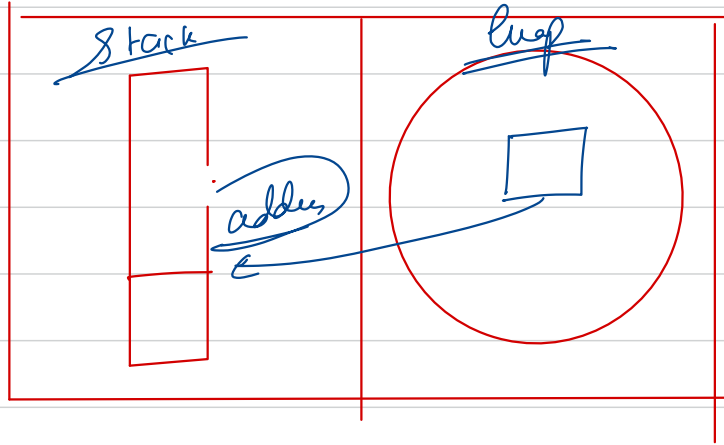


initialise a variable and, then get the address

How about, if we can get the address & bucket first & then set some value.

int x;





Function call

```
{  
  func() {  
    int a(7);  
  }  
}
```

```
int x=10;  
int y;
```

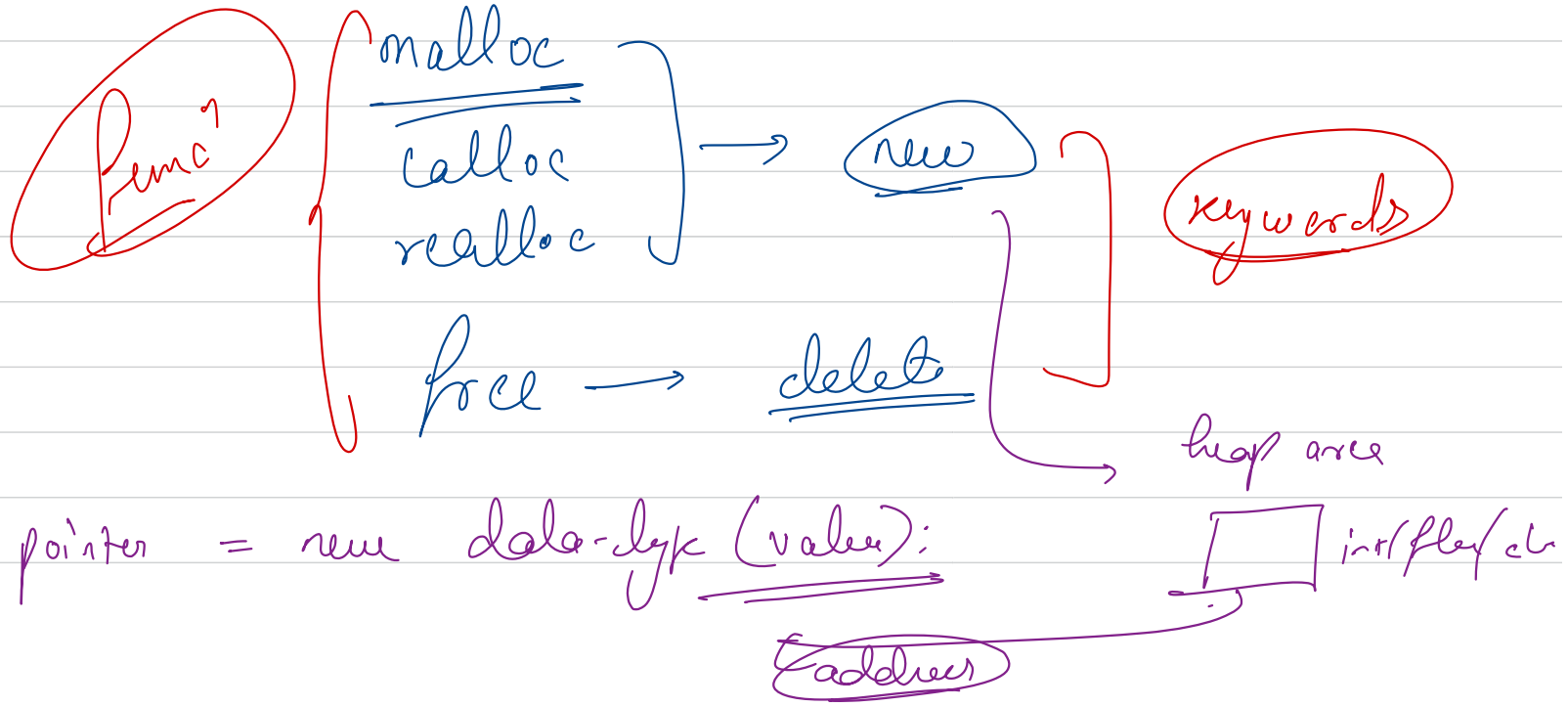
→ Stack
running

