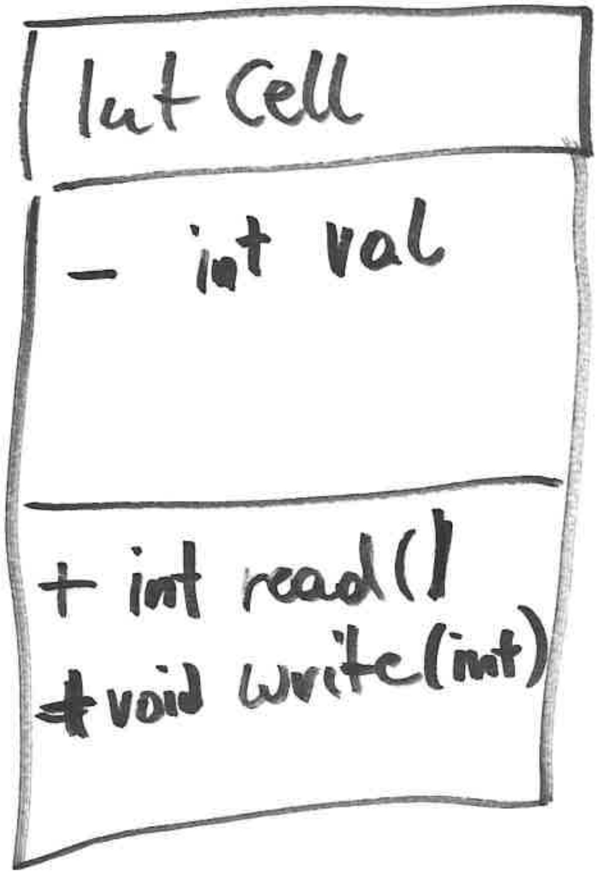
CS 2341

Chapter 1

C++ Programming

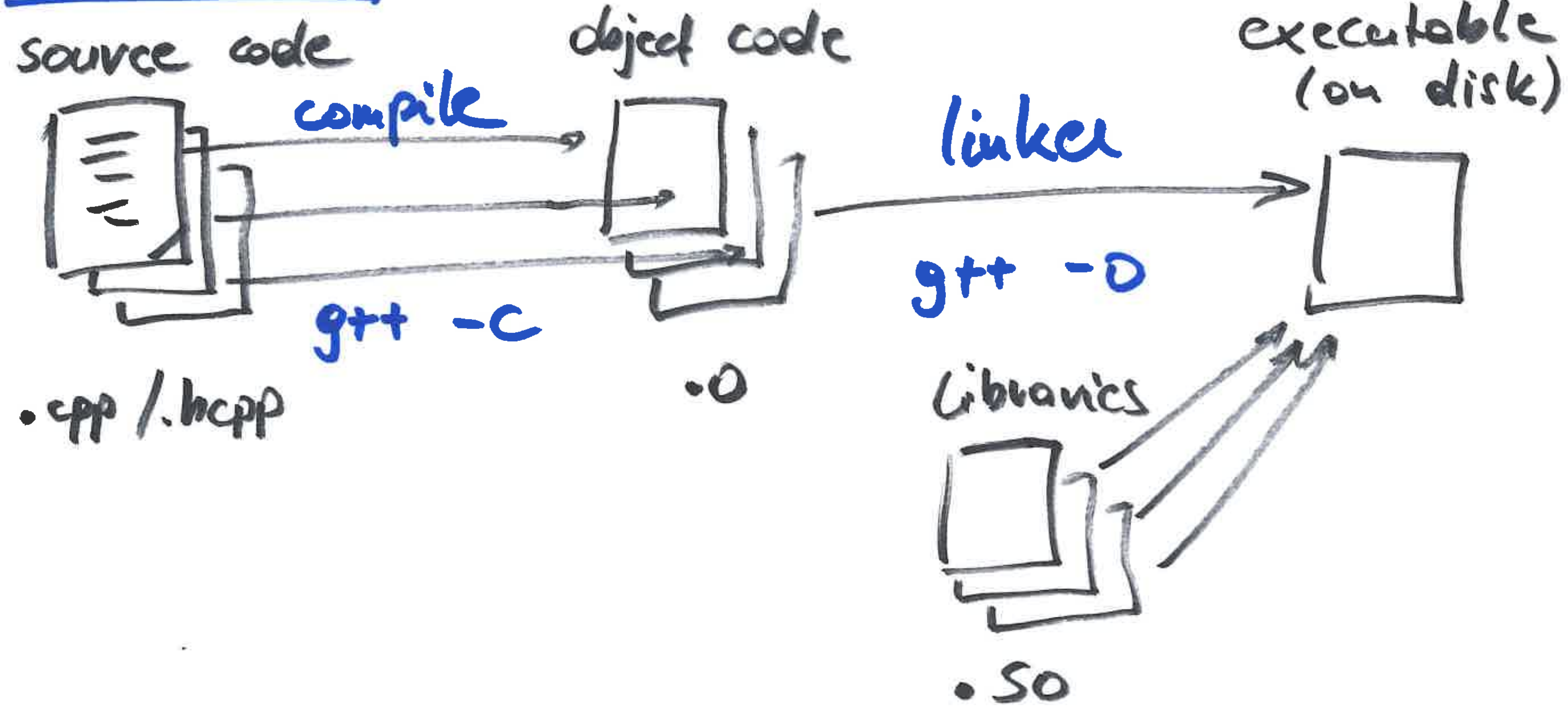
IntCell



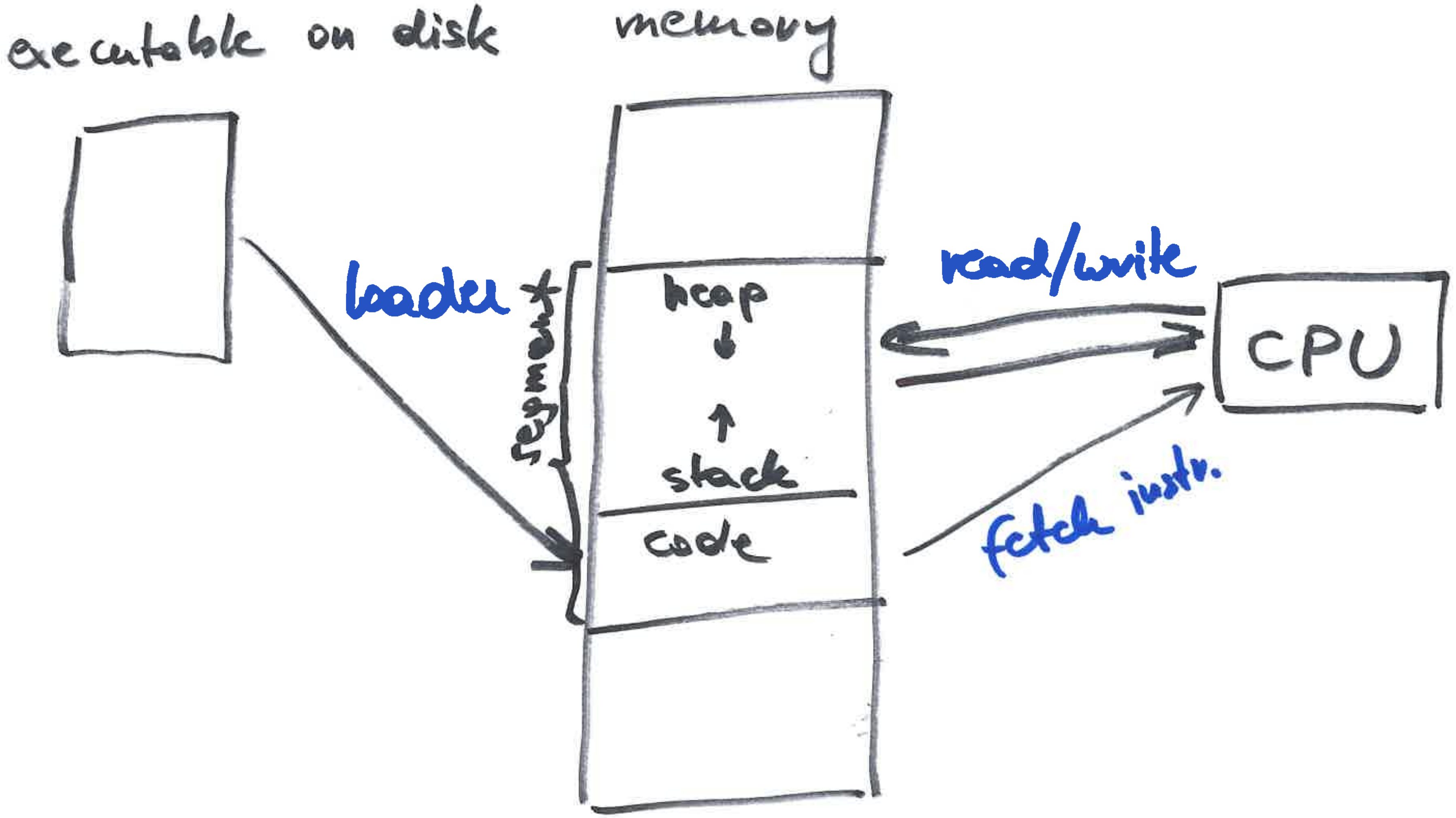
Steps

1. Design (UML) ✓
2. Test cases (main.cpp) ✓
3. Implement (IntCell.h, IntCell.cpp)

Compilation



Execution

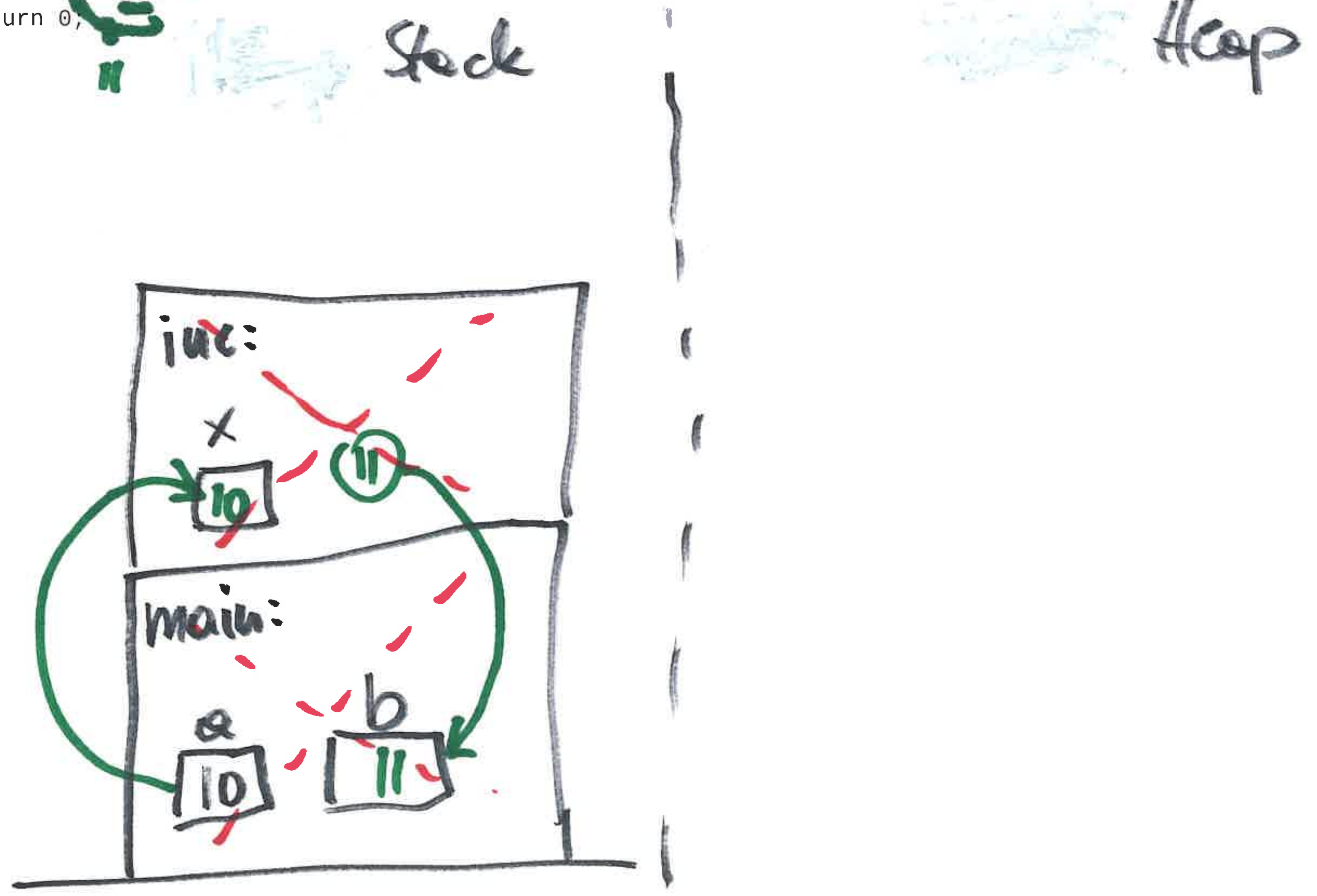


