SYSC 2006
Fall 2019

Carleton UNIVERSITY

Canada's Capital University

Pointers and Functions

Last revised, C.-H. Lung, Oct. 3, 2019

A Solution? Or … a Problem?

Here's a function that's supposed to swap its arguments:

```
void swap(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

What is the problem?

A Problem

- Typical call:

```
int a = 5, b = 10;  // Point A
swap(a, b);
```

- When the function returns, a and b are **unchanged**

To illustrate the problem, trace the code line-by-line from *`Point A`* until swap returns, drawing memory diagrams depicting the activation frames of swap and its caller

A Problem

- Why doesn't the function work?

- **Call-by-value parameter passing**: parameters x and y and arguments a and b are located in different activation records

- Changing the values stored in x and y doesn't modify the contents of a and b

Pass-By-Value Arguments

- Assigning a value to a parameter does not modify the corresponding argument
- This can lead to "tighter" code
- Revise the `swap` function to make it work

Fixing the Problem

- swap could modify variables in the caller's activation frame if C supported pass-by-reference parameters (C++ does)
  - while the function executed, parameters x and y would refer or point to the corresponding arguments; i.e., assigning a value to x would update the contents of the function's first argument
- With C, we can achieve the effect of pass-by-reference parameters by explicitly ***passing addresses of variables*** as function arguments

  - Addresses of variables means pointers

Addresses as Arguments

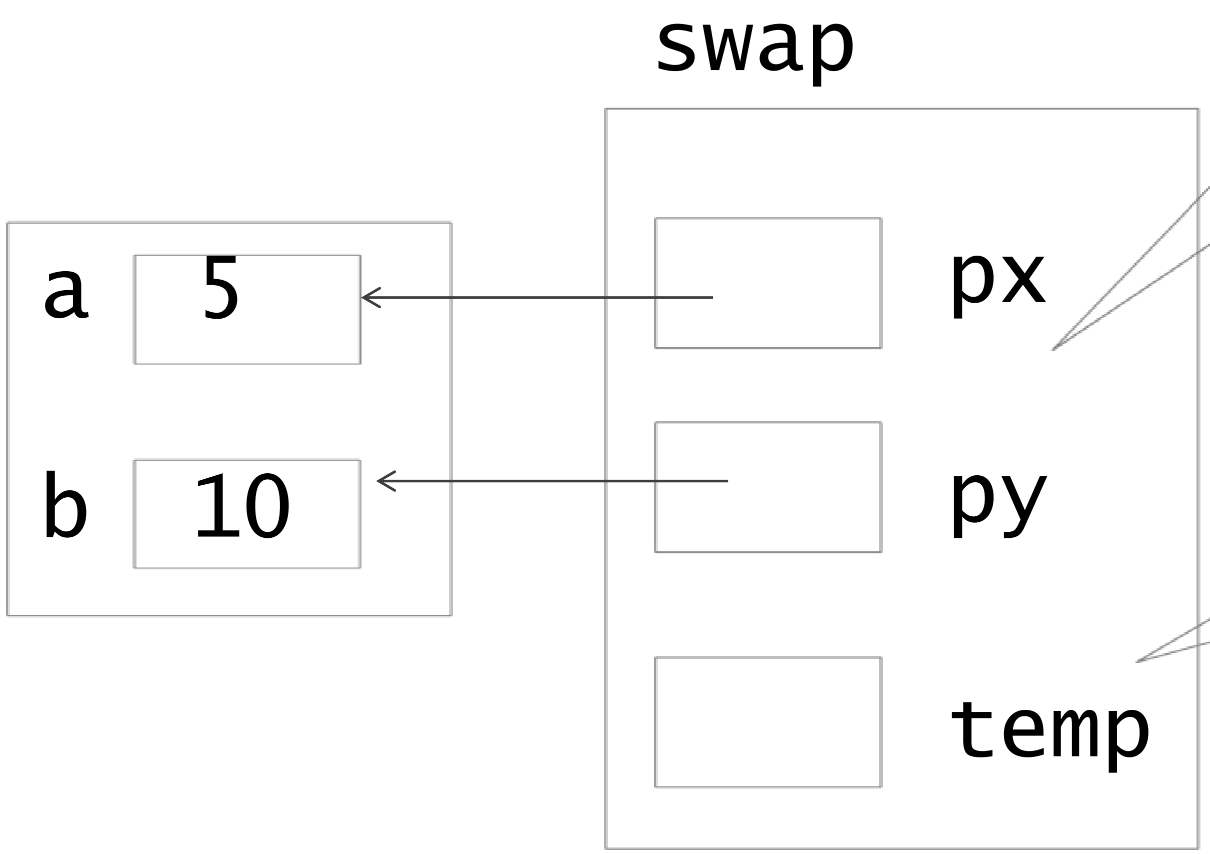
- Pass the ***addresses*** of a and b to swap.
- How to get the addresses of a and b?

```
    int a = 5, b = 10;
    swap(&a, &b);

void swap(?? px, ??py)
```

- Parameter names are px and py – how should they be declared?

Addresses as Arguments



What are the types of px and py?

What is the type of `temp`?

Revised Swap

- Revise swap so that the parameters are ***pointers*** to integers:

```
void swap(int *px, int *py)
{
    int temp;

    temp = *px;
    *px = *py;
    *py = temp;
}
```

Memory Diagram Exercise

```
int a = 5, b = 10; // Point A
swap(&a, &b);
```

To illustrate how `swap` correctly exchanges the contents of a and b, trace the code line-by-line from `Point A` until `swap` returns, drawing memory diagrams depicting the activation frames of `swap` and its caller

Memory Diagram Exercise

Why is this version of swap incorrect? Draw memory diagrams to determine the problem!

```
void swap(int *px, int *py)
{
    int *temp;

    *temp = *px;
    *px = *py;
    *py = *temp;
}
```

Memory Diagram Exercise

Why is this version of swap incorrect? Draw memory diagrams to determine the problem!

```
void swap(int *px, int *py)
{
    int temp;

    temp = &px;
    *px = *py;
    *py = temp;
}
```