

The background features a vibrant, abstract design with flowing, wavy lines in shades of blue, teal, and purple. Overlaid on this is a faint network diagram consisting of nodes (small circles) and connecting lines, suggesting a data structure or memory map.

POINTERS AND ARRAYS MEMORY MAP EXAMPLE

POINTERS AND MEMORY

- **memory cell** an individual storage location in memory
- **address of a memory cell** the relative position of a memory cell in the computer's main memory
- **contents of a memory cell** the information stored in a memory cell, either a *program instruction* or *data*
- **stored program concept** a computer's ability to store program instructions in main memory for execution

pointer (pointer variable)

- a memory cell that stores the address of a data item
- syntax: *type *variable*

```
int    m    = 25;  
int    *itemp;    /* a pointer to an integer */
```

DECLARING A POINTER

- Must be assigned a specific data type to point to

```
int *numPtr; //identifier ready to point to an integer
char *letterPtr; //identifier ready to point to a character
double *amtPtr; //identifier ready to point to a double
```

- Declaring a pointer just provides an identifier (name) for the pointer but it **points to nothing.**
- To use the pointer we must point it to a variable location.
- The data types must match.

```
int *numPtr; //identifier ready to point to an integer
int number = 25;
numPtr = &number; //pointers must be initialized using & and the variable name
```

ARRAYS

data structure a composite of related data items stored under the same name

array a collection of data items of the same type

Array declaration is a pointer allocates and points to the first element in the array

array element a data item that is part of an array

subscripted variable a variable followed by a subscript in brackets, designating an array element

$x[0]$

```
double x[8];
```

Array x

x[0] x[1] x[2] x[3] x[4] x[5] x[6] x[7]

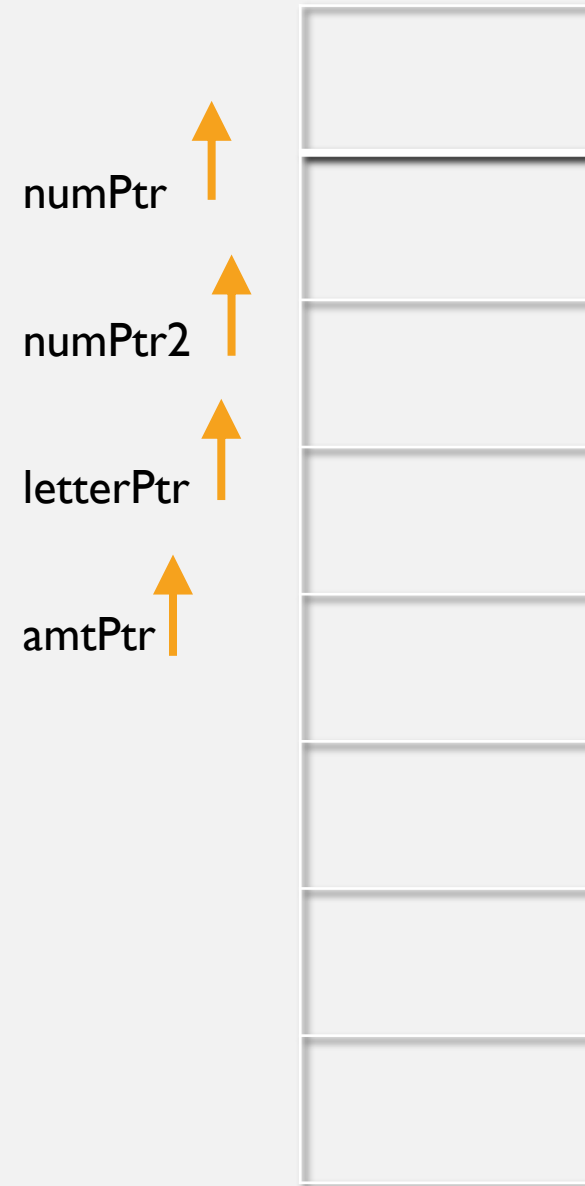
16.0	12.0	6.0	8.0	2.5	12.0	14.0	-54.5
------	------	-----	-----	-----	------	------	-------

POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer  
int *numPtr2; //integer pointer
```

