TSO REXX ® ™

TSO REXX: TSO REXX Programming Student Guide

RX201/E

Computer Associates™

EDJ24T20SGU

– PROPRIETARY AND CONFIDENTIAL INFORMATION –

These education materials and the related computer software program (hereinafter referred to as the "Education Materials") are for the end user's informational purposes only and are subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

These Education Materials may not be copied, transferred, reproduced, disclosed or distributed, in whole or in part, without the prior written consent of CA. These Education Materials are proprietary information and a trade secret of CA. Title to these Education Materials remains with CA, and these Education Materials are protected by the copyright laws of the United States and international treaties. All authorized reproductions must be marked with this legend.

RESTRICTED RIGHTS LEGEND

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

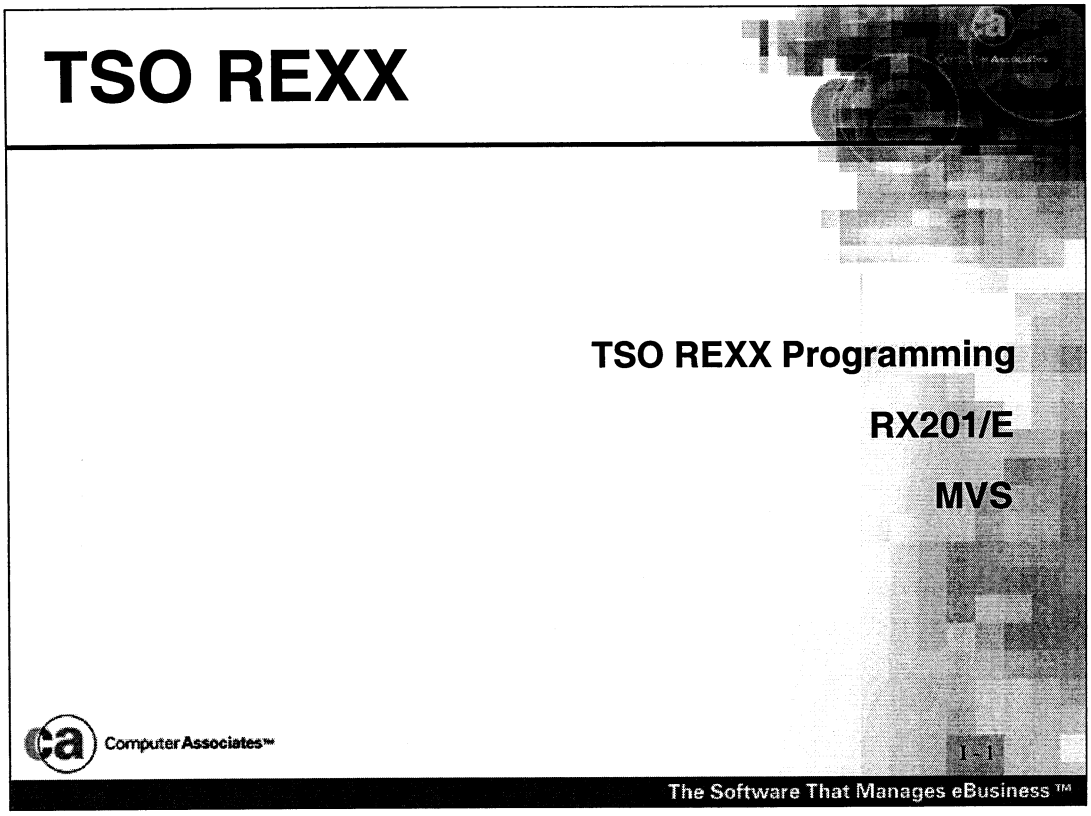
THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

