

# Audio Programming with Chuck

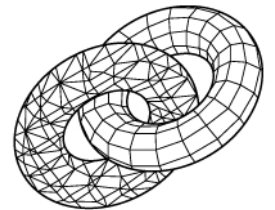
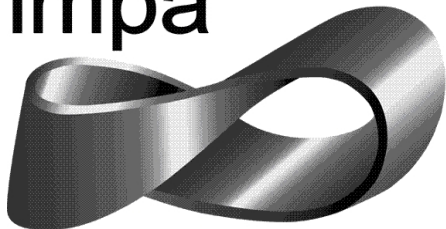
## Session 4: Functions

Vitor Guerra Rolla

Postdoctoral Fellow

[vitorgr@impa.br](mailto:vitorgr@impa.br)

impa



VisgrafLab

# Mini-course Schedule

08/01/2019	Session 1: Basics: Sound, Waves, and ChuckK initiation
10/01/2019	Session 2: MIDI, ChuckK Libraries, and Arrays
15/01/2019	Session 3: Sound File Manipulation
Today	Session 4: Functions
22/01/2019	Session 5: Unit Generators and Physical Models
24/01/2019	Session 6: Multi-Threading and Concurrency
29/01/2019	Session 7: Classes and Object-Oriented Programming

# Session 4: Functions

Introduction & Basic Example

Local vs. Global

Octave / Fifth

Gain swells

ABA form (Ternary)

Recursion - DrumRoll

# Functions

**MAIN PROGRAM**

Function I

Function II

Function III

# Example

```
fun int fname( int arg, int arg2 )  
{  
    int result;  
  
    // do something  
  
    return result;  
}
```

# Example

```
      RETURN TYPE      NAME      INPUT ARGUMENTS
      ↓                ↓                ↓
FUNCTION → fun int fname( int arg, int arg2 )
      {
        int result;

        // do something

        return result; ← OUTPUT
      }
```

# Local vs. Global variables

The scope of a declaration is the block of code where the variable is valid for use.

A global variable(declaration) is outside the main program.

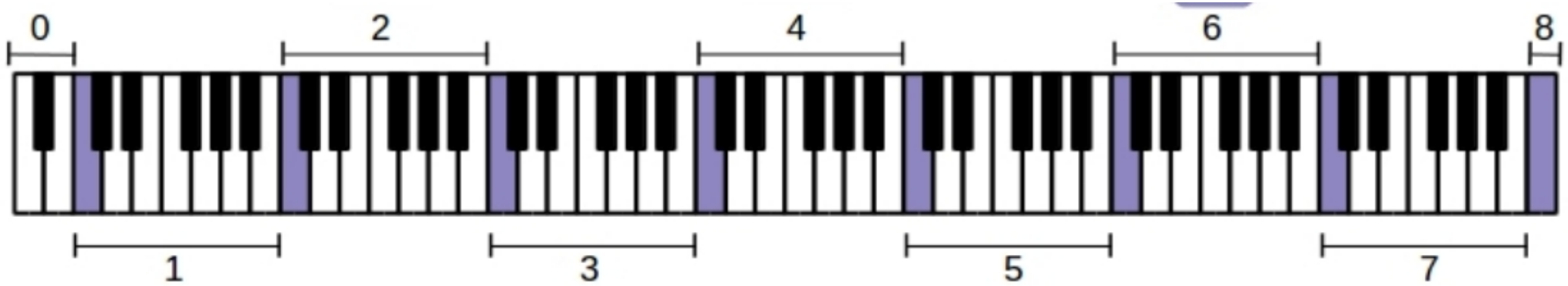
- Normally grouped with other global declarations and at the beginning of the program file.

A local variable (declaration) is inside the body of a function.

- Locally declared variables cannot be accessed outside of the function they were declared in.
- Possible to declare the same identifier name in different parts of the program.

