CSCI 2410 Systems Programming

Fall 2026

Instructor: Jie Zhou

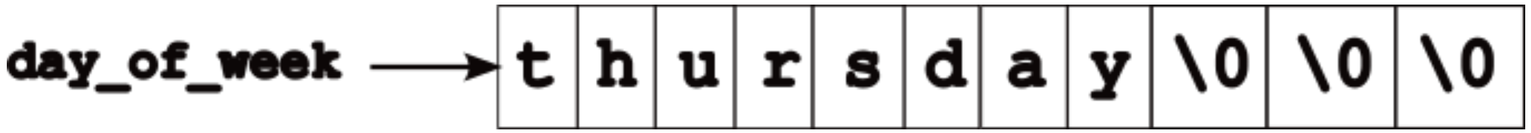
Department of Computer Science

George Washington University

Slides materials are mostly credited to Sibin Mohan and Gabriel Parmer at GWU.

Strings

- An array of characters that are null terminated, i.e., \0



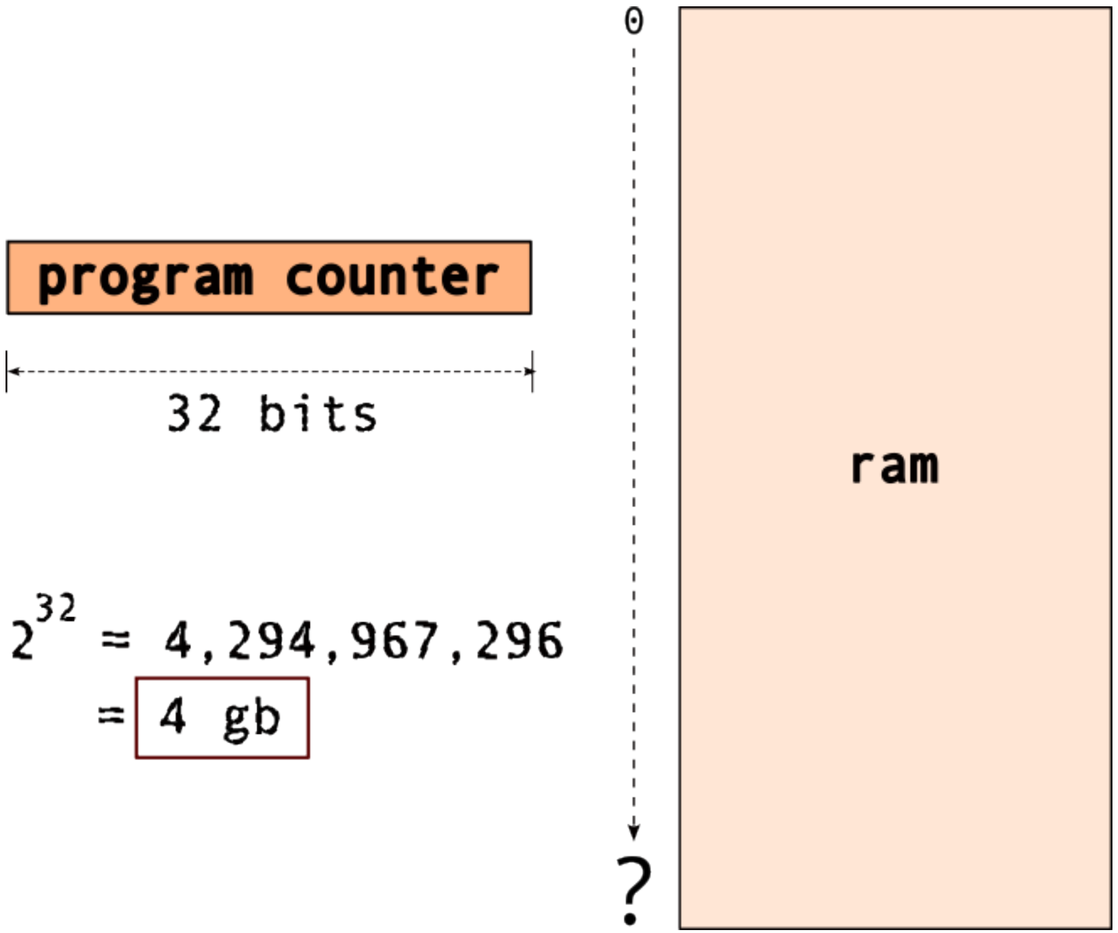
ASCII Table (7-bit)

