Review

- Compared to Python, a few more things need to be considered to play with functions in C
  - You need to write **data types** of its parameters and return value explicitly in the function header
  - You can **declare** a function using its prototype before its full definition

- C uses **stack** to manage memory operation of a function call
  - We have seen why using stack (LIFO data structure) makes sense
  - We have seen how all the information is pushed and popped correctly using only the two pointers (stack pointer and frame pointer)

Pointers – Motivation

Lecture 29-1

Hyung-Sin Kim

SNU Graduate School of Data Science

Practice – Call by Value

- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```

Let's type, compile, and execute!

What do you see on your screen?
Does swapping happen as intended?

Can you explain why you see such results by using memory operation we learned last time?

Practice – Call by Value (Solution)

- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```

Local variables firstVal and secondVal **die** when function swap ends!

Swapping does not happen!
(well… not surprising because swap function does not return anything)

Let's see what happens in the memory

Practice – Call by Value (Memory)

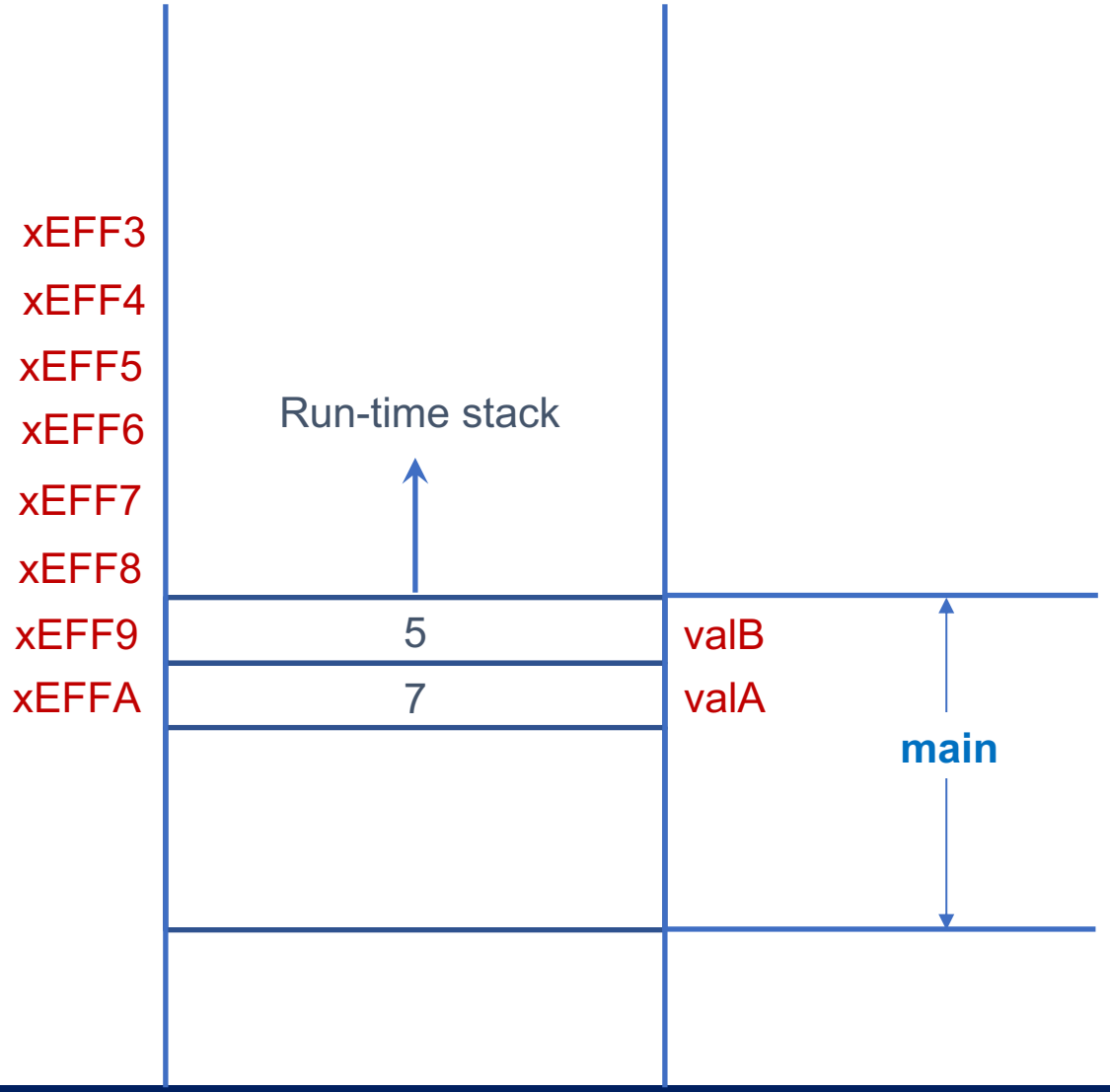
- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Practice – Call by Value (Memory)

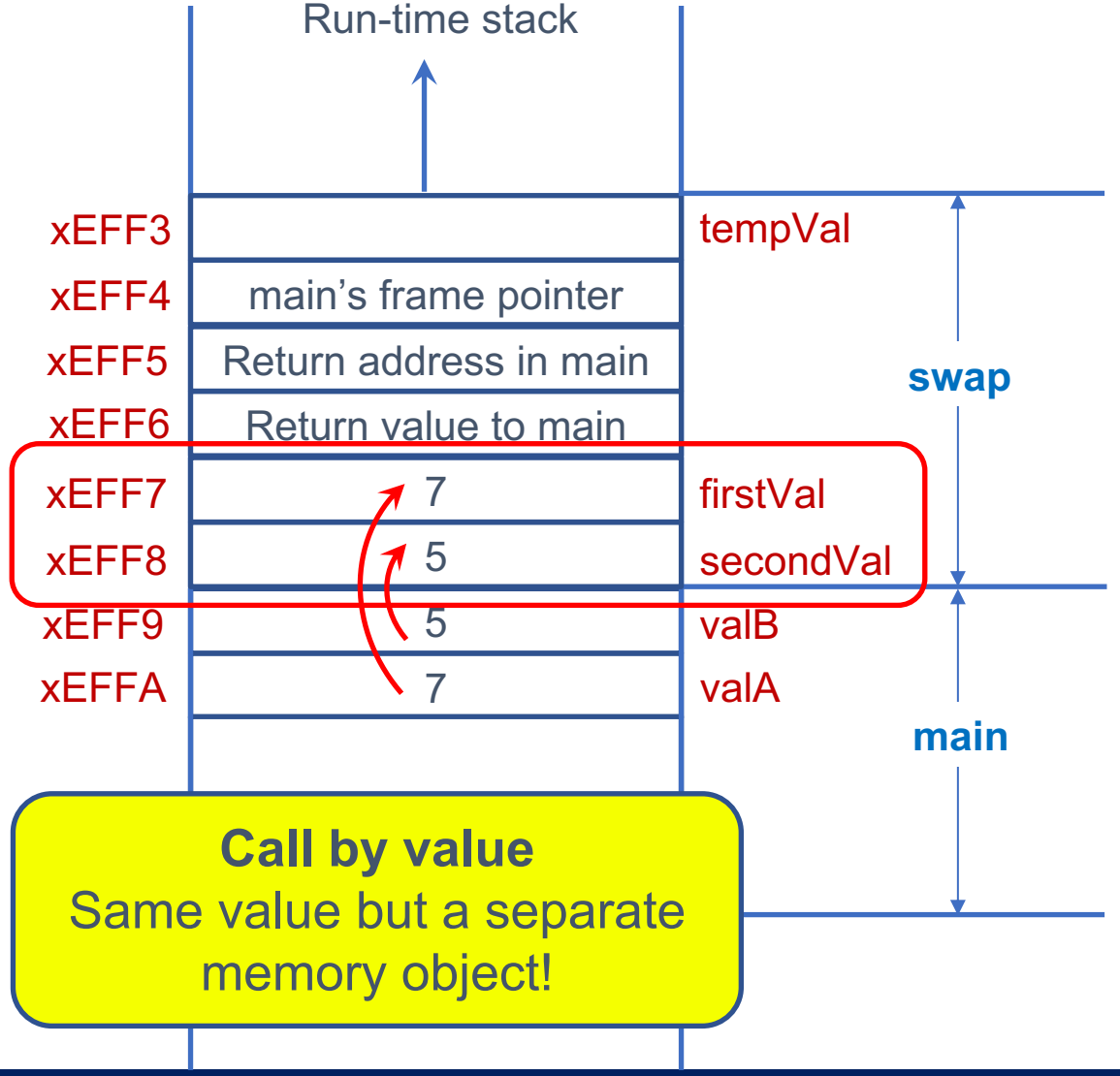
- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Practice – Call by Value (Memory)