```
char *letterPtr; //character pointer  
double *amtPtr; //double pointer
```

NOTE: The pointers are declared but they do not point to anything valid yet



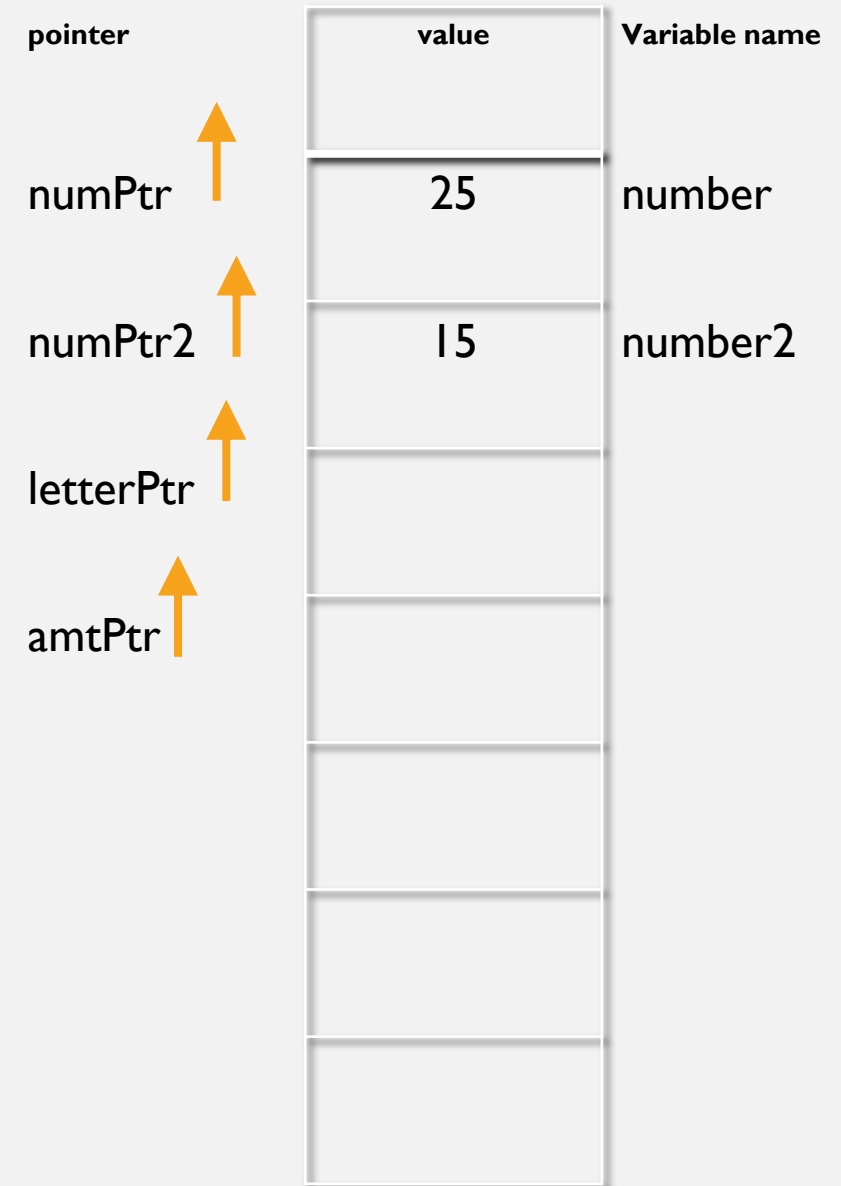
POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer  
int *numPtr2; //integer pointer
```

```
char *letterPtr; //character pointer  
double *amtPtr; //double pointer
```

NOTE: The pointers are declared but they do not point to anything valid yet

```
int number = 25, number2 = 15;
```

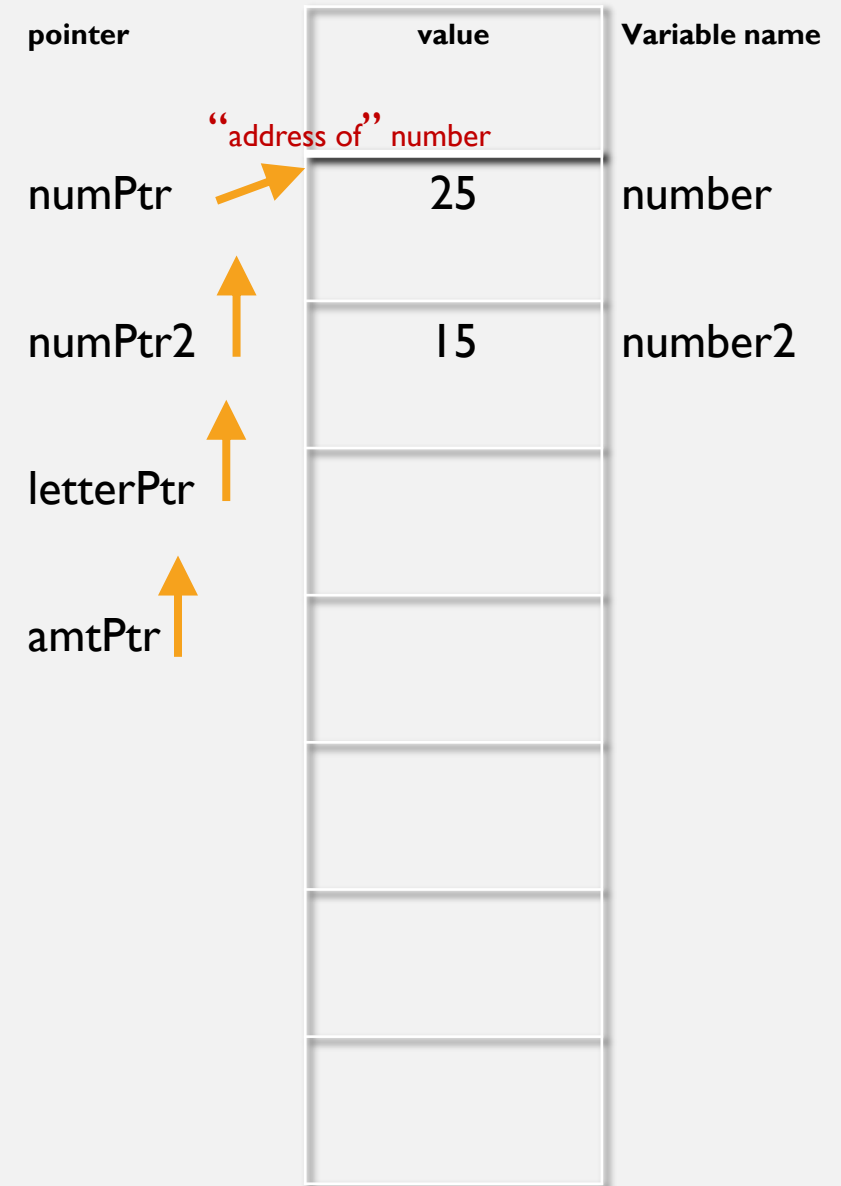


POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
int *numPtr2; //integer pointer