Dec	Hex	Char	Description	Dec	Hex	Char	Description	Dec	Hex	Char	Description	Dec	Hex	Char	Description
0	00		[NULL]	32	20	(space)	[SPACE]	64	40	@		96	60	`	
1	01		[START OF HEADING]	33	21	!		65	41	A		97	61	a	
2	02		[START OF TEXT]	34	22	"		66	42	B		98	62	b	
3	03		[END OF TEXT]	35	23	#		67	43	C		99	63	c	
4	04		[END OF TRANSMISSION]	36	24	$		68	44	D		100	64	d	
5	05		[ENQUIRY]	37	25	%		69	45	E		101	65	e	
6	06		[ACKNOWLEDGE]	38	26	&		70	46	F		102	66	f	
7	07		[BELL]	39	27	'		71	47	G		103	67	g	
8	08		[BACKSPACE]	40	28	(		72	48	H		104	68	h	
9	09		[HORIZONTAL TAB]	41	29	)		73	49	I		105	69	i	
10	0A		[LINE FEED]	42	2A	*		74	4A	J		106	6A	j	
11	0B		[VERTICAL TAB]	43	2B	+		75	4B	K		107	6B	k	
12	0C		[FORM FEED]	44	2C	,		76	4C	L		108	6C	l	
13	0D		[CARRIAGE RETURN]	45	2D	-		77	4D	M		109	6D	m	
14	0E		[SHIFT OUT]	46	2E	.		78	4E	N		110	6E	n	
15	0F		[SHIFT IN]	47	2F	/		79	4F	O		111	6F	o	
16	10		[DATA LINK ESCAPE]	48	30	0		80	50	P		112	70	p	
17	11		[DEVICE CONTROL 1]	49	31	1		81	51	Q		113	71	q	
18	12		[DEVICE CONTROL 2]	50	32	2		82	52	R		114	72	r	
19	13		[DEVICE CONTROL 3]	51	33	3		83	53	S		115	73	s	
20	14		[DEVICE CONTROL 4]	52	34	4		84	54	T		116	74	t	
21	15		[NEGATIVE ACKNOWLEDGE]	53	35	5		85	55	U		117	75	u	
22	16		[SYNCHRONOUS IDLE]	54	36	6		86	56	V		118	76	v	
23	17		[END OF TRANS. BLOCK]	55	37	7		87	57	W		119	77	w	
24	18		[CANCEL]	56	38	8		88	58	X		120	78	x	
25	19		[END OF MEDIUM]	57	39	9		89	59	Y		121	79	y	
26	1A		[SUBSTITUTE]	58	3A	:		90	5A	Z		122	7A	z	
27	1B		[ESCAPE]	59	3B	;		91	5B	[		123	7B	{	
28	1C		[FILE SEPARATOR]	60	3C	<		92	5C	\		124	7C	|	
29	1D		[GROUP SEPARATOR]	61	3D	=		93	5D	]		125	7D	}	
30	1E		[RECORD SEPARATOR]	62	3E	>		94	5E	^		126	7E	~	
31	1F		[UNIT SEPARATOR]	63	3F	?		95	5F	_		127	7F		[DEL]

What is a Pointer?

The Pointer, sometimes called the English Pointer, is a medium-sized breed of pointing dog developed in England.