executable on disk
memory
loader
segment
heap
stack
code
read/write
CPU
fetch instr.

```
int inc(int x){return x + 1;}

int main() {
  int a = 10;
  int b = inc(a);
  return 0;
}
```

11

Memory Diagram

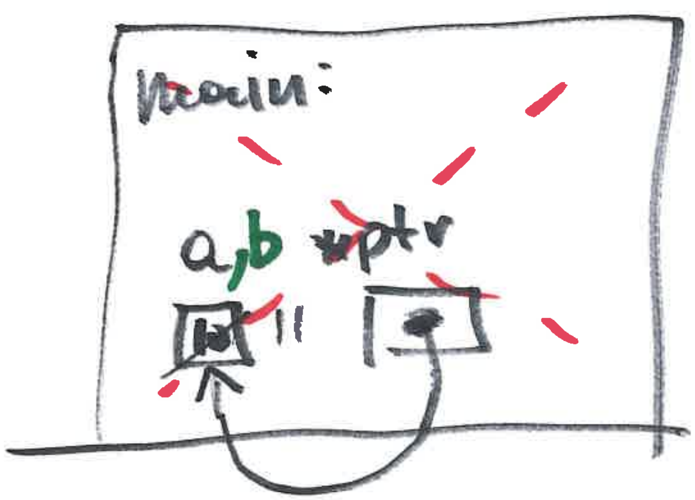


Pointers & References

```
int main() {

  int a = 10;
  int* ptr = &a; // create a pointer to a using the address-of-operator
  int& b = a;   // create a reference for a

  std::cout << "a = " << a << "; *ptr = " << *ptr << "; b = " << b << "\n";

  b++;
  std::cout << "a = " << a << "; *ptr = " << *ptr << "; b = " << b << "\n";

  return 0;
}
```

