

---

# CS 241

## Control Structures

Christopher A. Gantz

SPS Undergraduate Program  
Regis University  
cgantz@regis.edu

---

# Lecture Topic #14

## CS 241: Recursion

Christopher A. Gantz  
School of Professional Studies  
Regis University  
cgantz@regis.edu

# Overview

---

- Reading
  - Nance textbook Pages 407-427
- Recursion
  - Definition
  - Constraints

# Recursion

---

- The FOR, WHILE and REPEAT looping constructs implement the definition and control of an iterative process
- Can a subprogram invoke itself, within its own executable section, as a means of implementing an alternative that also defines and controls an iterative process?
- Many classes of subprograms can be implemented efficiently by allowing the invocation of itself.

# Recursion (Cont.)

---

- The process of a subprogram invoking itself within its own executable section is referred to as recursion.
- Subprograms that invoke themselves within their executable sections are referred to as recursive subprograms.

# Why is Recursion Necessary?

---

- Certain classes of mathematical operations are best implemented by recursion
  - E.g. Factorial, Fibonacci etc.
- The mathematical definition of factorial
- For  $n > 1$ ,  $n! = n * (n-1)!$

$$3! = 3 * 2!$$

$$= 3 * 2 * 1!$$

$$= 3 * 2 * 1 * 0!$$

# Recursion Structural Constraints

---

- The recursive process must contain a well defined termination condition
  - E.g.  
IF  $N = 0$  THEN  
Factorial := 1;
- The recursive process must contain well defined steps that progress to the termination condition referred to as the recursive steps

- E.g.

Factorial :=  $N * \text{Factorial}(N-1)$ ;