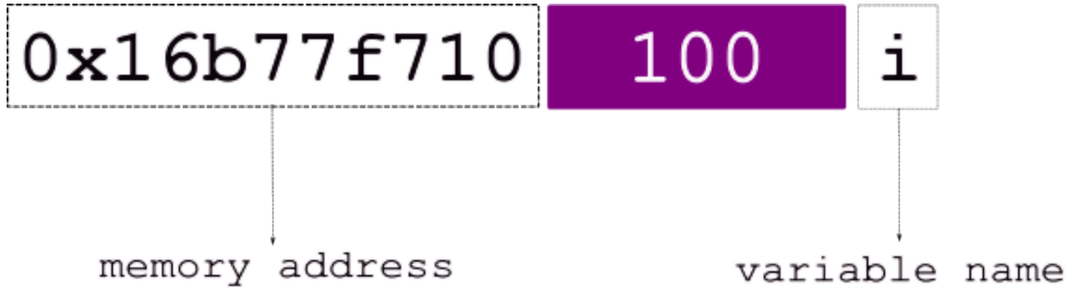
Memory Addressing



What happens when we declare a variable?

```
1   int i = 100 ;
```

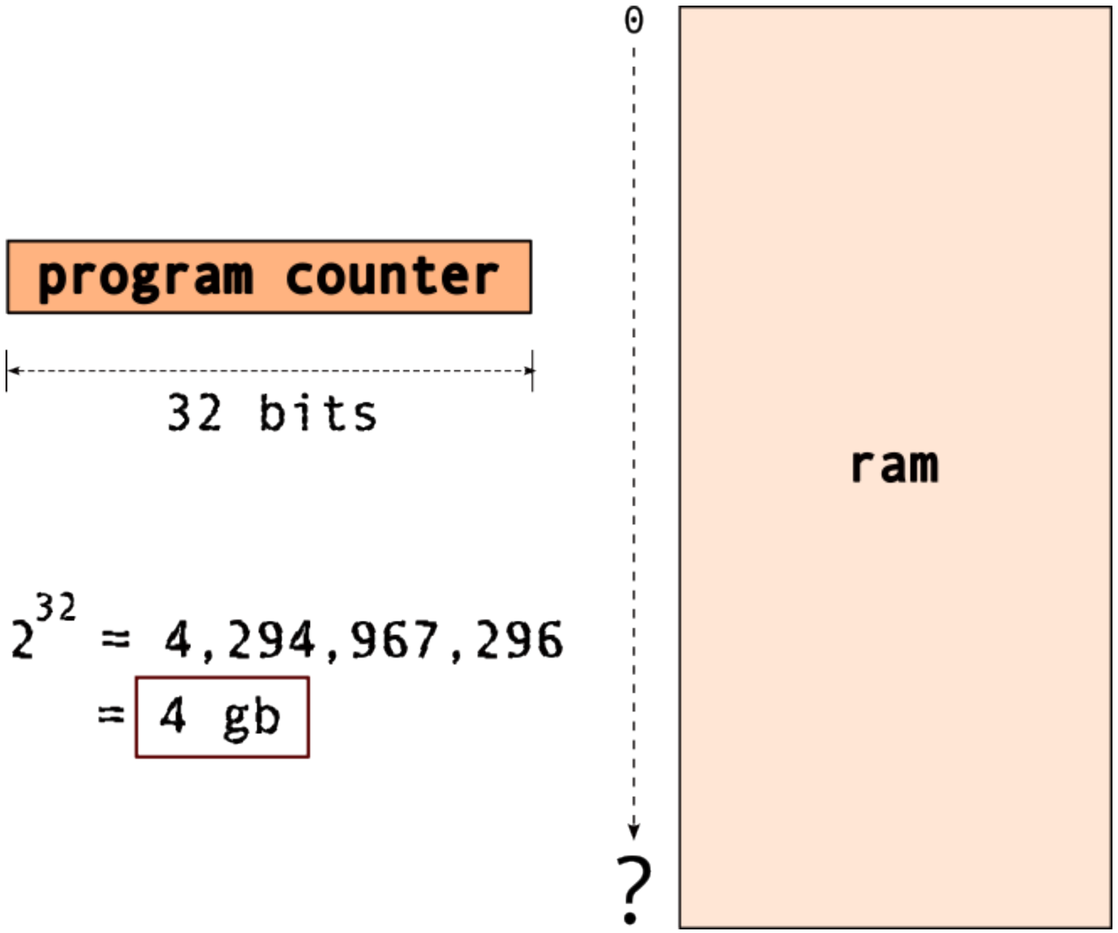
What happens when we declare a variable?