Stack

Heap

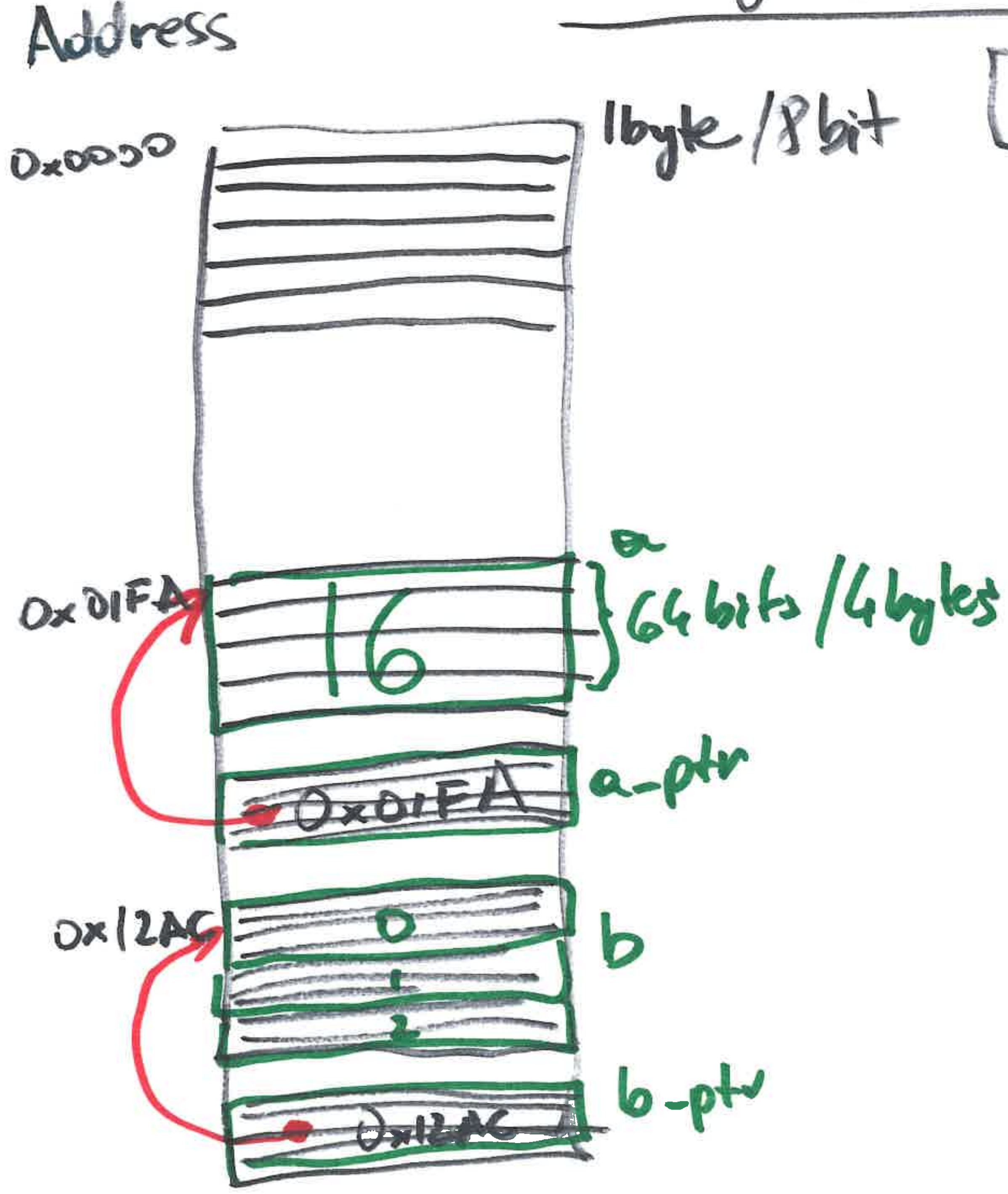


Output:

a = 10; *ptr = 10 ; b = 10

a = 11; *ptr = 11 ; b = 11

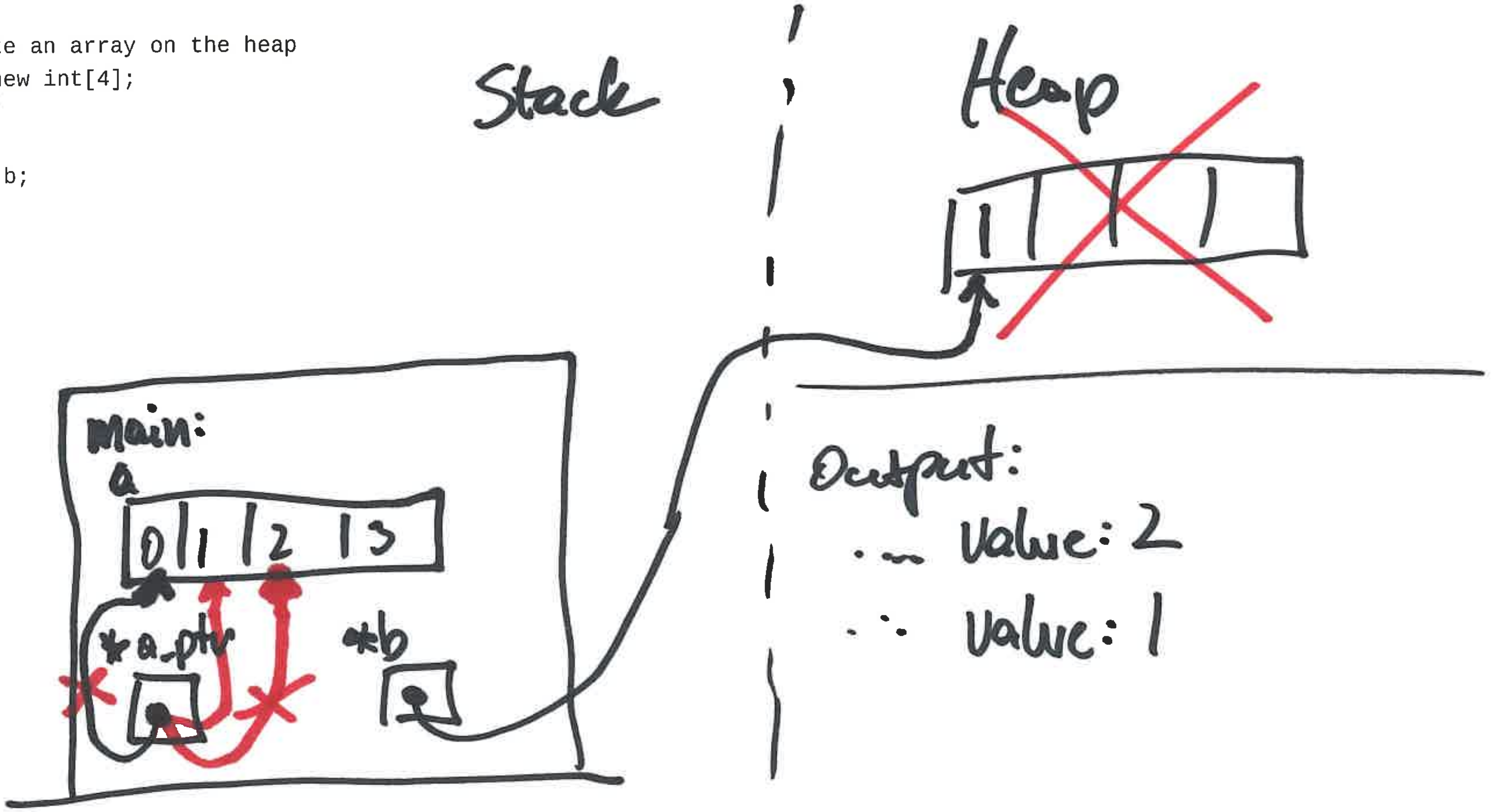
Memory & Pointers



$\boxed{\text{pointer = address}}$

```
int a=16;
int* a_ptr = &a;

int[3]b={0,1,2};
int* b_ptr = b;
b_ptr++;
```

```
int main() {

  // create an array on the stack (you can also use int a[4];)
  int a[] = {0,1,2,3};
  int* a_ptr = a;

  // this is the same as a[2] or *(a+2)
  a_ptr += 2;
  std::cout << "a_ptr point to value:" << *a_ptr << "\n";

  --a_ptr;
  std::cout << "a_ptr point to value:" << *a_ptr << "\n";

  // allocate an array on the heap
  int* b = new int[4];
  *b = 1;

  delete [] b;

  return 0;
}
```