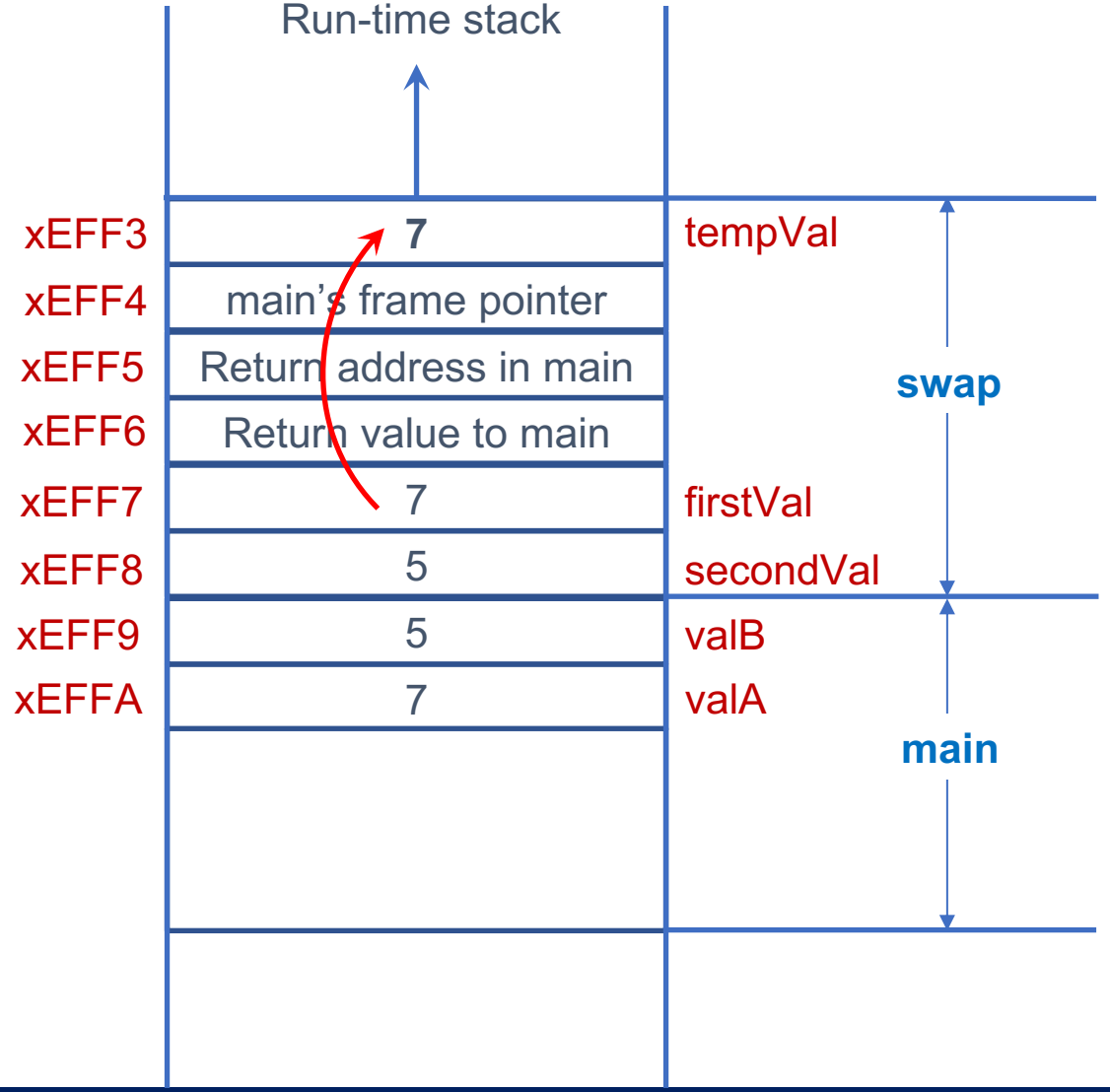
- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Practice – Call by Value (Memory)

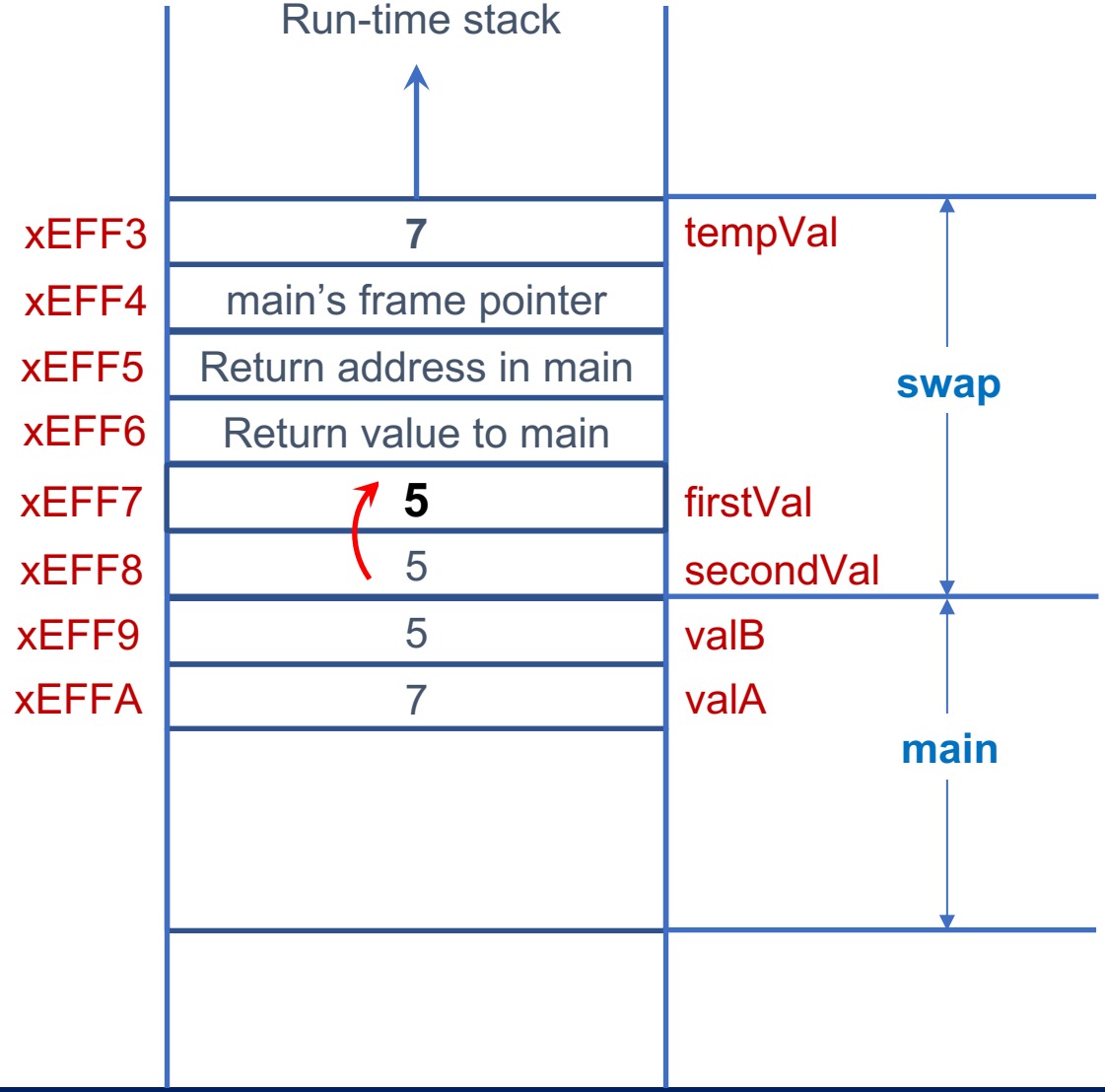
- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Practice – Call by Value (Memory)