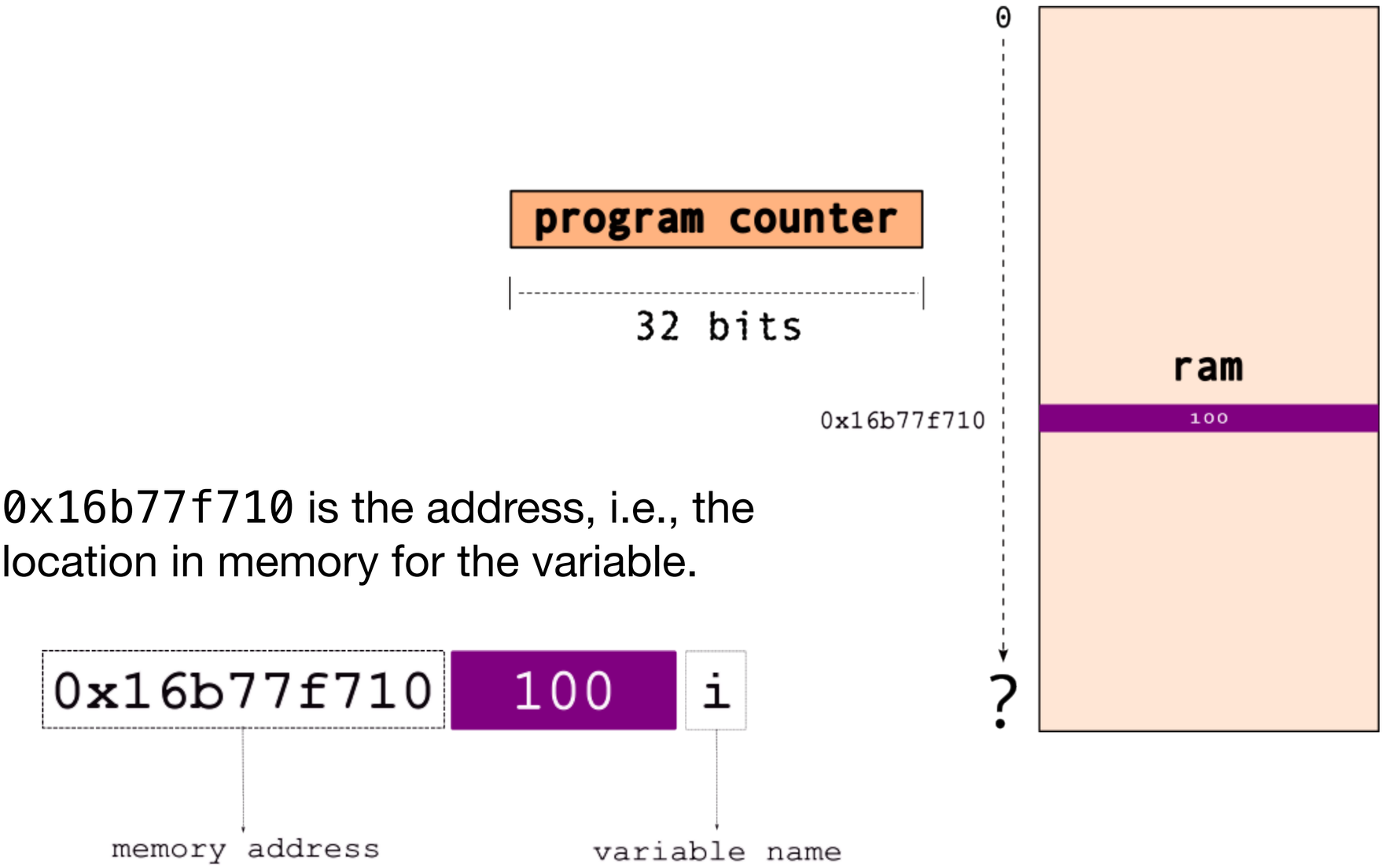
variable name	`i`
stored at address	`0x16b77f710`
value	`100`

What is a variable?

Memory Addressing

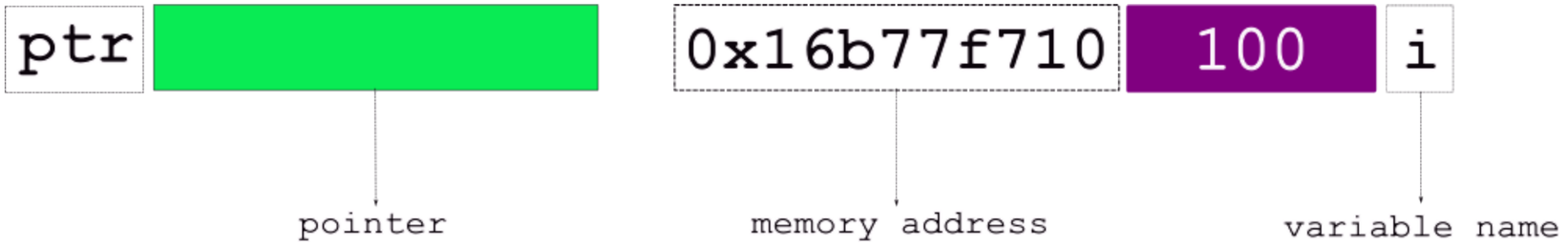


It could look something like:



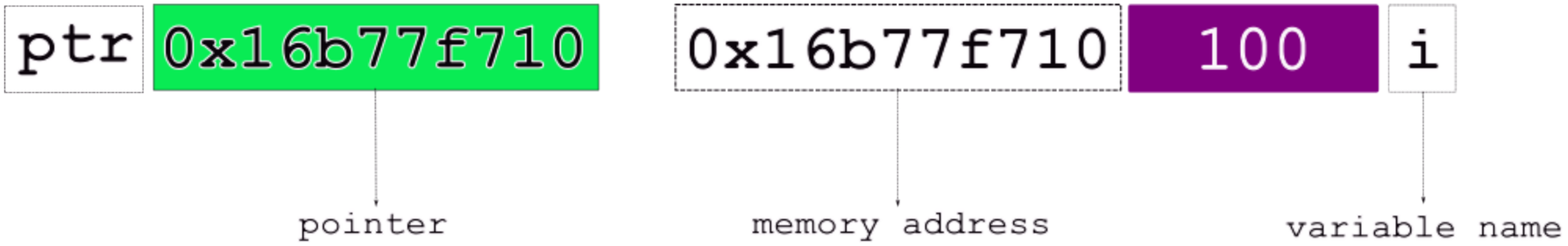
`0x16b77f710` is the address, i.e., the location in memory for the variable.

What is a pointer?



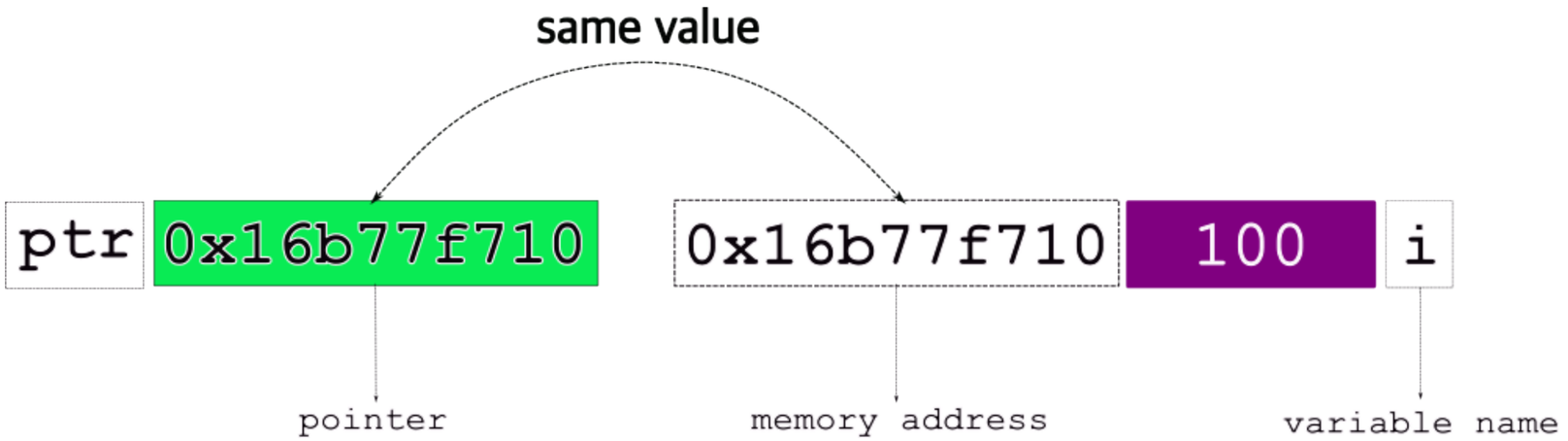
- A pointer *points to* another object.
  - In effect, points to a *memory location*.
- Pointers give programmers the ability to manipulate memory.
- In mainstream systems, C's pointers are implemented as an address.
- But this is not the only implementation option. Pointers can contain more.
  - e.g., a pointer may consists of an address and its referent's size.

What is a pointer?



- A pointer *points to* another object.
  - In effect, points to a *memory location*.
- Pointers give programmers the ability to manipulate memory.
- Here, `ptr` stores the **address** of the object it points to.

