

---

CS 241  
Control Structures

Christopher A. Gantz

SPS Undergraduate Program  
Regis University  
cgantz@regis.edu

---

# Lecture Topic #8

## CS 241: Repetition Statements

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

# Overview

---

- Reading
  - Nance textbook Pages 237-279
- Repetition Statements
  - FOR ... TO ... DO
  - WHILE ... DO
  - REPEAT ... UNTIL
  - Loop Assertions

# Repetition Statements

---

- Language constructs for repeating sequences of statements for some number of finite iterations
- Also commonly referred to as a Loop
- Pascal provides the following three types of repetition statements
  - FOR ... TO ... DO
  - WHILE ... DO
  - REPEAT ... UNTIL

# Repetition Statements (Cont.)

---

- In general repetition/loop structures all work in the following manner:
  - Assigning an initial value to a variable utilized to control the loop iterations
  - Evaluating the value of the loop control variable either before or after each loop iteration
  - Terminate the loop iterations when the variable reaches some pre-determined value

•

# Pre-test vs Post-test

---

- Loops utilize a condition (i.e. a boolean expression) to determine whether or not the body of the loop is executed
- If the condition is executed/evaluated ***before*** the body of the loop can be executed it is referred to as a “***pre-test***” condition
- If the condition is executed/evaluated ***after*** the body of the loop is executed it is referred to as a “***post-test***” condition

# Pre-test vs Post-test (Cont.)

---

- Pre-test conditions work in the following manner
  - If the “*pre-test*” condition is **true** then the body of the loop is executed
  - If the “*pre-test*” condition is **false** then the body of the loop is NOT executed and the statement following the repetition is executed
- FOR ... TO ... DO and WHILE ... DO loops are pre-test loops
- Pre-test loops are also referred to as “*entrance-controlled*” loops

# Pre-test vs Post-test (Cont.)

---

- Whereas post-test conditions work in the following manner
  - If the “*post-test*” condition is **false** then the body of the loop is executed
  - If the “*post-test*” condition is **true** then the body of the loop is NOT executed and the statement following the repetition is executed
- REPEAT ... UNTIL loops are “*post-test*” loops
- Post-test loops are also referred to as “*exit-controlled*” loops

# Fixed Iteration vs Variable Condition loops

---

- Fixed iteration loops are those loops that utilized when it is none apriori the number of iterations that are required
  - The FOR ... TO ... DO loop is a fixed iteration loop
- Variable condition loops are those that utilize a condition that will/should eventually be satisfied (i.e. become true) so that the loop will terminate
- The variables that comprise the condition/boolean expression are modified during each execution of the body of the loop
  - The WHILE ... DO and REPEAT ... UNTIL loop are

---

variable condition loops

# FOR Loop

---

- FOR ... TO ... DO General Syntactic Form
  - FOR <index> := <initial value> TO <final value> DO
  - <statement>; { Body of the Loop }
- The FOR loop is a single executable statement
- <index> is an identifier/variable that is assigned values for each repetition of the loop
- <index> is the loop control variable

# FOR Loop (Cont.)

---

- FOR ... TO ... DO internal operation (i.e. how it works) is as follows:
  - 1.  $\langle \text{index} \rangle = \langle \text{initial value} \rangle$
  - 2. if  $\langle \text{index} \rangle \leq \langle \text{final value} \rangle$  then
    - Execute body of loop
    - $\langle \text{index} \rangle = \langle \text{index} \rangle + 1$
    - If  $\langle \text{index} \rangle > \langle \text{final value} \rangle$  then
      - Goto statement following loop
-

# FOR Loop (Cont.)

---

- FOR ... TO ... DO Example:
  - Total := 0;
  - FOR Count := 1 TO 100 DO
  - Total := Total + Count;
- Above code fragment is an example of an accumulator
- Where an accumulator is simply a variable that sums values
- See page 240 of text for another accumulator example

# FOR Loop (Cont.)

---

- Note: since the char data type is an ordinal data type it can be utilized withing a FOR ... TO ... DO loop (See example 6.5 on pg 243 of text)
- Common FOR loop variation
- FOR ... DOWNTO ... DO
  - Properties
    - Still a pre-test loop
    - Still a fixed iteration loop
- Difference is that instead of increasing <index> on each iteration <index> is decreased

# FOR Loop (Cont.)

---

- FOR ... DOWNTO ... DO General Syntactical Form
  - FOR <index> := <initial value> DOWNTO <final value> DO
  - <statement>;   { Body of the loop }
- See Example 6.10 on page 246 of text

# Loop Writing Style

---

- Body of loop should be indented to visually recognize it
- See sample code fragment on page 248 of text
-

# WHILE ... DO loop

---

- Properties:
  - Pre-test loop
  - Variable condition loop
- General Syntactical Form
- WHILE <Boolean Expression 3> DO
- <Statement>; {Body of the loop}

# WHILE ... DO Flow Diagram

---

# WHILE ... DO loop (Cont.)

---

- Characteristics/Properties of WHILE ... DO loop
  - Boolean Expression can be any expression that generates a boolean value
  - Boolean Expression must be true prior to entering and executing the body of the loop
  - Body of the loop can be
    - Single statement
    - Compound statement
  - Body of the loop must effect the boolean expression on every iteration

# WHILE ... DO loop (Cont.)

---

- - Body of the loop must effect the boolean expression on every iteration
- If above property is violated the loop will/can become an infinite loop
- ***“An infinite loop is a loop that never terminates”***
-

# WHILE ... DO loop (Cont.)

---

- WHILE loops often utilize a sentinel value
- A sentinel value is a special value that indicates the end of a set of data or process
- See Example 6.14 on pg 255

# Looping idioms

---

- A priming read – a read statement that occurs before a loop is entered
- A counter variable – a variable that is utilized to count the number of times a body of the loop is executed
- WHILE NOT eoln or eof
  - Utilized when reading input from a data file
- See Examples on pg 258 thru 260 of text

# Repeat ... Until Loop

---

- Properties
  - Post-test/exit controlled loop
  - Variable condition loop
- Template Example

REPEAT

{process data until sentinel value }

UNTIL <variable> = <sentinal value>;