Back to basics: Pointers in C

Programming basics start with pointers - pointers are conceptually difficult and hard to wrap one's head around

However they are the foundation around which hard programming problems are built - they force conceptualization of memory layout and allocation which are useful for everything we build on top.

Even if you do not plan to code in C, working with pointers is a great warm up exercise to get started with preparing for interviews.

Before we get to pointers...

How is memory laid out for a program?

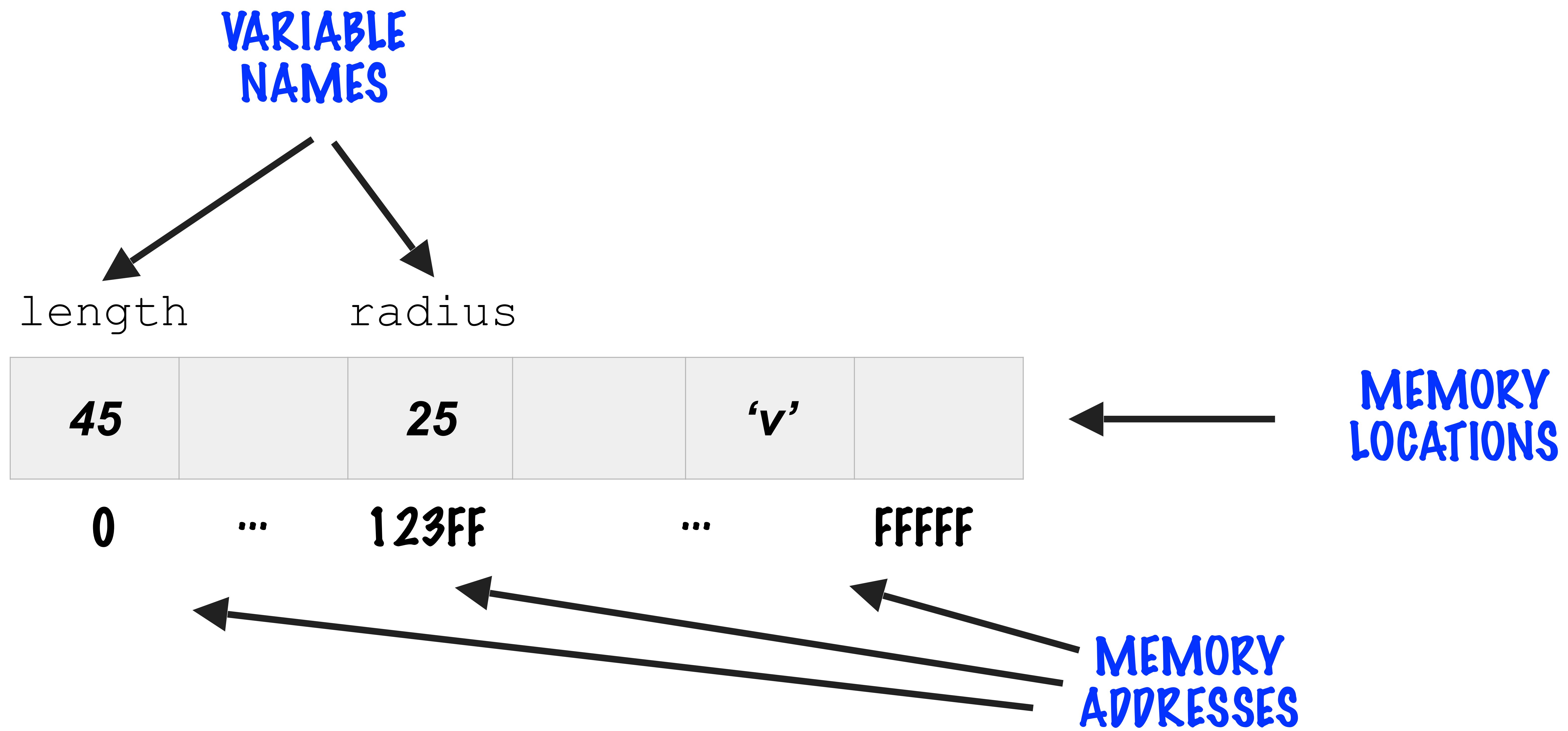
Memory is divided into blocks of a certain size (32-bits or 64-bits depending on your OS)

Each block has a specified address - this means every location in memory can be accessed by using its address

That’s pretty painful which is why we do not deal with addresses, instead have variables which symbolically represent a block in memory