char *letterPtr; //character pointer
double *amtPtr; //double pointer

int number = 25, number2 = 15;
numPtr = &number; //numPtr = "address of" number
```

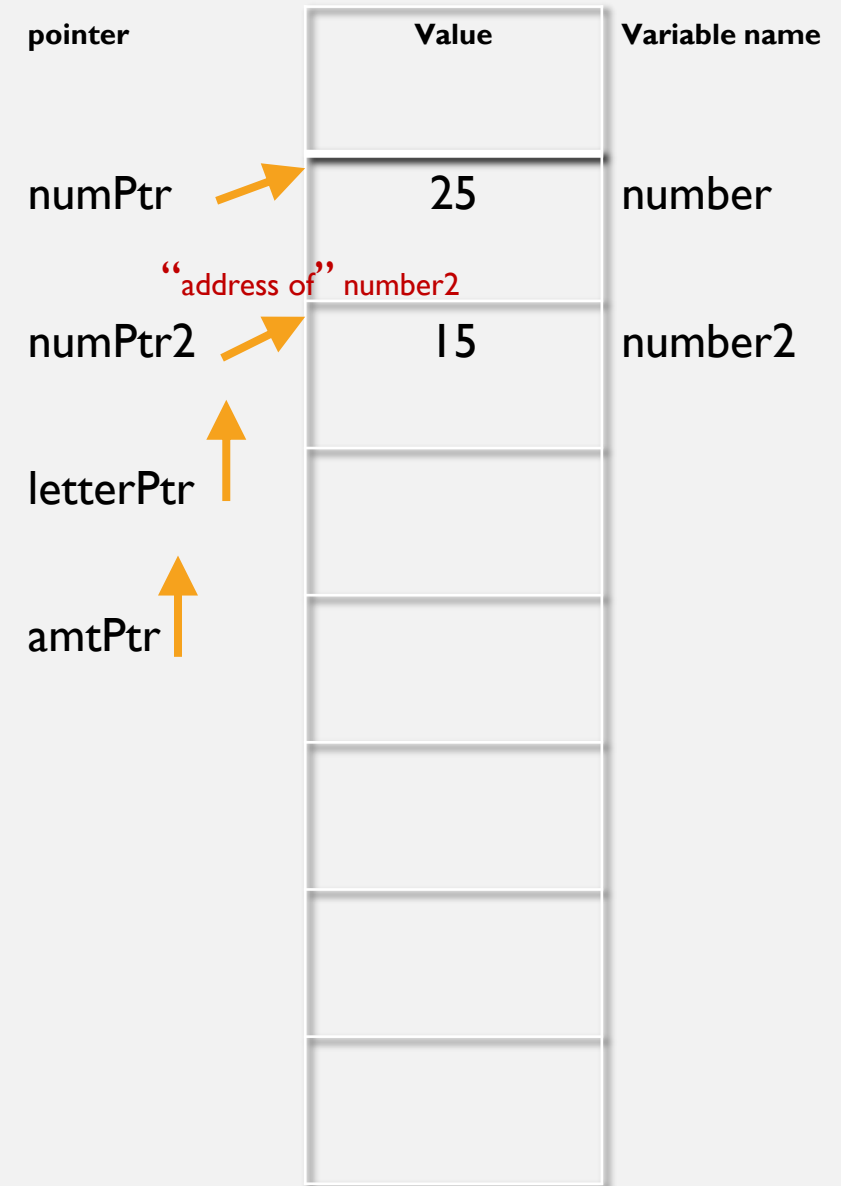


POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
int *numPtr2; //integer pointer

char *letterPtr; //character pointer
double *amtPtr; //double pointer

int number = 25, number2 = 15;
numPtr = &number; //numPtr = "address of" number
numPtr2 = &number2; //numPtr2 = "address of" number2
```



POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
int *numPtr2; //integer pointer

char *letterPtr; //character pointer
double *amtPtr; //double pointer

int number = 25, number2 = 15;
numPtr = &number; //numPtr = "address of" number
numPtr2 = &number2; //numPtr2 = "address of" number2

char letter = 'X';
letterPtr = &letter; //letterPtr = "address of" letter

double amount = 55.5;
amtPtr = &amount; //amtPtr = "address of" amount
```

pointer		Value	Variable name <i>*reference</i>
numPtr	→	25	number
numPtr2	→	15	number2
letterPtr	→ "address of" letter	'X'	letter
amtPtr	→ "address of" amount	55.5	amount

POINTER EXAMPLE MEMORY MAP