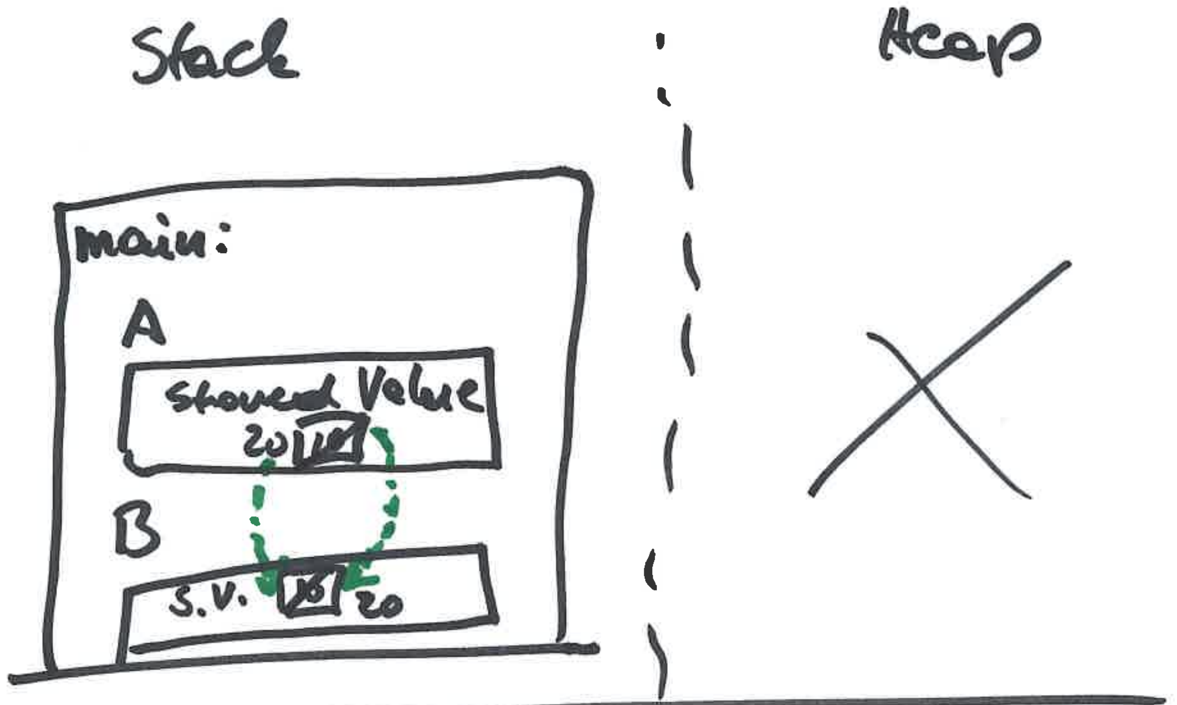


Dynamic Memory Allocation

	C	C++	Java	Python
Request dynamic memory	malloc()	new	new	Constructor Function
Return dynamic memory	free()	delete delete[]	Garbage Collection "Reference Count"	G.C.