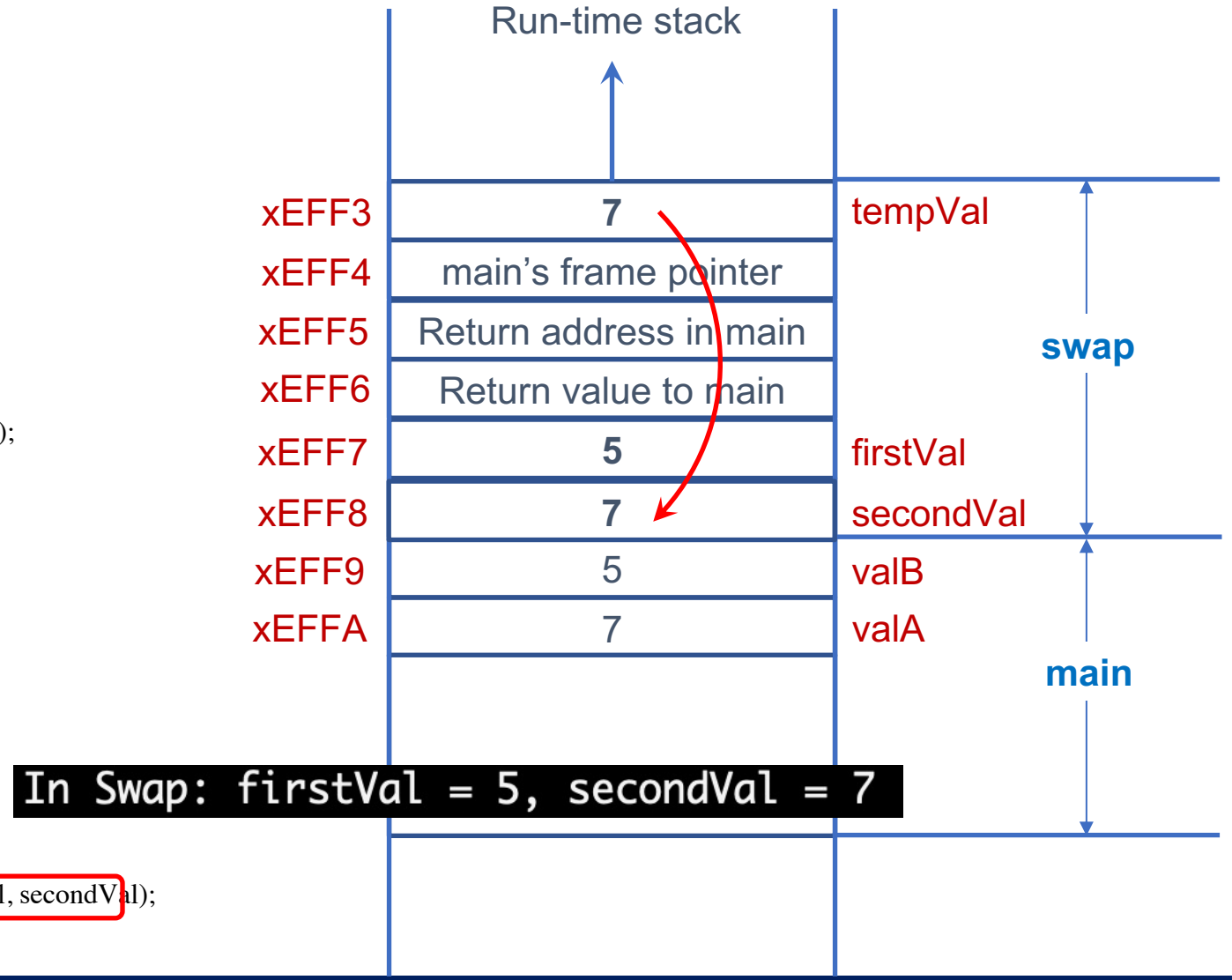
- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Practice – Call by Value (Memory)

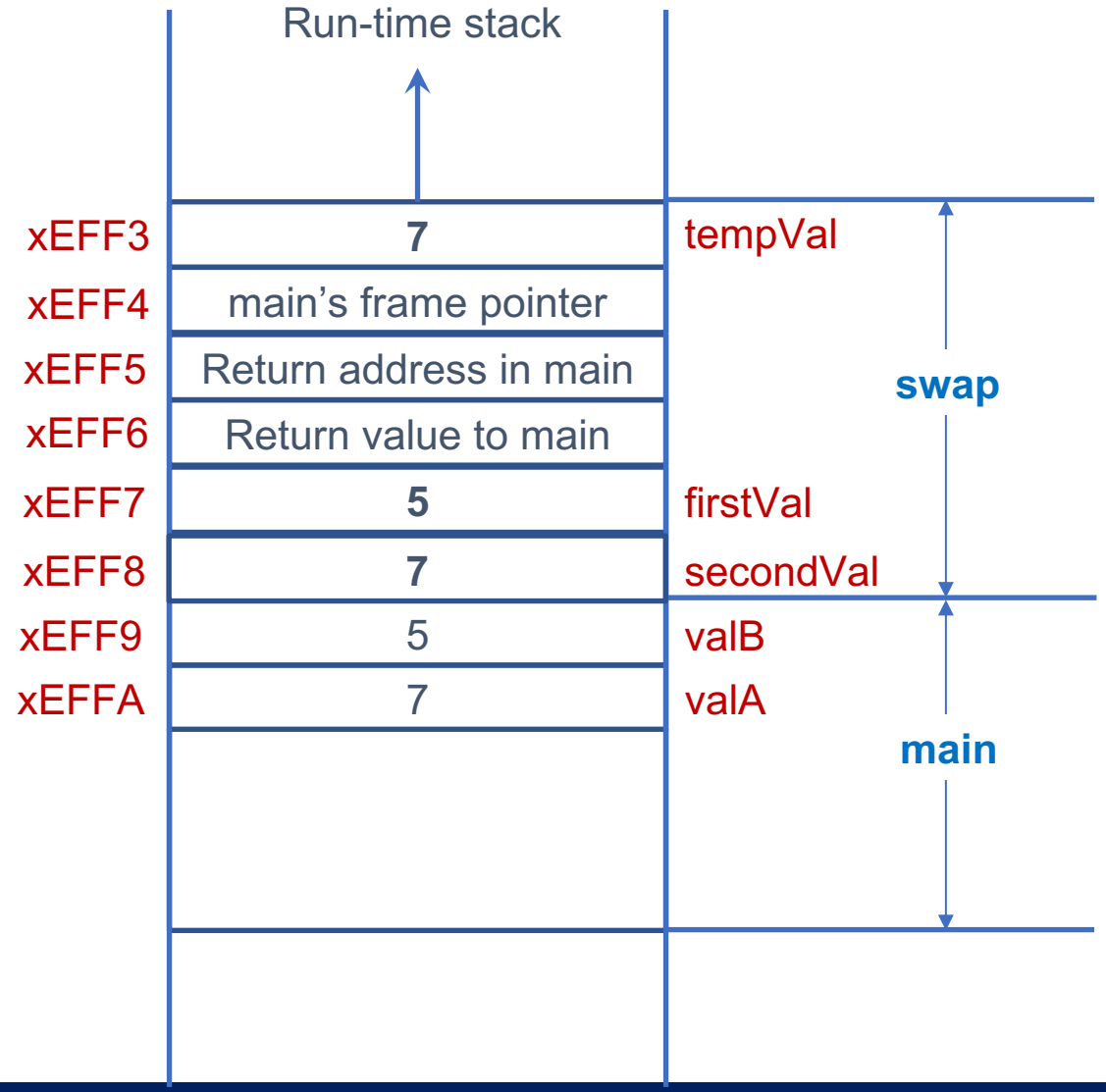
- Swapping function in C

```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Practice – Call by Value (Memory)

- Swapping function in C

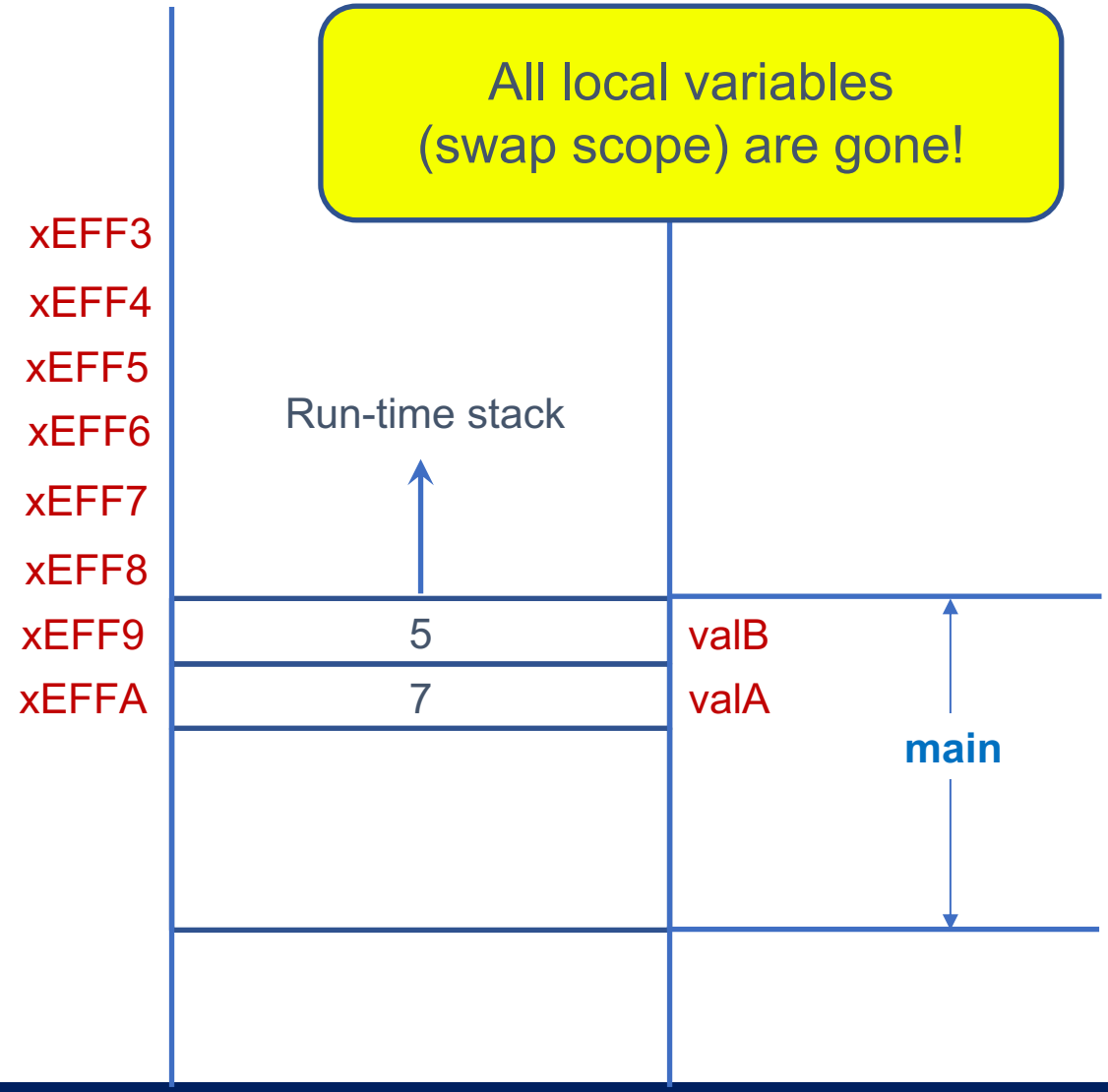
```
#include <stdio.h>