```
int *numPtr; //integer pointer
int *numPtr2; //integer pointer

char *letterPtr; //character pointer
double *amtPtr; //double pointer

int number = 25, number2 = 15;
numPtr = &number; //numPtr = "address of" number
numPtr2 = &number2; //numPtr = "address of" number2

char letter = 'X';
letterPtr = &letter; //letterPtr = "address of" letter

double amount = 55.5;
amtPtr = &amount; //amtPtr = "address of" amount
```

pointert		Value	Variable name <i>*reference</i>
numPtr		25	number
numPtr2		15	number2
letterPtr		'X'	letter
amtPtr		55.5	amount

POINTER EXAMPLE MEMORY MAP





```
int *numPtr; //integer pointer
int *numPtr2; //integer pointer

char *letterPtr; //character pointer
double *amtPtr; //double pointer

int number = 25, number2 = 15;
numPtr = &number; //numPtr = "address of" number
numPtr2 = &number2; //numPtr2 = "address of" number2

char letter = 'X';
letterPtr = &letter; //letterPtr = "address of" letter

double amount = 55.5;
amtPtr = &amount; //amtPtr = "address of" amount
```

pointer		Value	Variable name <i>*reference</i>
numPtr &number		25	number <i>*numPtr</i>
numPtr2 &number2		15	number2 <i>*numPtr2</i>
letterPtr &letter		'X'	letter <i>*letterPtr</i>
amtPtr &amount		55.5	amount <i>*amtPtr</i>

POINTER EXAMPLE MEMORY MAP

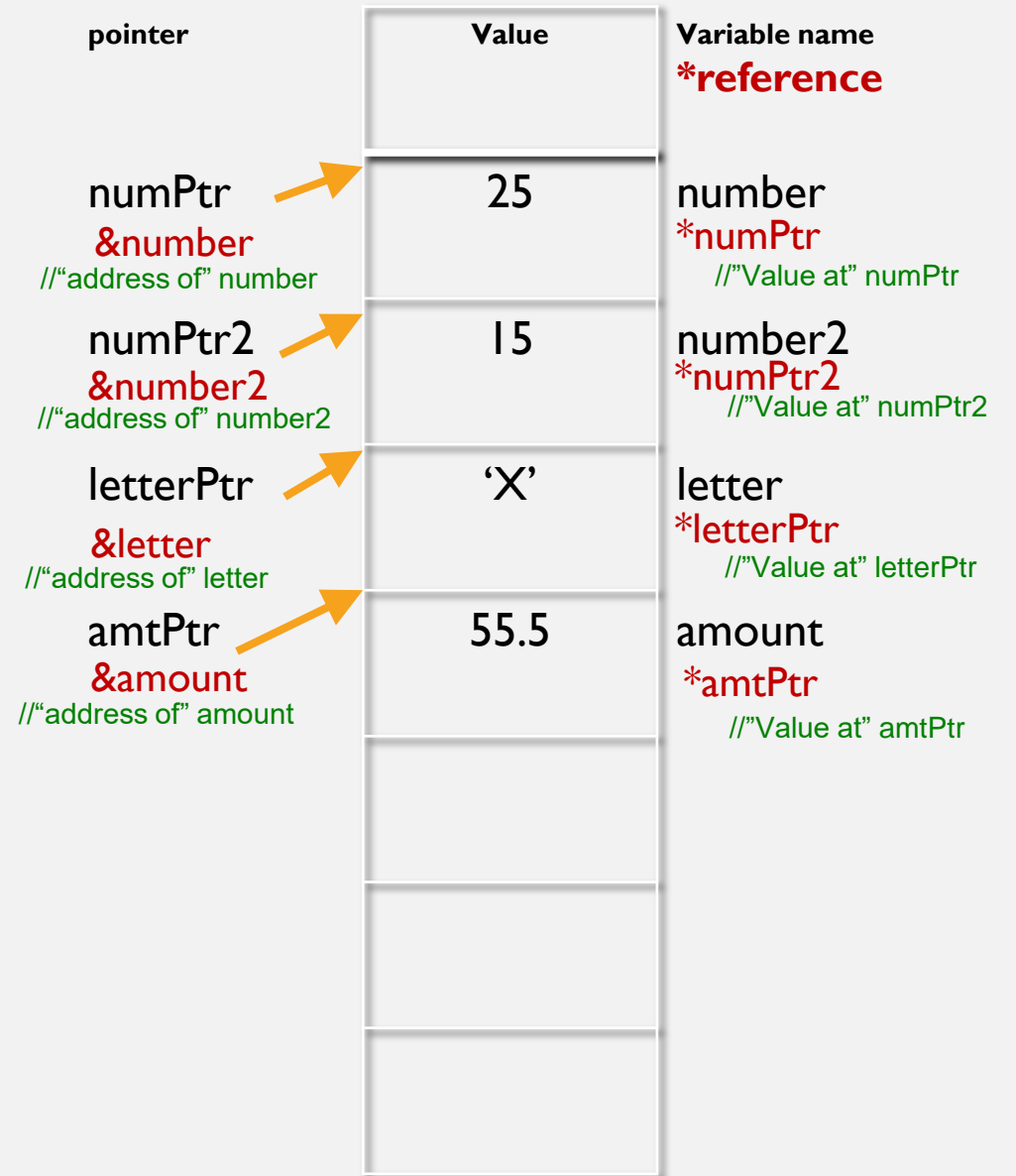
```
int *numPtr; //integer pointer
int *numPtr2; //integer pointer
```