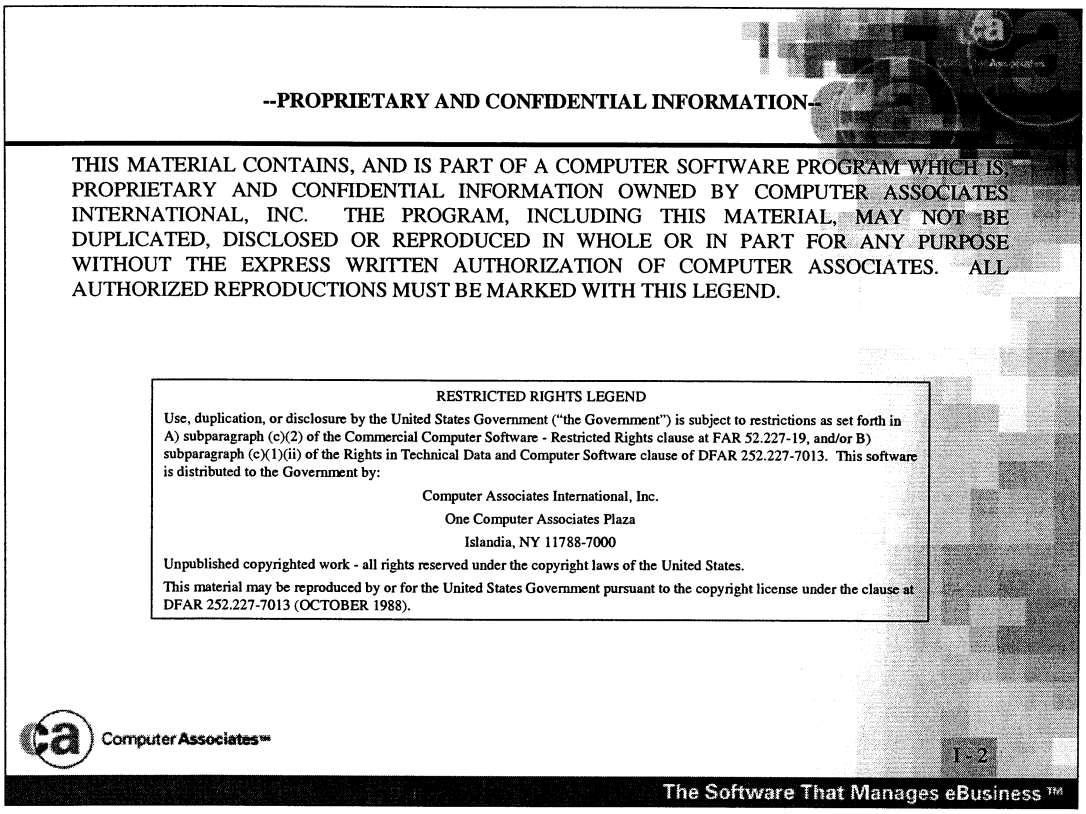
The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

Call Computer Associates technical services for any information not covered in this manual or the related publications. In North America, see your Computer Associates *Product Support Directory* for the appropriate telephone number to call for direct support, or you may call 1-800-645-3042 or 631-342-4683 and your call will be returned as soon as possible.

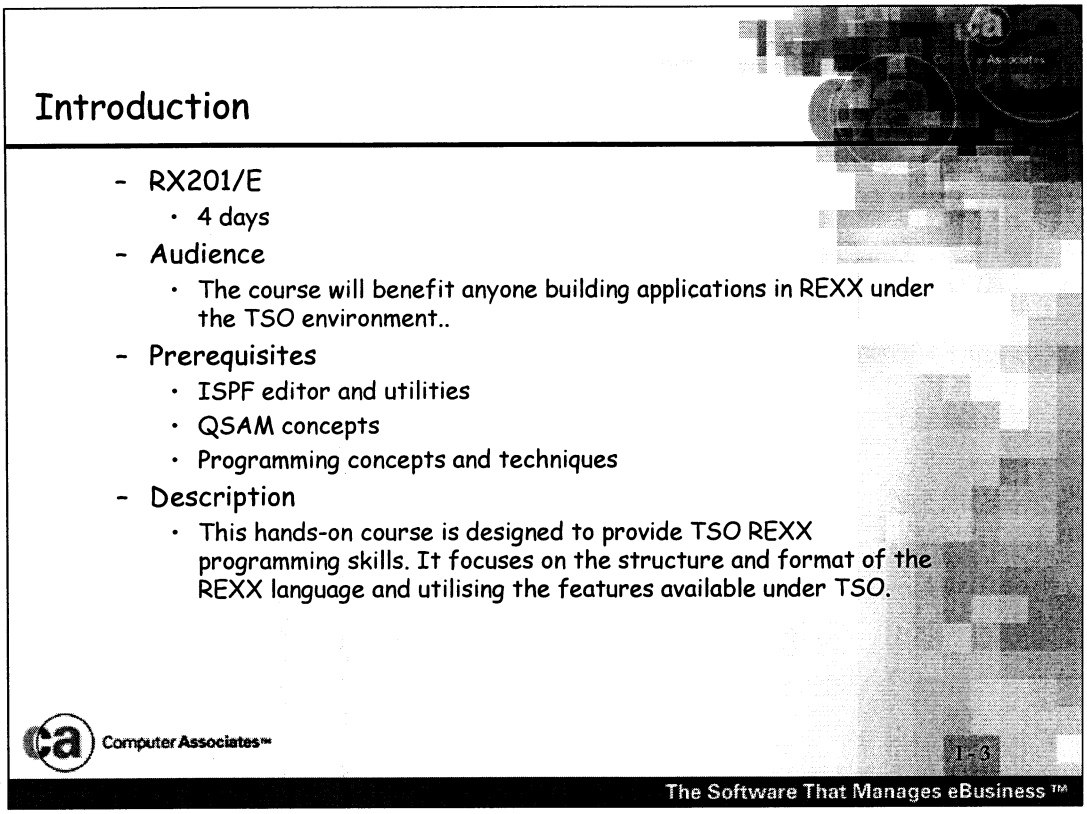
Outside North America, contact your local Computer Associates technical support center for assistance.