void swap(int firstVal, int secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   swap(valA, valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}
```

```
After Swap: valA = 7, valB = 5
```

```
void swap(int firstVal, int secondVal) {
   int tempVal;
   tempVal = firstVal;
   firstVal = secondVal;
   secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", firstVal, secondVal);
}
```



Pointers

Lecture 29-2

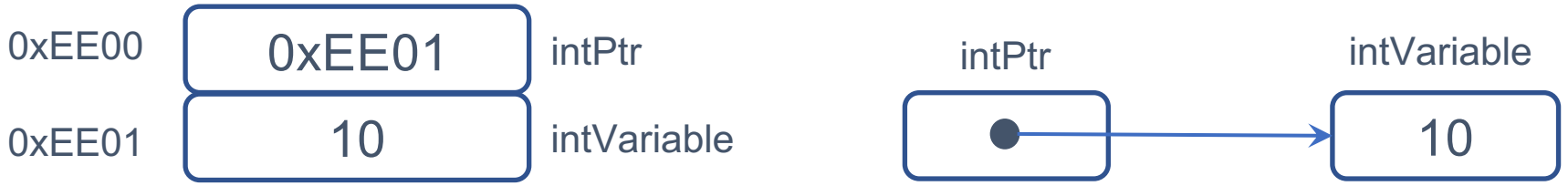
Hyung-Sin Kim

SNU Graduate School of Data Science

How can we make the swap function impact the arguments?

Pointer – Declaration

- A pointer variable contains an **address** of a memory object (e.g., variable)
  - <type> *<name>
    - int *ptr; ➡ ptr is a variable that contains an address of an integer variable