```
char *letterPtr; //character pointer
double *amtPtr; //double pointer
```

```
int number = 25, number2 = 15;
numPtr = &number; //numPtr = "address of" number
numPtr2 = &number2; //numPtr2 = "address of" number2
```

```
char letter = 'X';
letterPtr = &letter; //letterPtr = "address of" letter
```

```
double amount = 55.5;
amtPtr = &amount; //amtPtr = "address of" amount
```



POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

pointer

numPtr
&number



Value
25

Variable name
***reference**

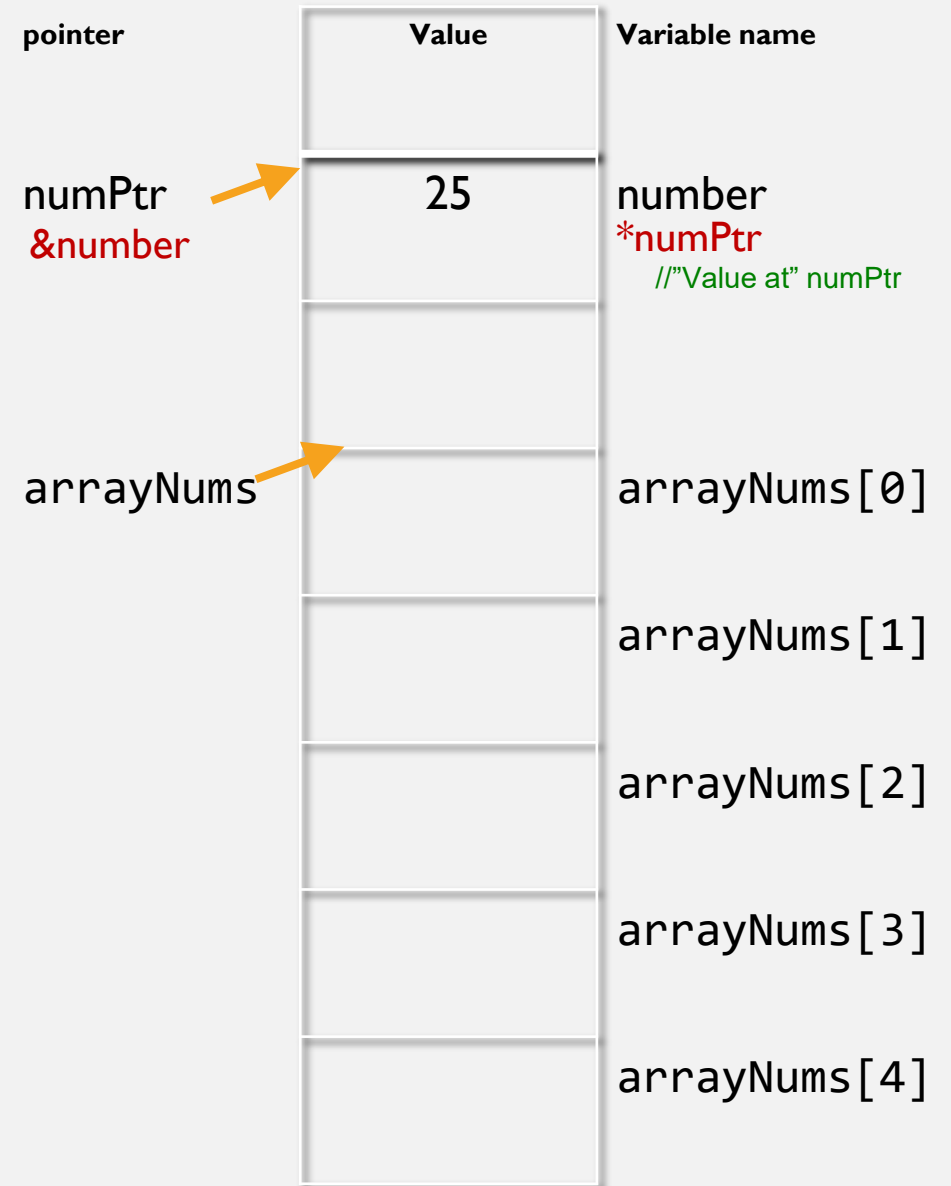
number
***numPtr**
// "Value at" numPtr

ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];
```