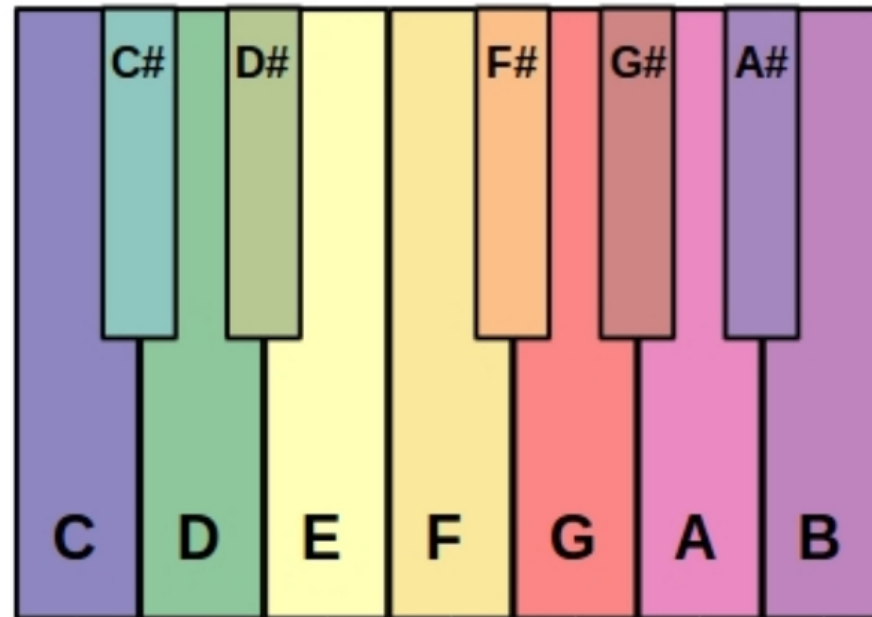
# Octave

2x freq  
+12 midi



# Fifth

1.5x freq  
+7 midi





# Gain Swells

ABA form – Ternary

# Recursion

is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem.

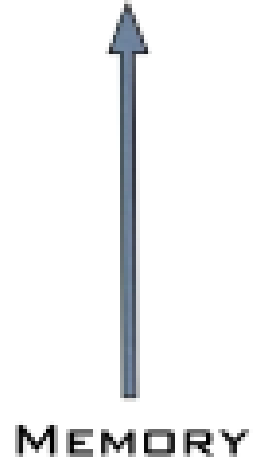
# Recursion

FUNCTION CALL

X

ANSWER EQUATION

ANSWER



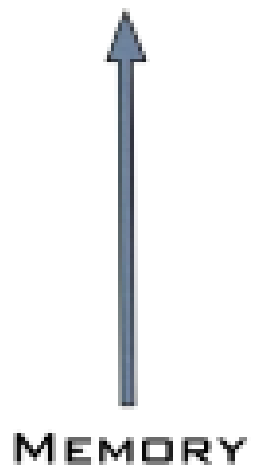
FACTORIAL(4)

4

4\*FACTORIAL(3)

?

# Recursion



<u>FUNCTION CALL</u>	<u>X</u>	<u>ANSWER EQUATION</u>	<u>ANSWER</u>
FACTORIAL(3)	3	3*FACTORIAL(2)	?
FACTORIAL(4)	4	4*FACTORIAL(3)	?

# Recursion

↑  
MEMORY

<u>FUNCTION CALL</u>	<u>X</u>	<u>ANSWER EQUATION</u>	<u>ANSWER</u>
FACTORIAL(2)	2	2*FACTORIAL(1)	?
FACTORIAL(3)	3	3*FACTORIAL(2)	?
FACTORIAL(4)	4	4*FACTORIAL(3)	?

# Recursion

↑  
MEMORY

<u>FUNCTION CALL</u>	<u>X</u>	<u>ANSWER EQUATION</u>	<u>ANSWER</u>
FACTORIAL(1)	1	1	1
FACTORIAL(2)	2	2*FACTORIAL(1)	?
FACTORIAL(3)	3	3*FACTORIAL(2)	?
FACTORIAL(4)	4	4*FACTORIAL(3)	?