    - char *ptr; ➡ ptr is a variable that contains an address of a character variable
- Address operator & and indirection operator *
  - int intVariable = 10; // Assume that intVariable's address is 0xEE01
  - int *intPtr;
  - intPtr = &intVariable;
    - Now intPtr contains intVariable's address
    - *intPtr is the value in the memory object that intPtr points to (i.e., intVariable's value, 10)
    - "*intPtr = *intPtr + 2" is the same as "intVariable = intVariable + 2"



Let’s print some values!

Pointer – Swap (Call by Reference)

- Swapping function in C (pointer version)

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```

Let's type, compile, and execute!

What do you see on your screen?
Does swapping happen as intended?

Pointer – Swap (Call by Reference)

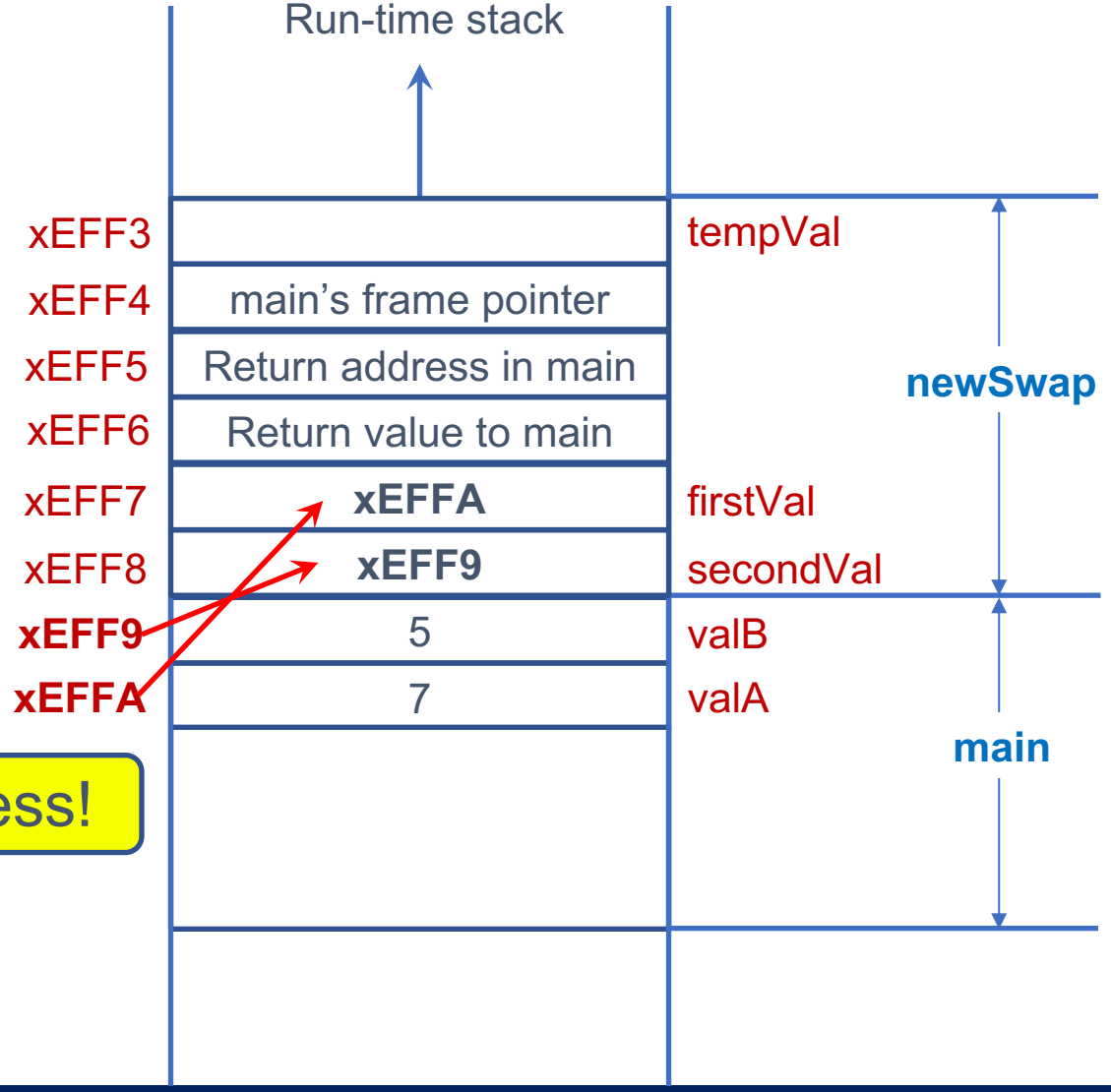