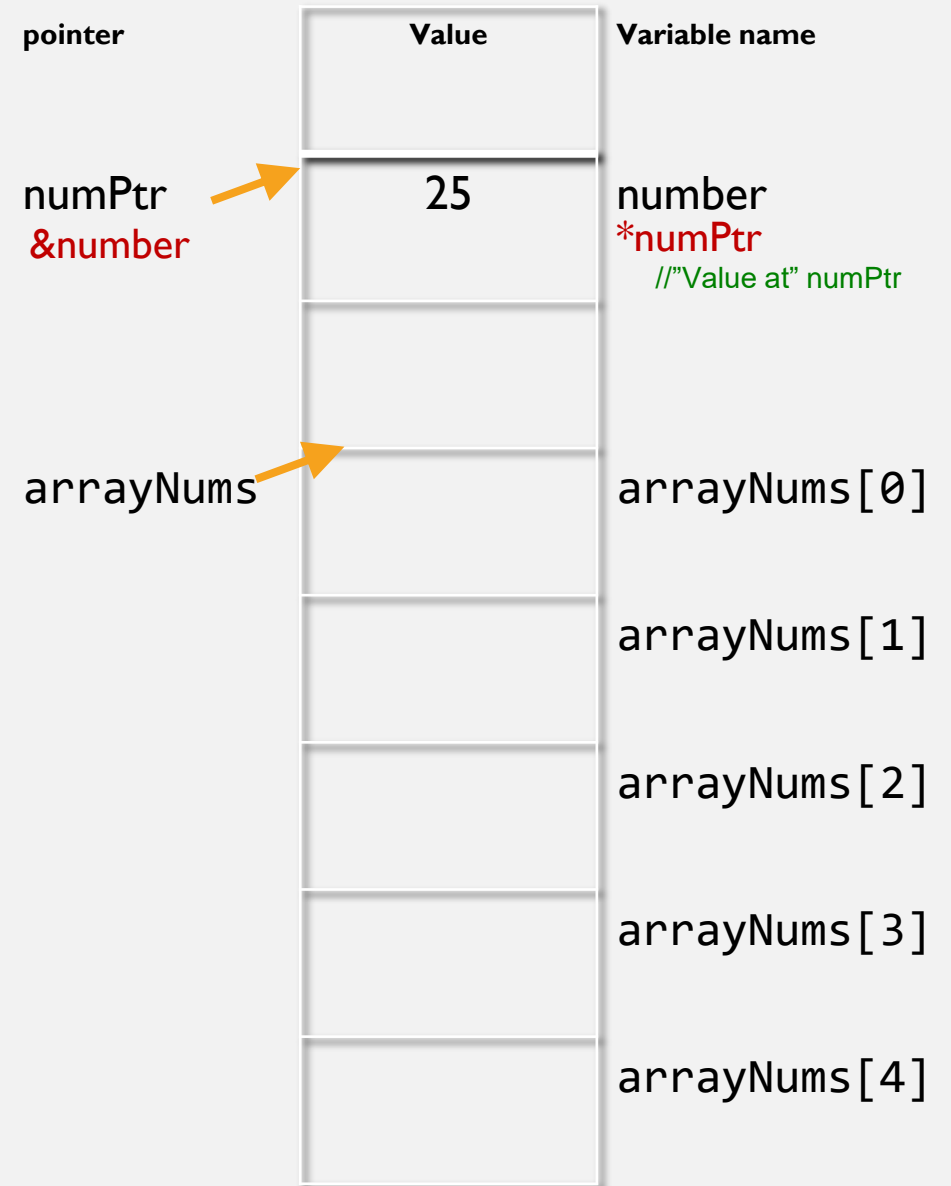


ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];
```

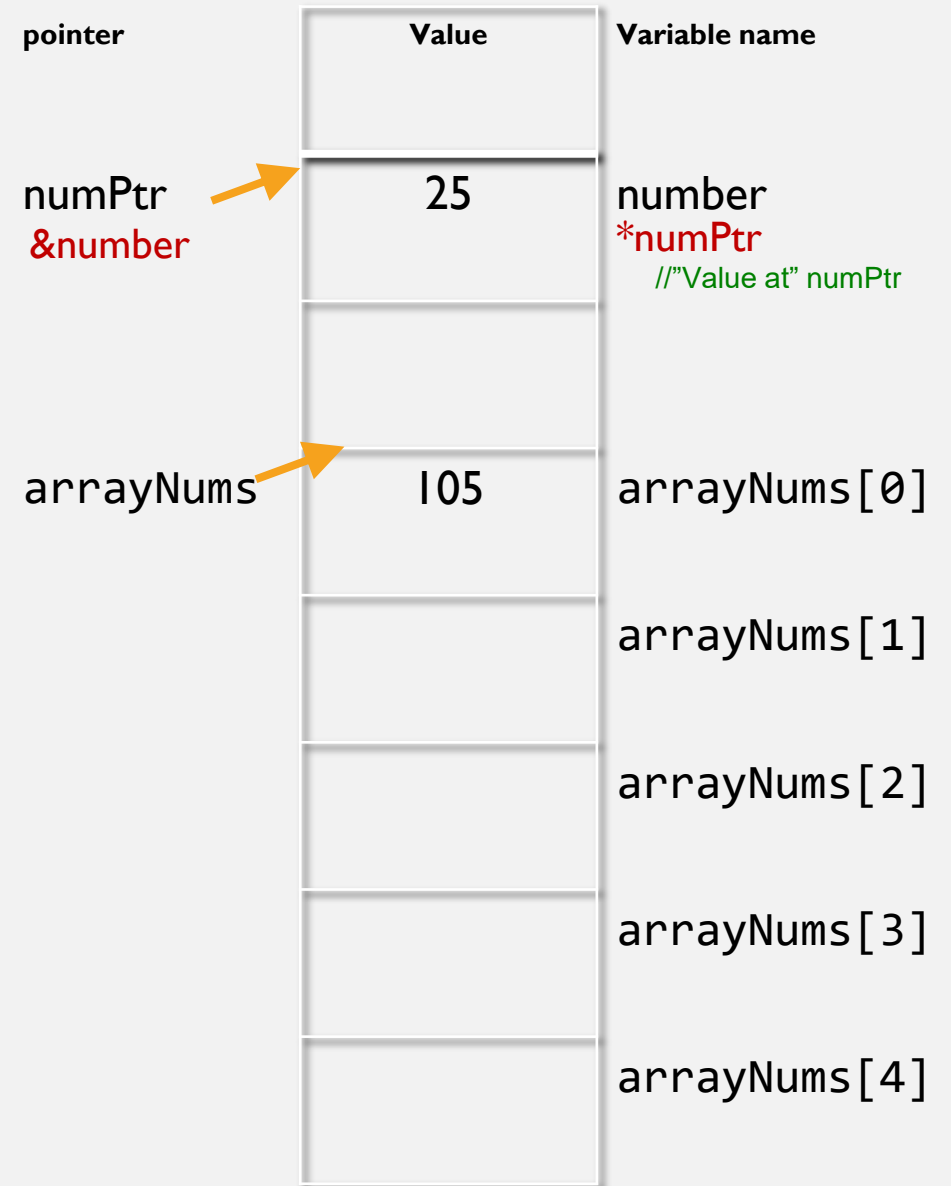


ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;
```

