

# POINTERS IN C, PASSING POINTERS TO FUNCTIONS

# Parameter Passing in Python

- In Python, parameters are passed two ways:
  - ▣ For numbers, a copy of the number is passed to the function
  - ▣ For mutable objects (like lists), a **reference** to the object is passed to the function

```
def swapInts(x, y):
```

```
    x,y = y,x
```

```
x,y = 2, 5
```

```
swapInts (x, y)
```

```
def swapListElements(alist, i, j):
```

```
    alist[i], alist[j] = alist[j], alist[i]
```

```
alist = [3, 4, 5, 6]
```

```
swapListElements(alist, 1, 3)
```

aList



# References in C

- In C, you can obtain a reference to **any** variable.
  - ▣ These references are called **pointers**.
  - ▣ By “reference”, we mean the *address or memory location* of the variable.
- If we pass a variable’s address as a parameter to a function, the function can change the value of that variable.
- Overview:
  - ▣ To get an address, use &
  - ▣ To get a variable referenced by a pointer, use \*
  - ▣ To declare a pointer variable, use \*

# Visualizing Pointers

## Box and Pointer Diagrams

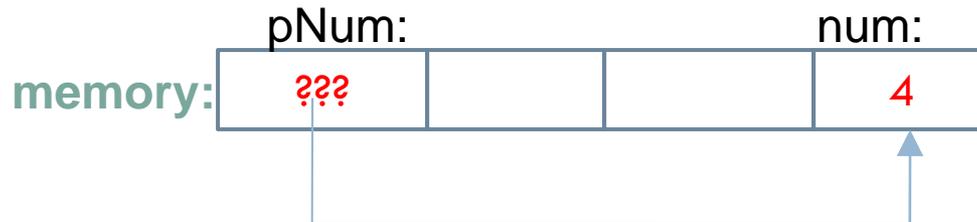
```
int num = 4;
```

```
int *pNum;
```

```
pNum = &num;
```

Use the **\*** in a **declaration** to indicate that a variable is a pointer.

Use the **&** in an **expression** to get the address of a variable.



How do we obtain the value to which pNum refers (a.k.a. the “pointee”)?

# Visualizing Pointers – Part 2

```
int num = 4;
```

```
int *pNum;
```

```
pNum = &num;
```

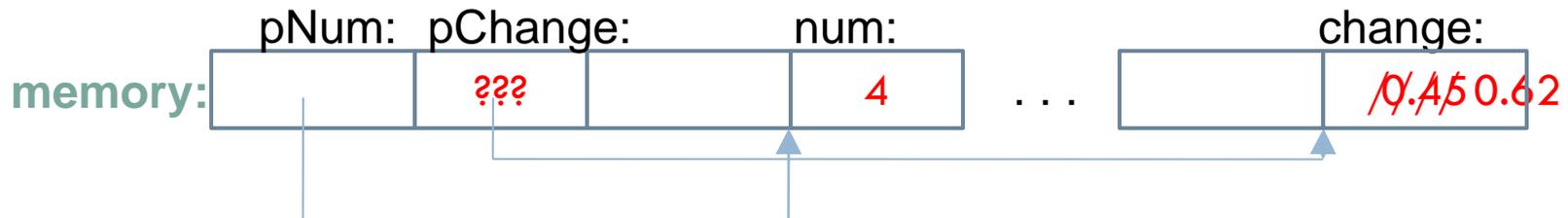
```
double change = 0.45;
```

```
double *pChange = &change;
```

We can declare and initialize a pointer in a single statement.

```
*pChange = .62;
```

Use the **\*** in an **expression** to get the value referenced by a pointer.



# Summary of Pointers

- Example of a pointer variable: `int *pNum;`
- Example of a integer variable: `int num;`
- Assigning a value to an int: `num = 4;`
- Obtaining the address of a variable: `&num`
- Assigning an address to a pointer variable:  
`pNum = &num;`
- Assigning a value to the variable to which a  
pointer variable points:  
`*pNum = 7;`

# Here's Binky!

- Ignore malloc for now
- Vocabulary
  - ▣ Pointee: the thing referenced by a pointer
  - ▣ Dereference: obtain the pointee
- See <http://cslibrary.stanford.edu/104/>
- What name did we give pointer “sharing” in Python?

# Proof that pointers store memory locations

- Checkout the **PointersInClass** project.
- Run it in the debugger
  - ▣ The console is a separate window
  - ▣ It automatically inserts a breakpoint at the start of main()
- Let's start quiz questions 6-8 together

# Using pointers with functions

- We claimed earlier that if we passed a variable's reference as a parameter to a function, the function could change that variable.
- Reminder:
  - ▣ To get an address, use &
  - ▣ To get a variable referenced by a pointer, use \*
  - ▣ To declare a pointer variable, use \*

# An example together

- In Eclipse, run `downAndUp`
- Change the function and how it's called so that it works!
- When you are done, please answer the quiz question.

# A simple example for reference

```
□ void foo(int *a){  
    *a = 7;  
    printf("%d\n", *a);  
}
```

Receive an address

Modify value at address

```
int b = 3;  
foo(&b);  
printf("%d\n", b);
```

Send the address of b

# Practice with Pointers

```
1.     int x = 3, y = 5;
2.     int *px = &x;
3.     int *py = &y;
4.     printf("%d %d\n", x, y);
5.     *px = 10;
6.     printf("%d %d\n", x, y); /* x is changed */
7.     px = py;
8.     printf("%d %d\n", x, y); /* x not changed */
9.     *px = 12;
10.    printf("%d %d\n", x, y); /* y is changed */
```

# Pointer Pitfalls

- Don't try to dereference an unassigned pointer:
  - `int *p;`  
`*p = 5;`
  - `/* immediate crash! */`
- Pointer variables must be assigned *address* values.
  - `int x = 3;`  
`int *p;`  
`p = x;`
  - `/* eventual crash */`
- Be careful how you increment
  - `*p += 1;`      `/* is not the same as ... */`
  - `*p++;`

# In-class exercise on pointer pitfalls

- Turn in part 1 of the quiz.
- The rest of today's quiz lets you see some pointer pitfalls in action. These make great exam questions!
  - ▣ Do it now
- When you are done, start the homework:
  - ▣ A **written portion** (box and pointer diagrams)
  - ▣ More pointer output
  - ▣ Writing functions to change variables
    - doubleMe
    - swap
  - ▣ scanf revisited