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

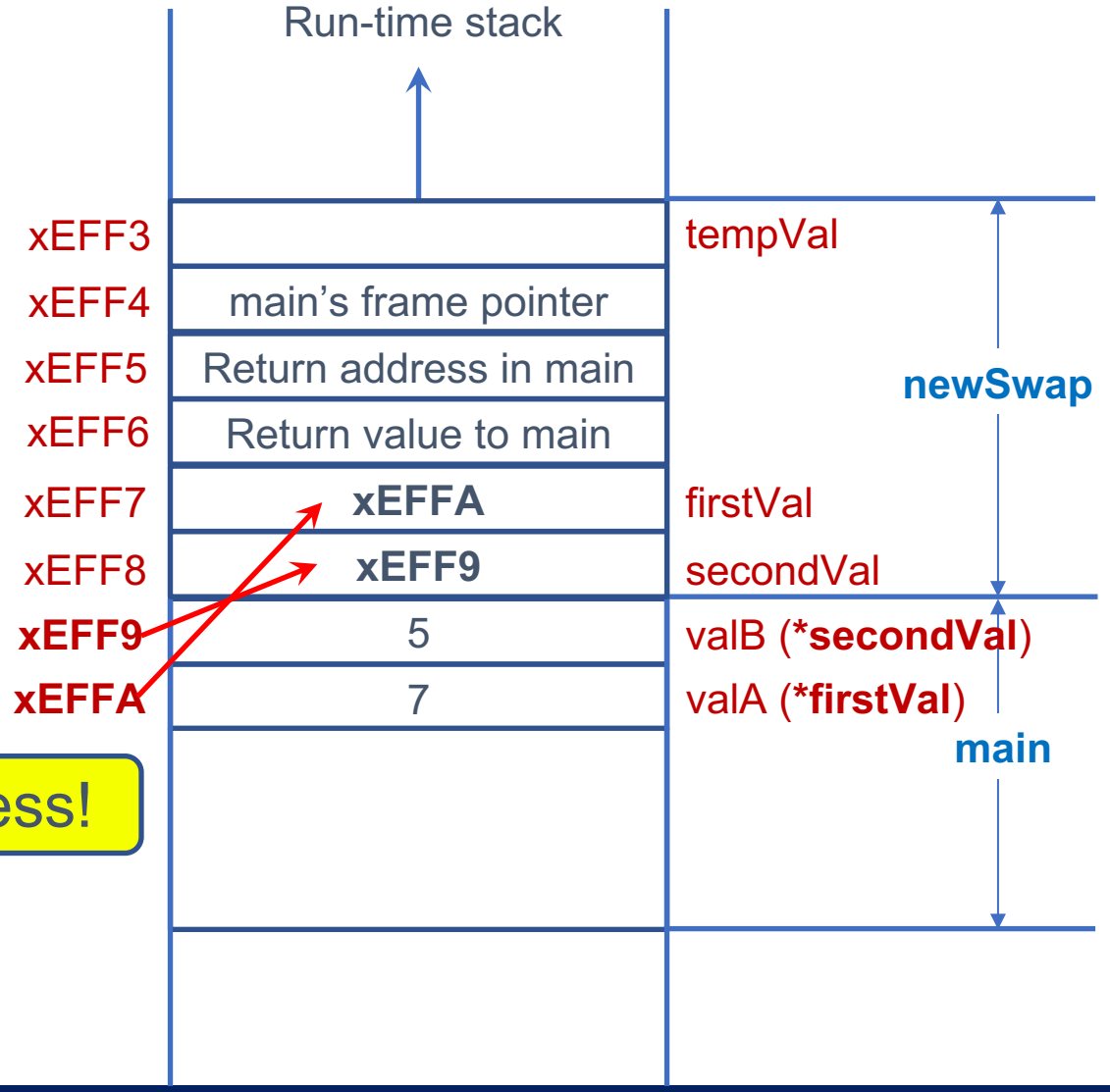
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

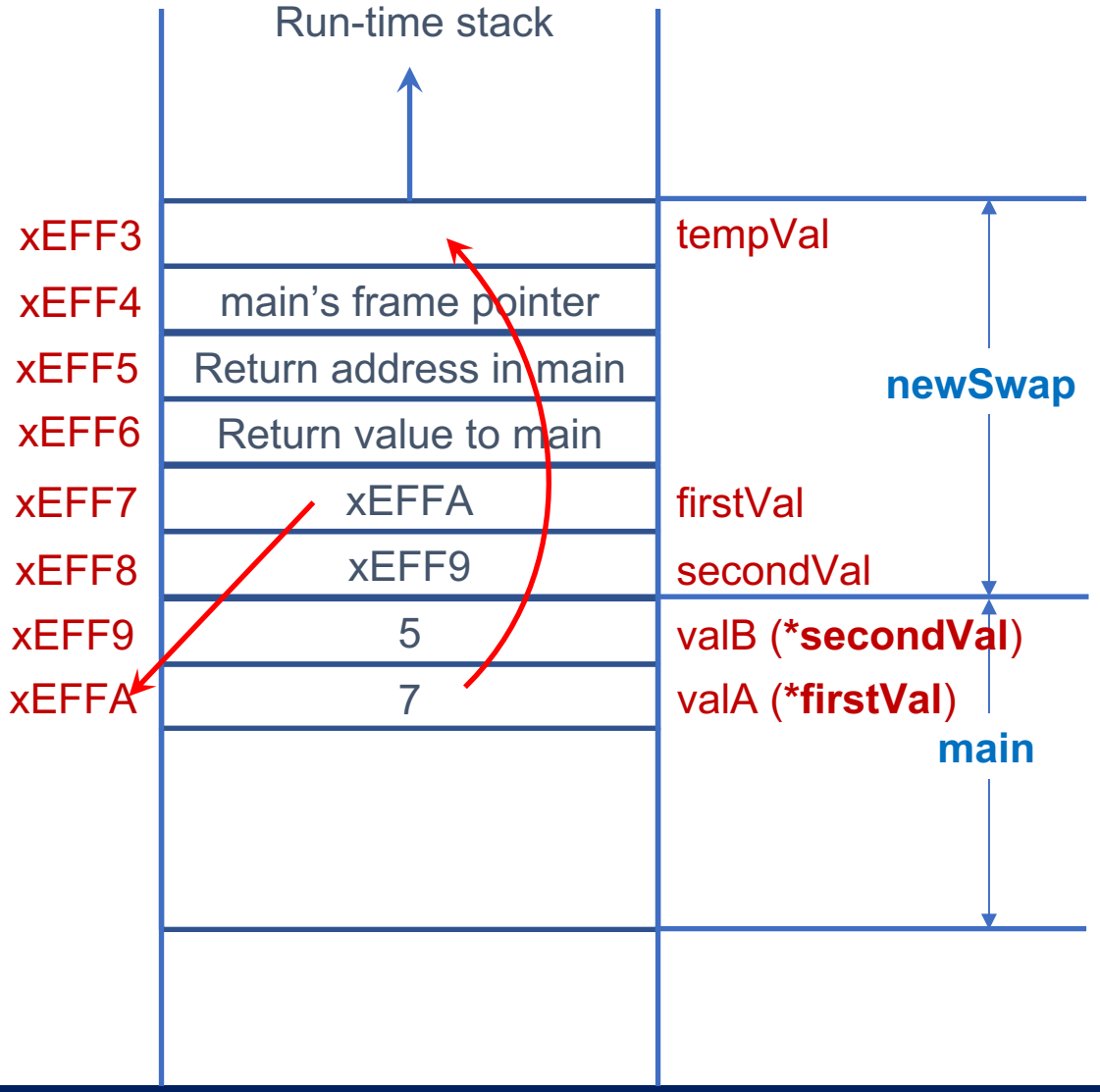
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 7;
   int valB = 5;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

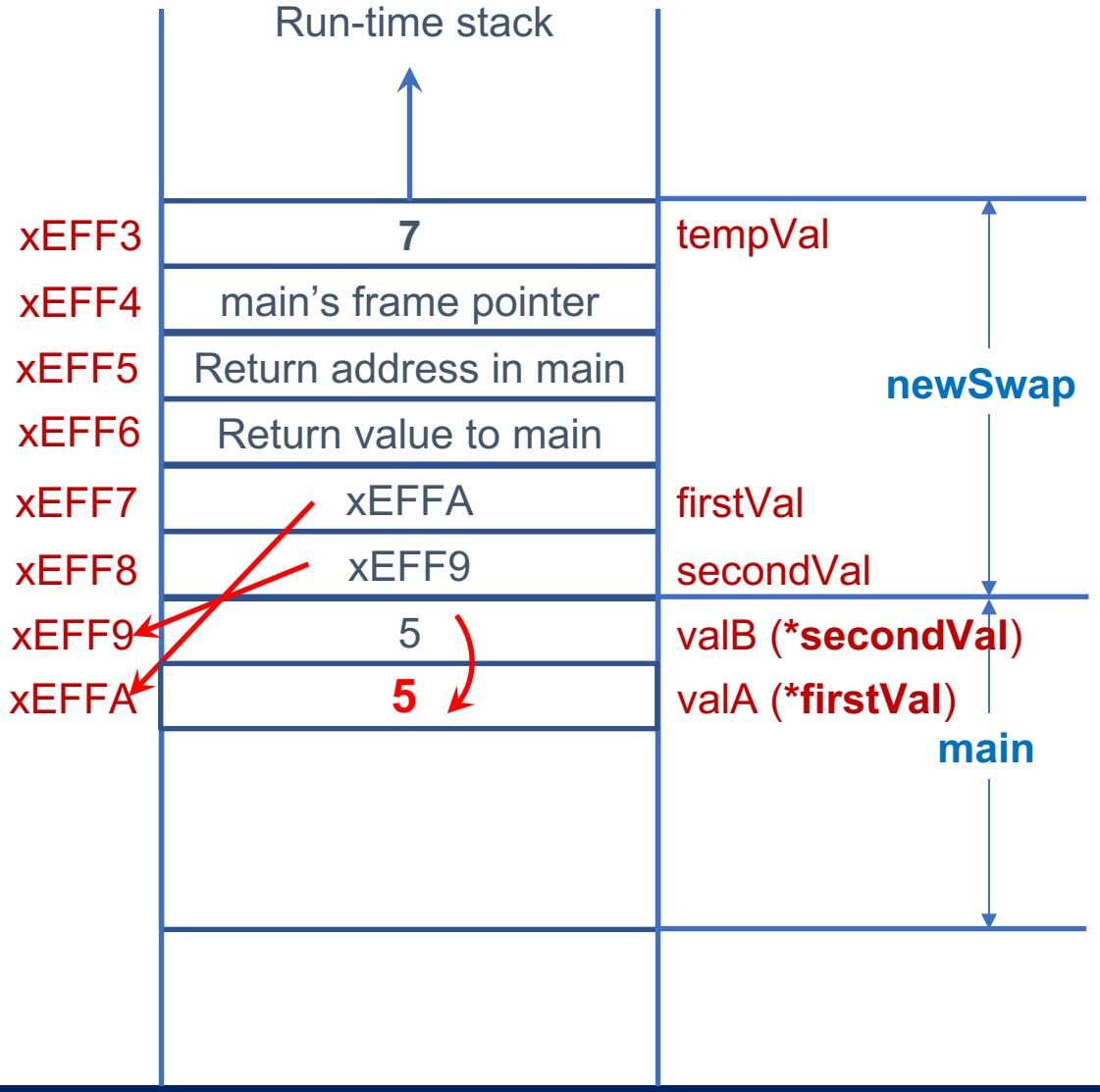
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 3;
   int valB = 4;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

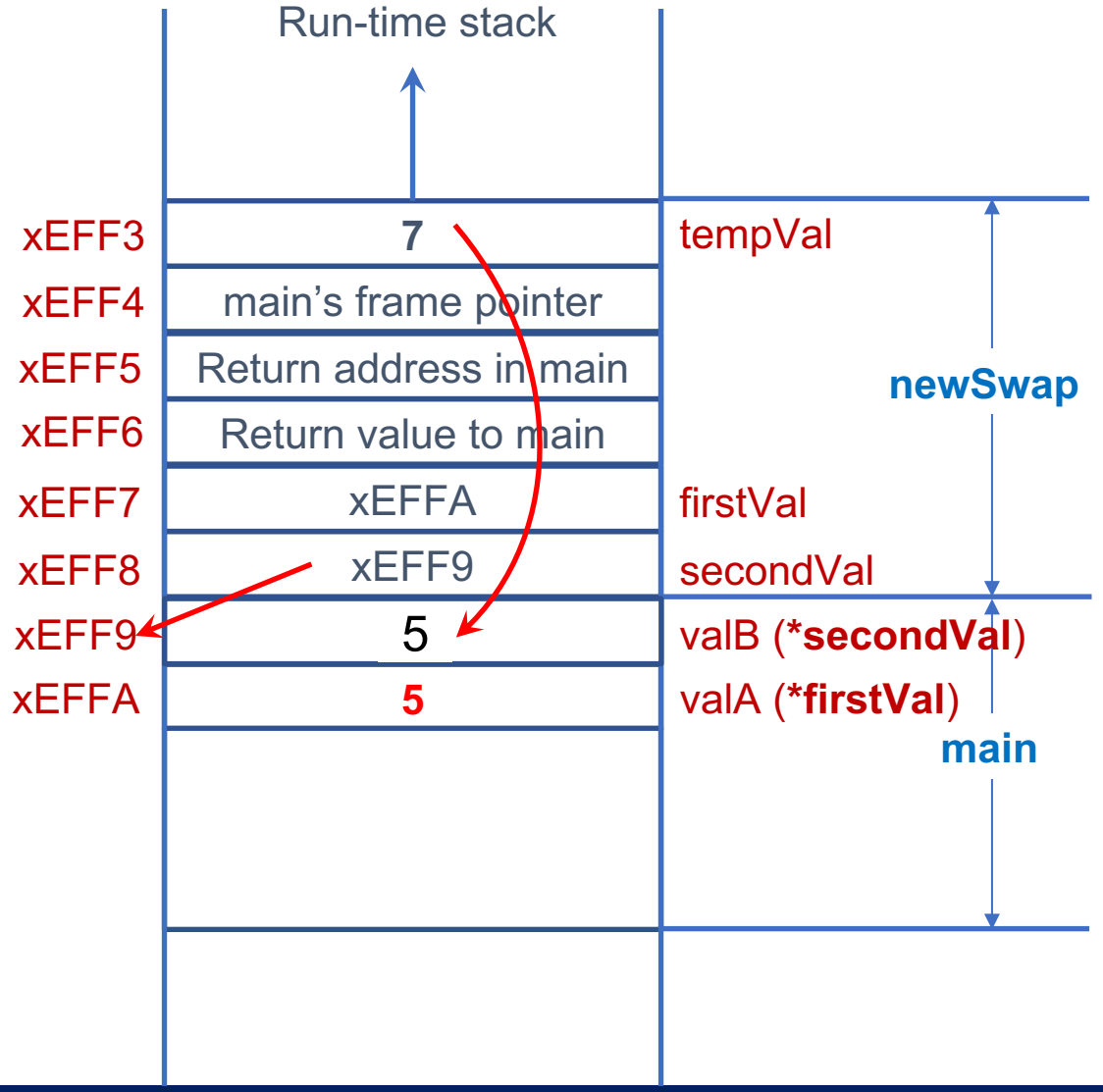
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 3;
   int valB = 4;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

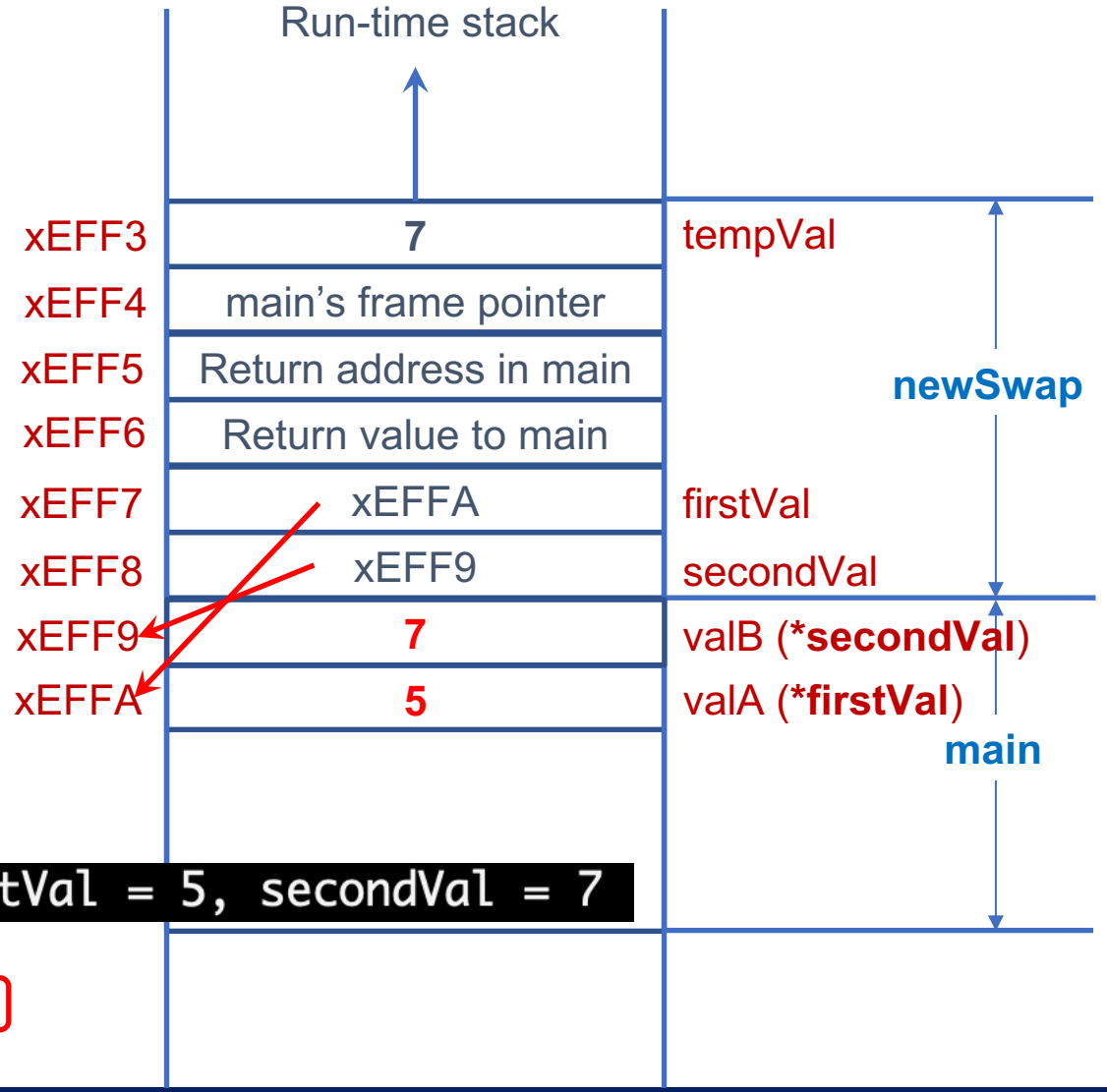
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 3;
   int valB = 4;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```