# Recursion

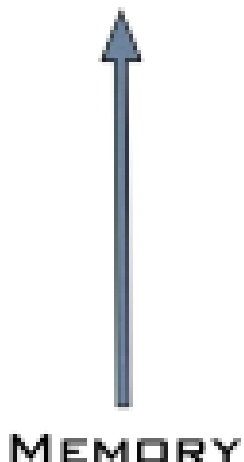
↑  
MEMORY

<u>FUNCTION CALL</u>	<u>X</u>	<u>ANSWER EQUATION</u>	<u>ANSWER</u>
FACTORIAL(2)	2	2*FACTORIAL(1)	2
FACTORIAL(3)	3	3*FACTORIAL(2)	?
FACTORIAL(4)	4	4*FACTORIAL(3)	?



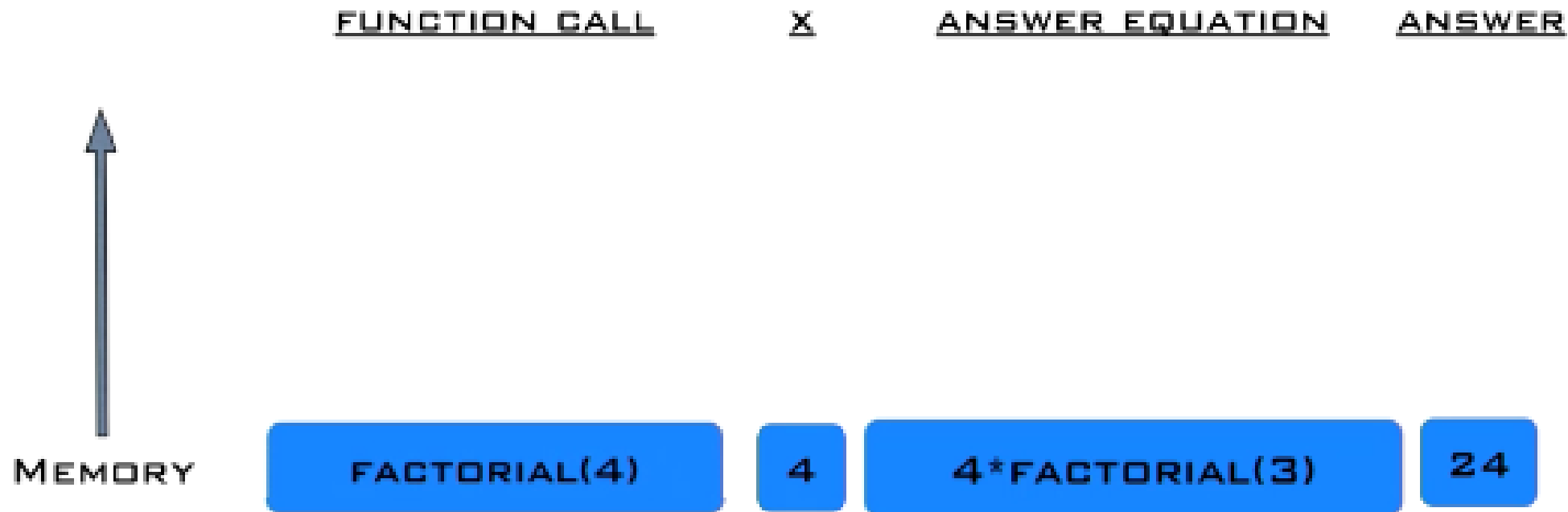
# Recursion

FUNCTION CALL      X      ANSWER EQUATION      ANSWER



FACTORIAL(3)	3	3*FACTORIAL(2)	6
FACTORIAL(4)	4	4*FACTORIAL(3)	?

# Recursion



DrumRoll