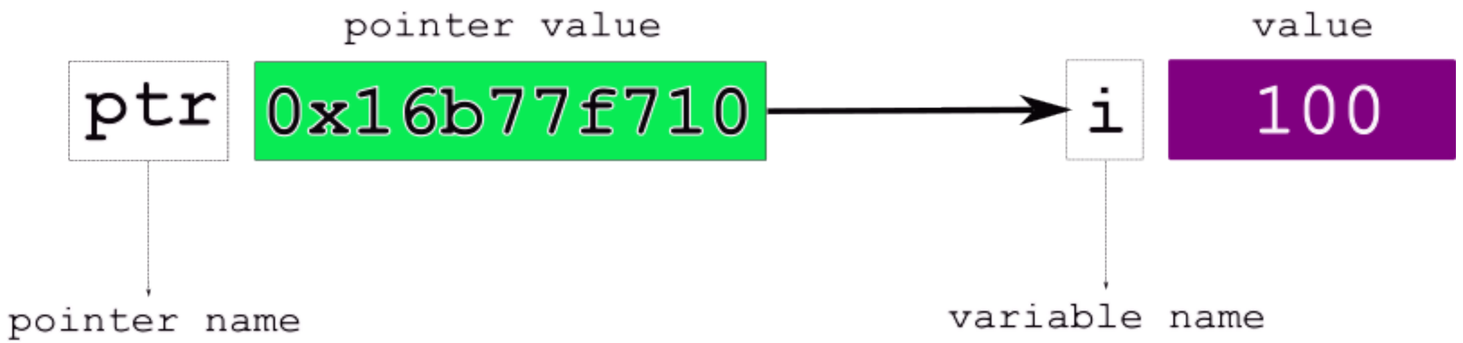
What is a pointer?



- A pointer *points to* another object.
  - In effect, points to a *memory location*.
- Pointers give programmers the ability to manipulate memory.
- Here, `ptr` stores the **address** of the object it points to.
- Hence, effectively,

Final View of A Pointer

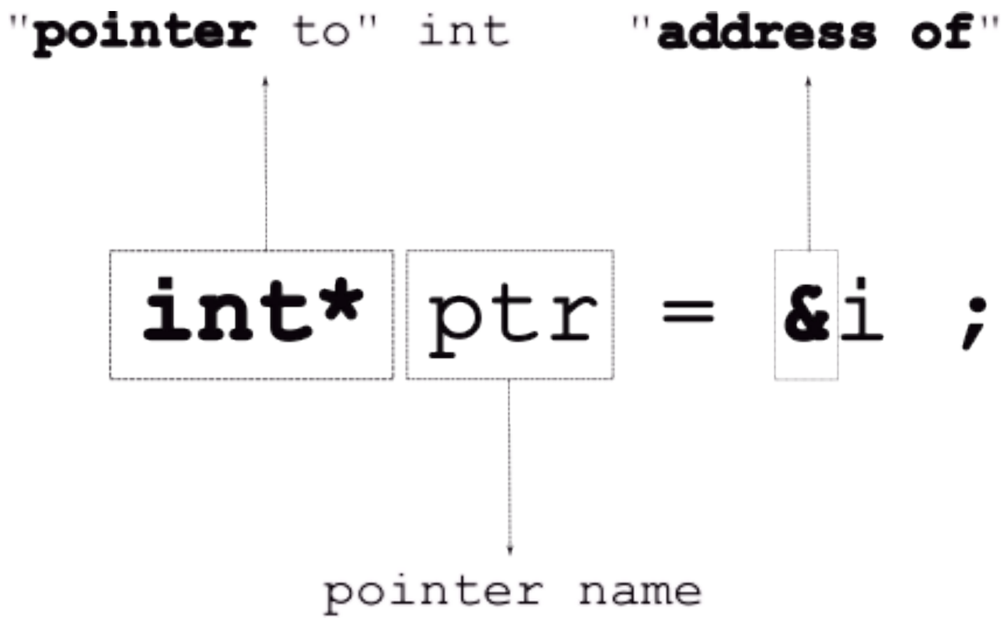
- `ptr` points to a variable `i`.



variable name	`i`
stored at address	`0x16b77f710`
value	`100`
point name	`ptr`
pointer value	`0x16b77f710`, i.e., address of `i`

Pointer Definition and Use

```
1   int i = 100 ;
2   int* ptr = &i ;
```



- The address-of operator ‘&’ takes a variable and returns its address.
- `printf(“%p”, ptr);` prints the address of i.
- The ‘*’ operator dereferences a pointer.
  - “`*ptr`” gives us 100.