HOW IS THIS INFORMATION STORED IN MEMORY?

```
int radius = 25;
int length = 45;
```



radius: SYMBOLICALLY REFERS TO A LOCATION IN MEMORY

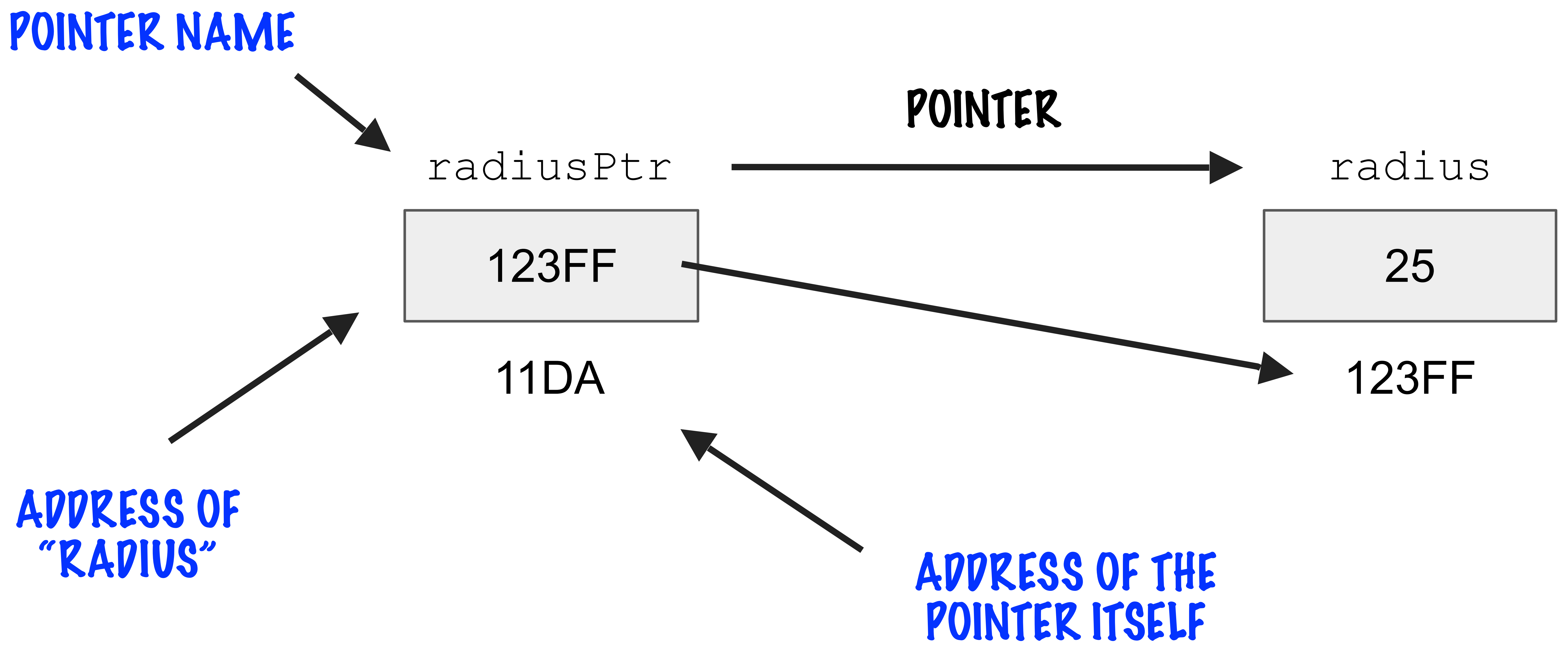
25: IS THE VALUE STORED IN THAT LOCATION, IT LIVES IN THAT MEMORY SPACE REFERRED TO BY "RADIUS"

THE LOCATION HAS AN UNDERLYING ADDRESS THAT IS KNOWN BY YOUR C PROGRAM, WE SIMPLY REFER TO IT AS "RADIUS"

WHAT IS A POINTER?

A POINTER IS A SPECIAL TYPE OF VARIABLE. REGULAR VARIABLES STORE INTEGERS, FLOATS, CHARACTERS ETC IN C. A POINTER HOWEVER STORES **AN ADDRESS** IN MEMORY.

WE CAN SAY THAT A POINTER "POINTS TO" AN ADDRESS IN MEMORY



HOW DO WE SET UP radiusPtr TO POINT TO radius?

A POINTER TO AN INTEGER

ADDRESS OF THE VARIABLE

```
int *radiusPtr = &radius;
```

What types do pointer variables have?

Pointers can point to any type of variable - integer, char, boolean, float etc.

The types of pointers depend on what pointers point to
- pointer to integer (int* intPtr)
- pointer to char (char* charPtr)
- pointer to float (float* floatPtr)

If the pointer is just an address why does it need a type? Aren't all addresses the same type, just addresses?

How much space does stuff occupy?

Different types of variables occupy different amounts of space - an integer is typically 4 bytes, float is 4 bytes, a char is 1 byte and so on.

Pointers need to know what type they point to so we can do something interesting called “pointer arithmetic” - more on this later.

Every data type available in C, can have a corresponding pointer type which points to the address which holds a value of that data type.

COMMON OPERATIONS WITH POINTERS

```
int radius = 25;

int *radiusPtr = &radius;

int *otherRadiusPtr;

otherRadiusPtr = radiusPtr;
```

POINTER TO AN INTEGER, NOTE THAT WE SPECIFY THE TYPE TO WHICH THIS POINTER POINTS

THIS POINTER HOLDS THE ADDRESS OF RADIUS OR ON IN OTHER WORDS "POINTS TO" RADIUS

ANOTHER POINTER TO INTEGER

POINTERS ASSIGNED JUST LIKE ANY OTHER VARIABLE