```
In Swap: firstVal = 5, secondVal = 7
```

Pointer – Swap (Call by Reference)

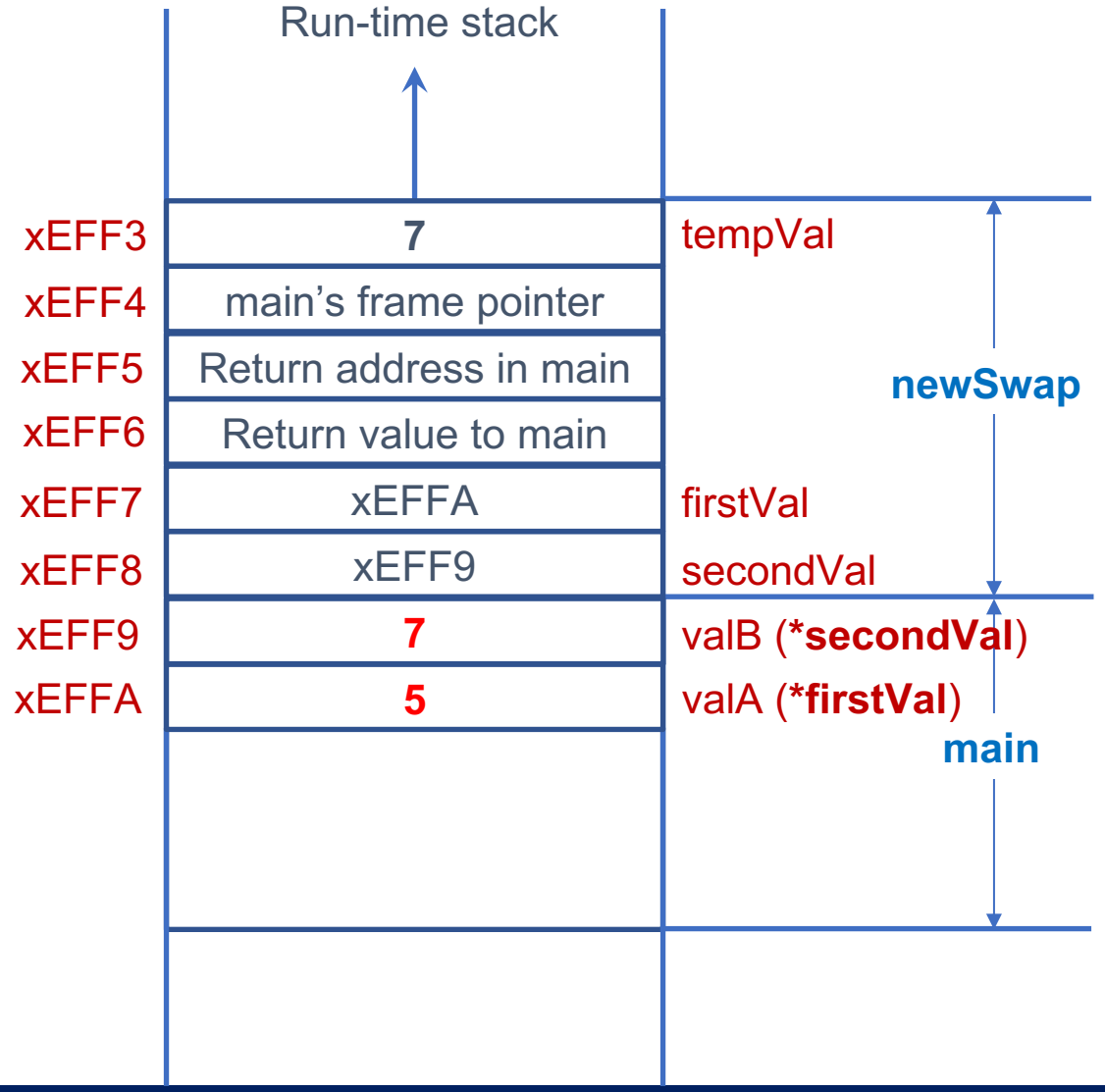
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 3;
   int valB = 4;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

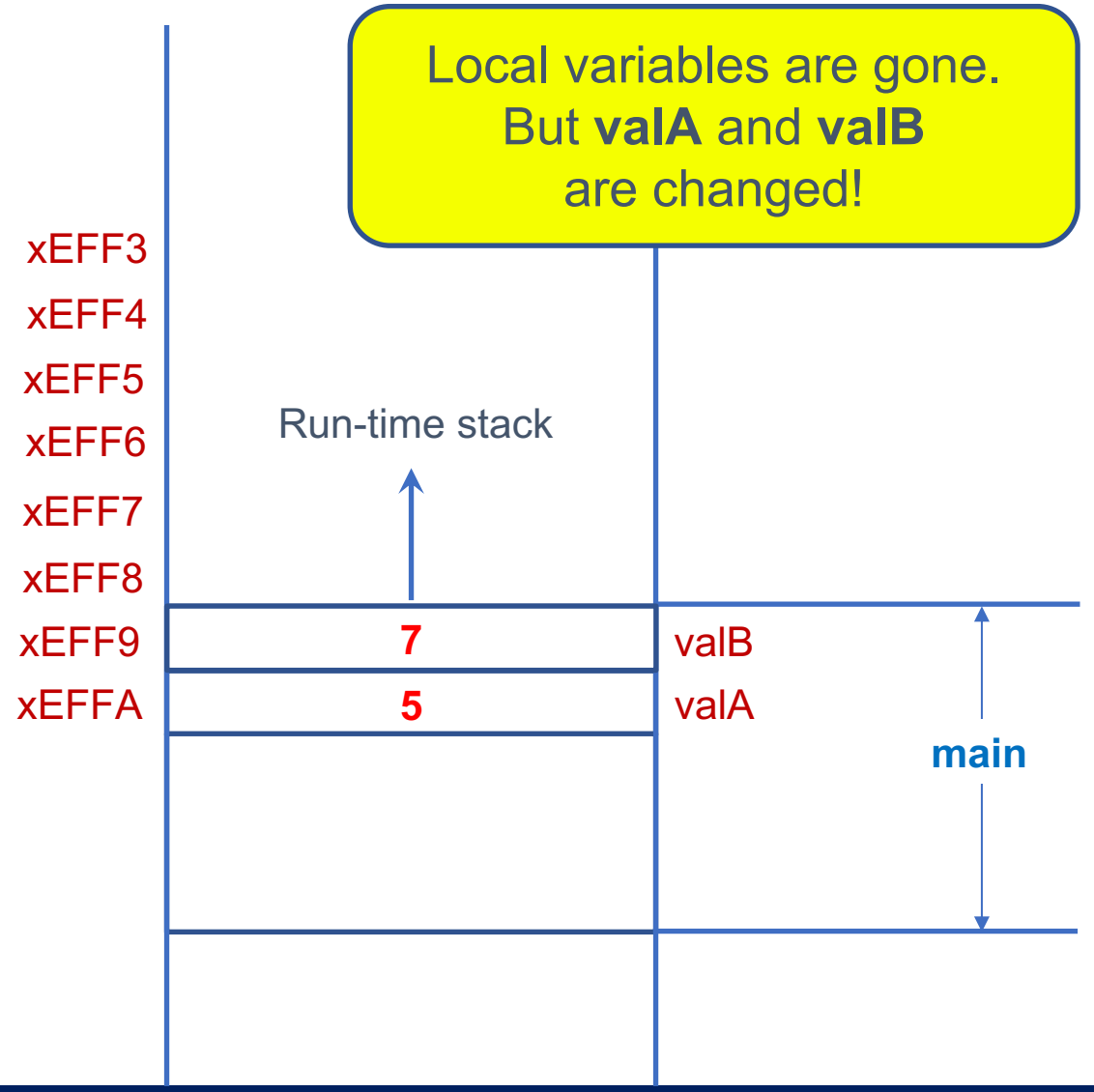
- Swapping function in C