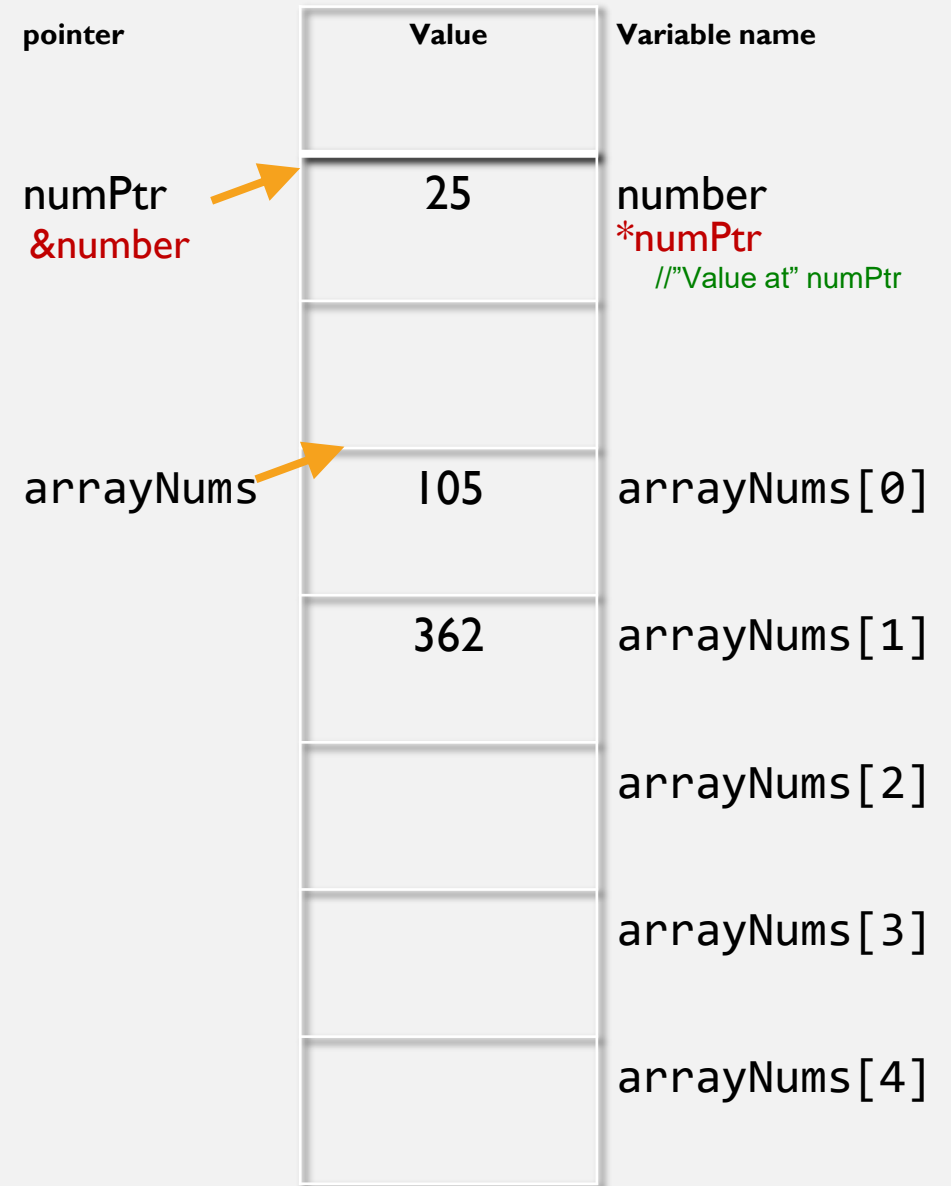


ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;  
arrayNums[1] = 362;
```



ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;  
arrayNums[1] = 362;  
arrayNums[2] = 9;
```

pointer	Value	Variable name
numPtr &number	25	number *numPtr // "Value at" numPtr
arrayNums	105	arrayNums[0]
	362	arrayNums[1]
	9	arrayNums[2]
		arrayNums[3]
		arrayNums[4]

ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;  
arrayNums[1] = 362;  
arrayNums[2] = 9;  
arrayNums[3] = arrayNums[0] + 3;
```

pointer	Value	Variable name
numPtr &number	25	number *numPtr // "Value at" numPtr
arrayNums	105	arrayNums[0]
	362	arrayNums[1]
	9	arrayNums[2]
	108	arrayNums[3]
		arrayNums[4]

ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;  
arrayNums[1] = 362;  
arrayNums[2] = 9;  
arrayNums[3] = arrayNums[0] + 3;  
arrayNums[4] = 900;
```

pointer	Value	Variable name
numPtr &number	25	number *numPtr // "Value at" numPtr
arrayNums	105	arrayNums[0]
	362	arrayNums[1]
	9	arrayNums[2]
	108	arrayNums[3]
	900	arrayNums[4]

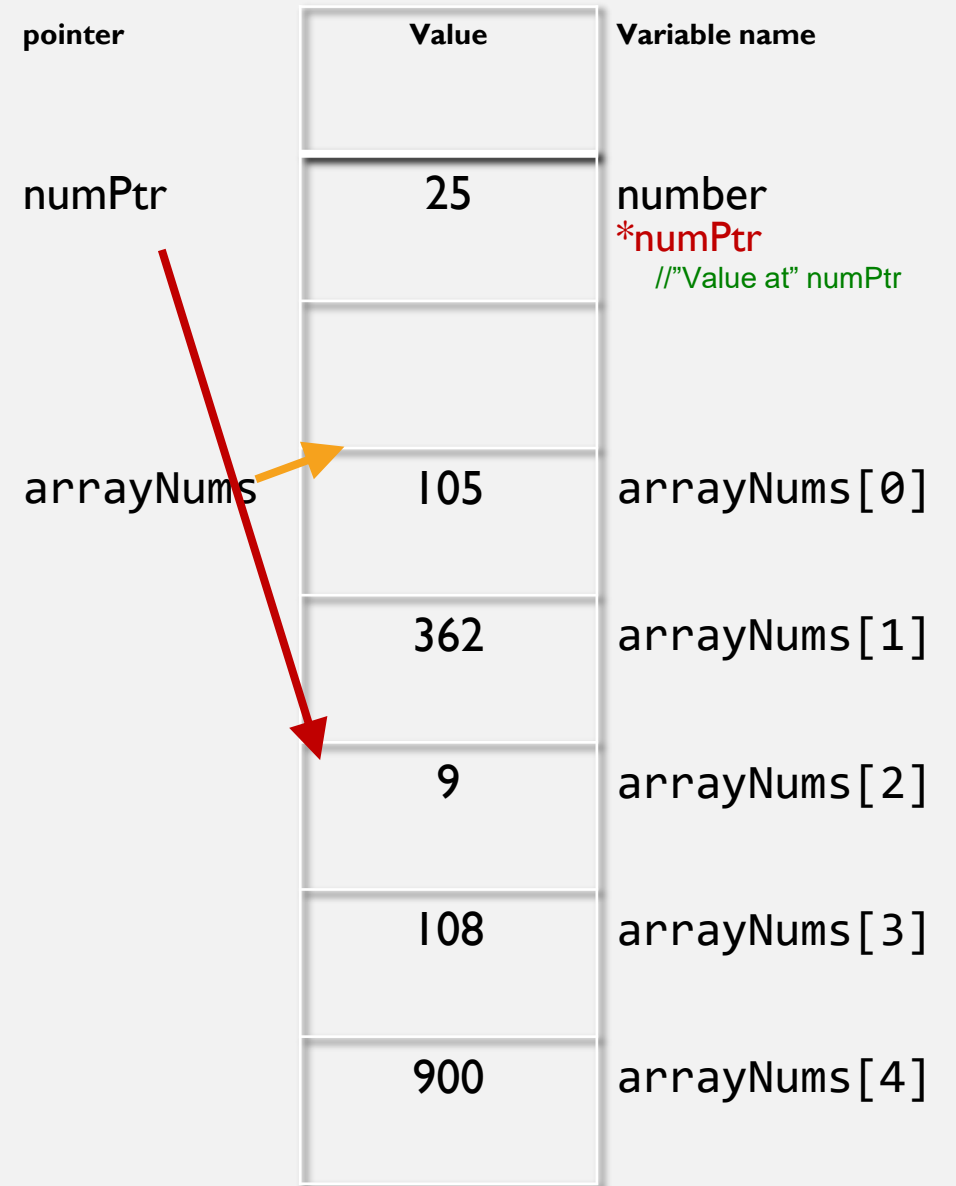
ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;  
arrayNums[1] = 362;  
arrayNums[2] = 9;  
arrayNums[3] = arrayNums[0] + 3;  
arrayNums[4] = 900;
```

```
numPtr = &arrayNums[2];
```



ARRAY AND POINTER EXAMPLE MEMORY MAP

```
int *numPtr; //integer pointer
```

```
int number = 25;  
numPtr = &number;
```

```
int arrayNums[5];  
arrayNums[0] = 105;  
arrayNums[1] = 362;  
arrayNums[2] = 9;  
arrayNums[3] = arrayNums[0] + 3;  
arrayNums[4] = 900;
```

```
numPtr = &arrayNums[2];  
*numPtr = 10109;
```