④ Wild pointer \rightarrow An uninitialised pointer is a wild pointer.

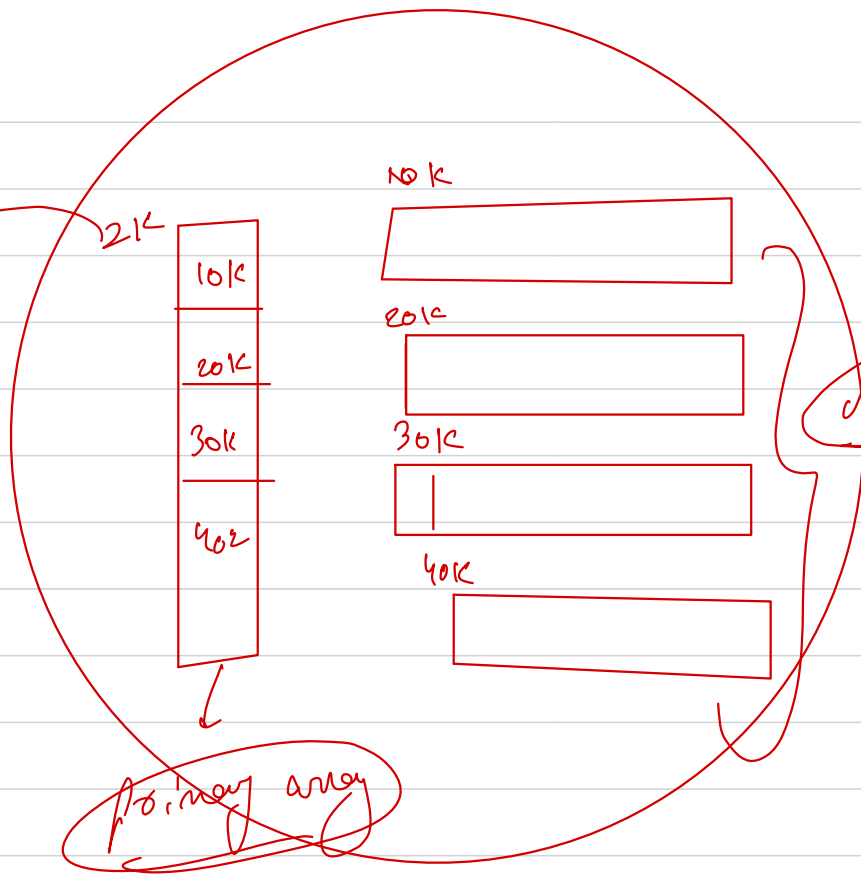
$\text{int}^* p;$

⑤ function pointer \rightarrow

Let's say our use case is simple, we just want to allocate in heap memory



24
↓
int *** arr =



void fun(int *z) {

3

pointer variable
made in
stack frame

z = 10K fun(z)

main

z = 10K



10K

int *z = new int[n];

n = 3