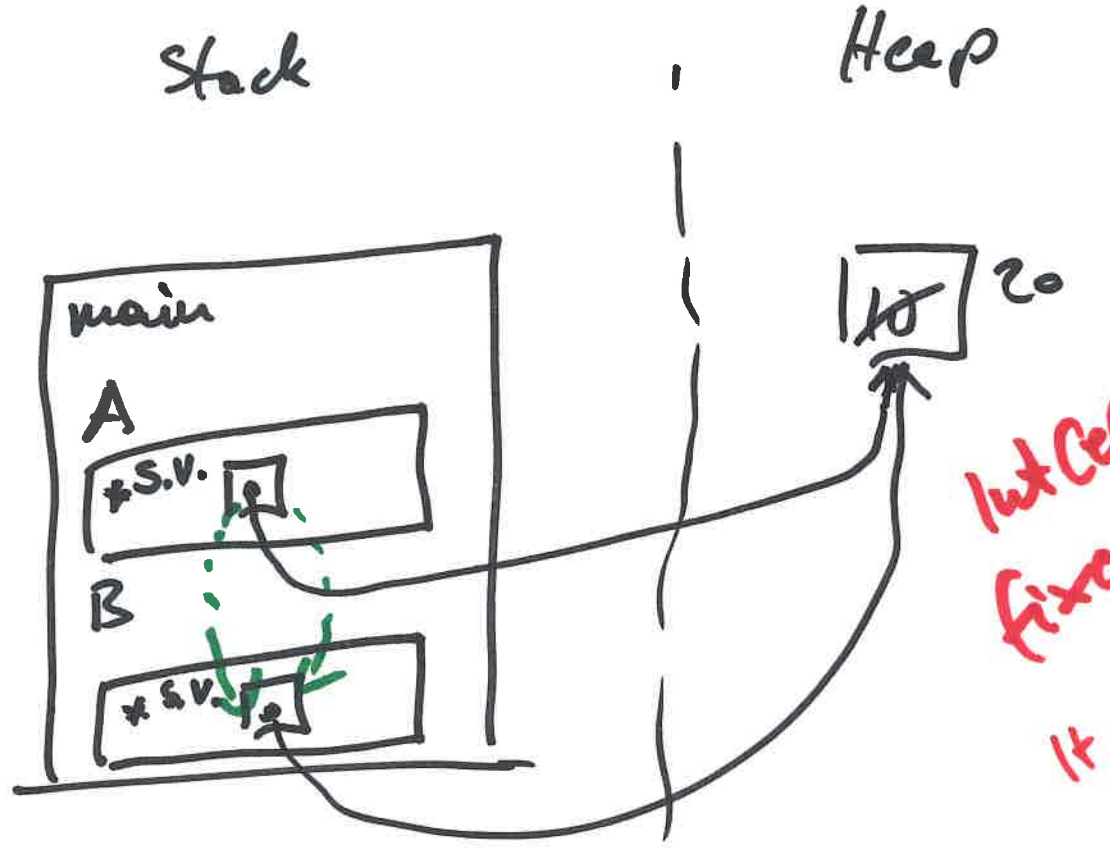
Rule-of-3

IntCell



IntCellPtr

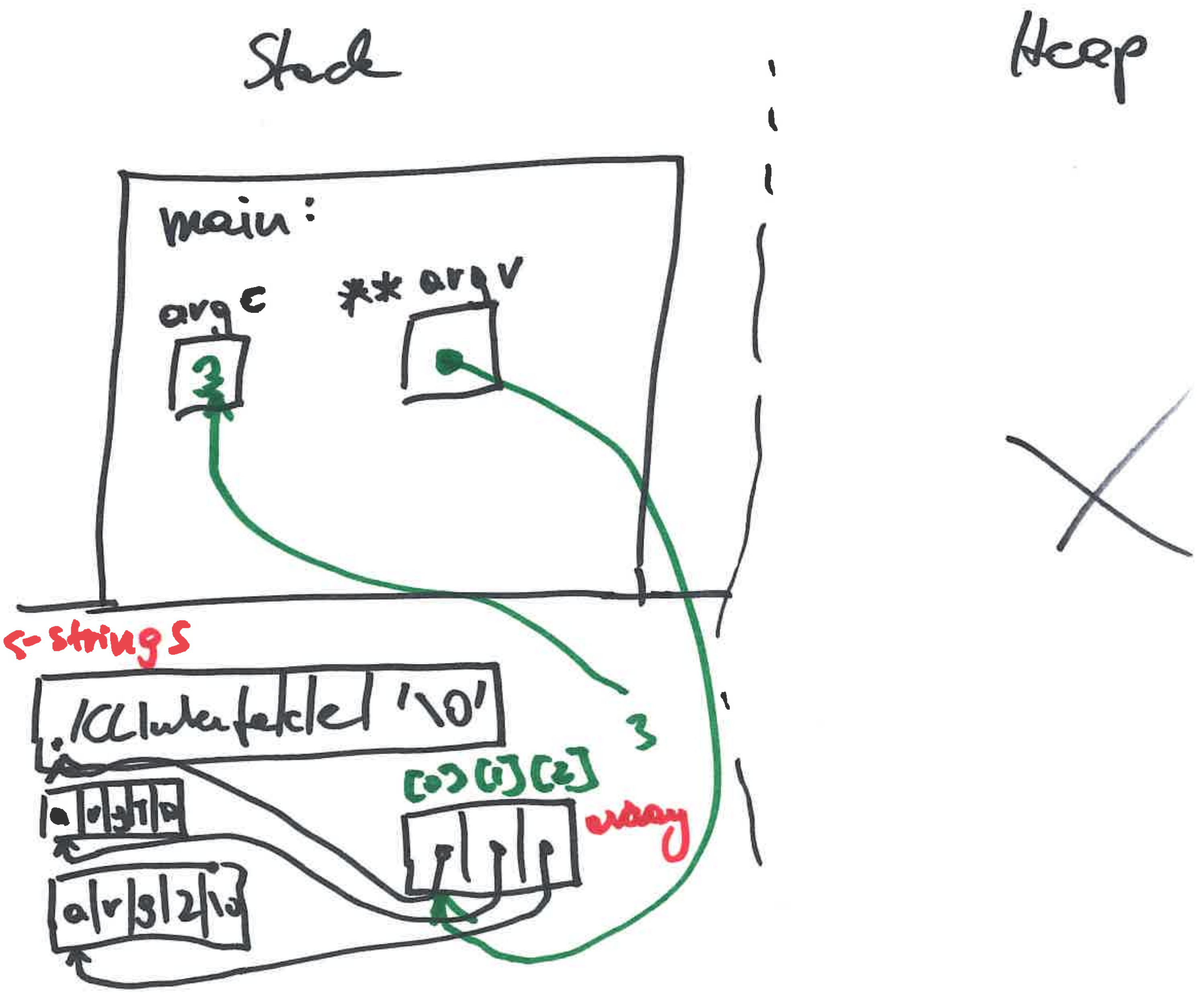
→ Default copy constructor etc. create "shallow copies"



IntCellPtrOK fixes this issue! It provides the rule-of-3 to create "deep copies"

Command Line Arguments

```
> ./CInterface    arg1    arg2
```



Stack
Heap
main:
argc
3
**argv
<-strings
./CInterface '\0'
a|r|g|1|\0
a|r|g|2|\0
3
[0] [1] [2]
array