```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 3;
   int valB = 4;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}

void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – Swap (Call by Reference)

- Swapping function in C

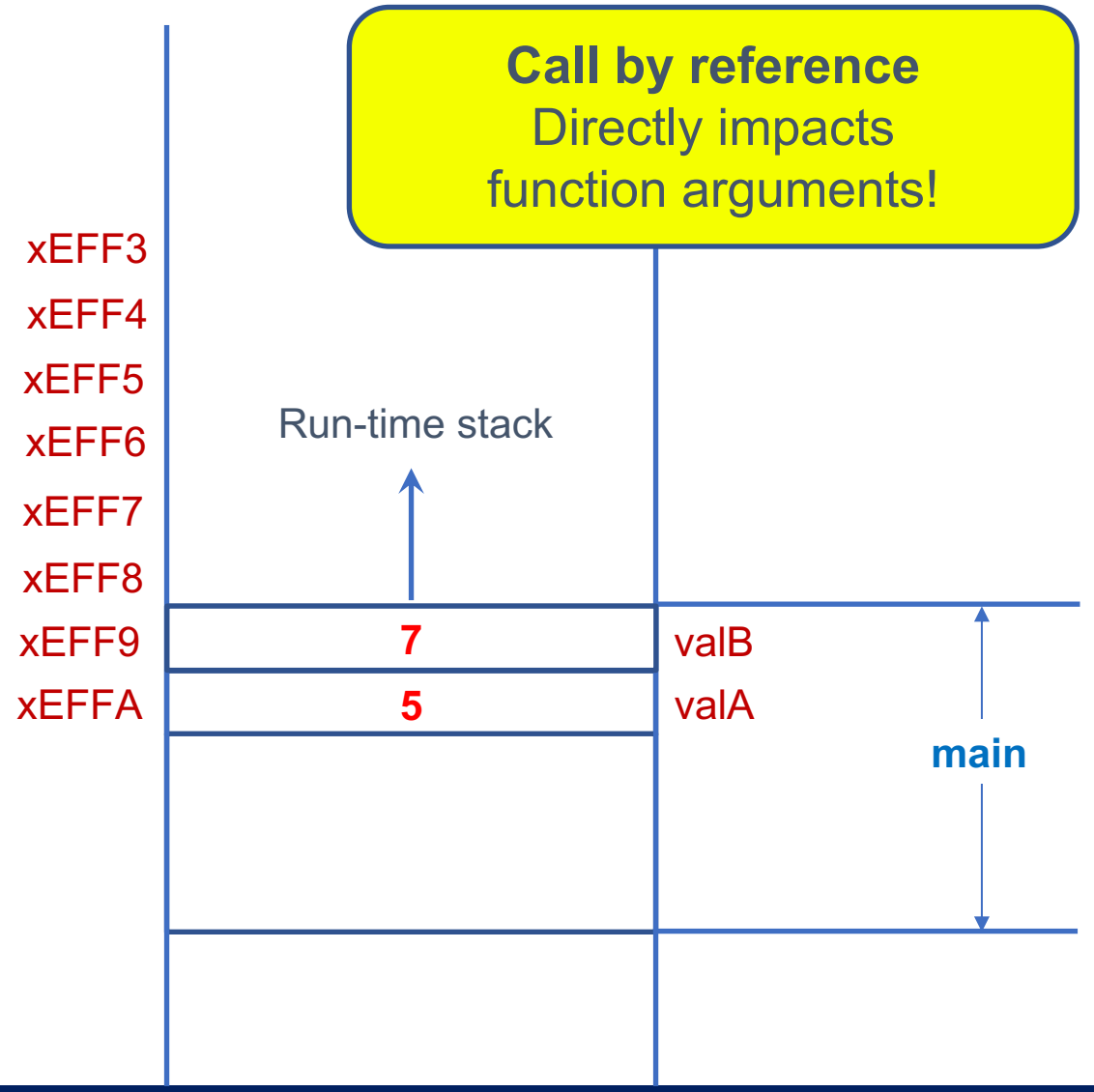
```
#include <stdio.h>

void newSwap(int *firstVal, int *secondVal);

int main(void) {
   int valA = 3;
   int valB = 4;
   printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
   newSwap(&valA, &valB);
   printf("After Swap: valA = %d, valB = %d\n", valA, valB);
   return 0;
}
```

```
After Swap: valA = 5, valB = 7
```

```
void newSwap(int *firstVal, int *secondVal) {
   int tempVal;
   tempVal = *firstVal;
   *firstVal = *secondVal;
   *secondVal = tempVal;
   printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
}
```



Pointer – etc.

- Null pointers – A special case pointer that points to **nothing**
  - int *ptr;
  - ptr = NULL; // NULL is a specially defined preprocessor macro that contains a value 0
  - It is useful to **initialize** a pointer to NULL when it does not point anything yet

- Demystifying the syntax
  - Pointer declaration (e.g., int *ptr;)
    - Declaring a variable **ptr** that, when the **indirection operator *** is applied to it, generates a value of type **int**
    - That is, ***ptr** is integer type
  - Input library function – scanf("%d", **&**input);
    - To change the value of the function argument **"input,"** scanf must have the **address** of "input"
    - If you omit **&**, C compiler will kindly give an error message

Summary

- Pointer
  - Motivation
  - Declaration
  - Swap

Thanks!