TSO REXX
TSO REXX Programming
RX201/E
MVS
ca Computer Associates™
I - 1
The Software That Manages eBusiness ™



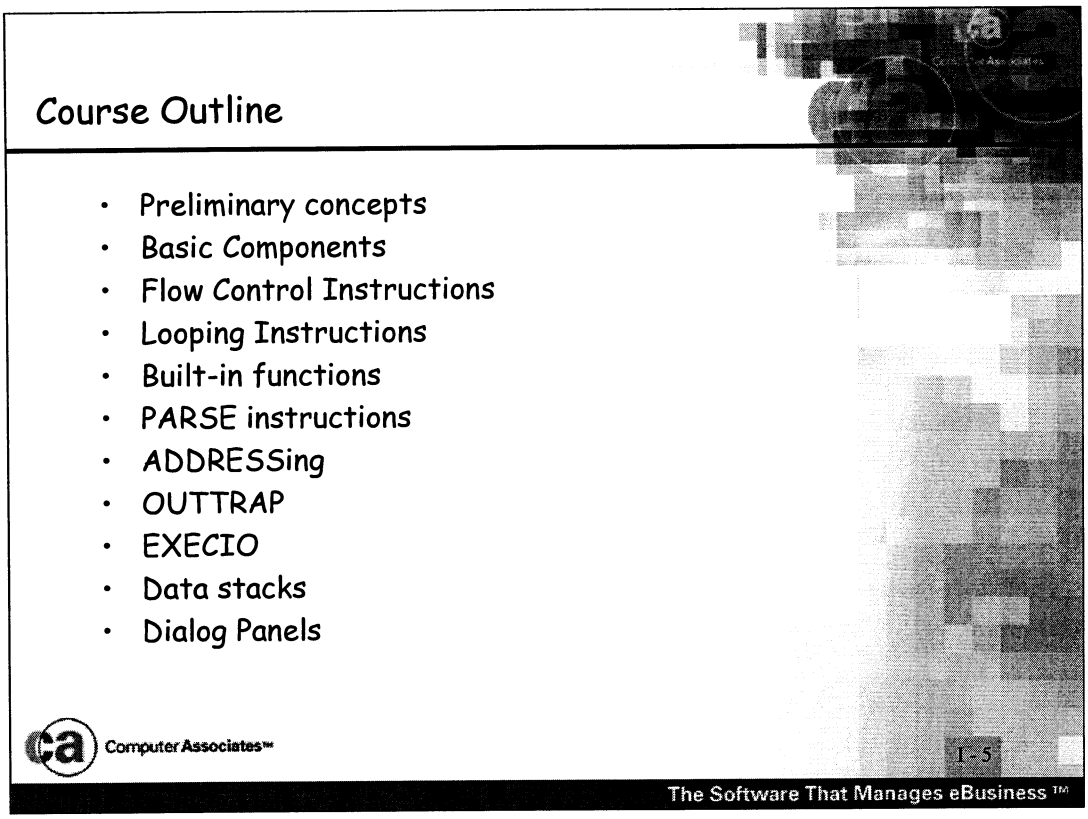
Call Computer Associates technical services for any information not covered in this manual or the related publications. In North America, see your Computer Associates Product Support Directory for the appropriate telephone number to call for direct support, or you may call 1-800-645-3042 or 516-342-4683 and your call will be returned as soon as possible.

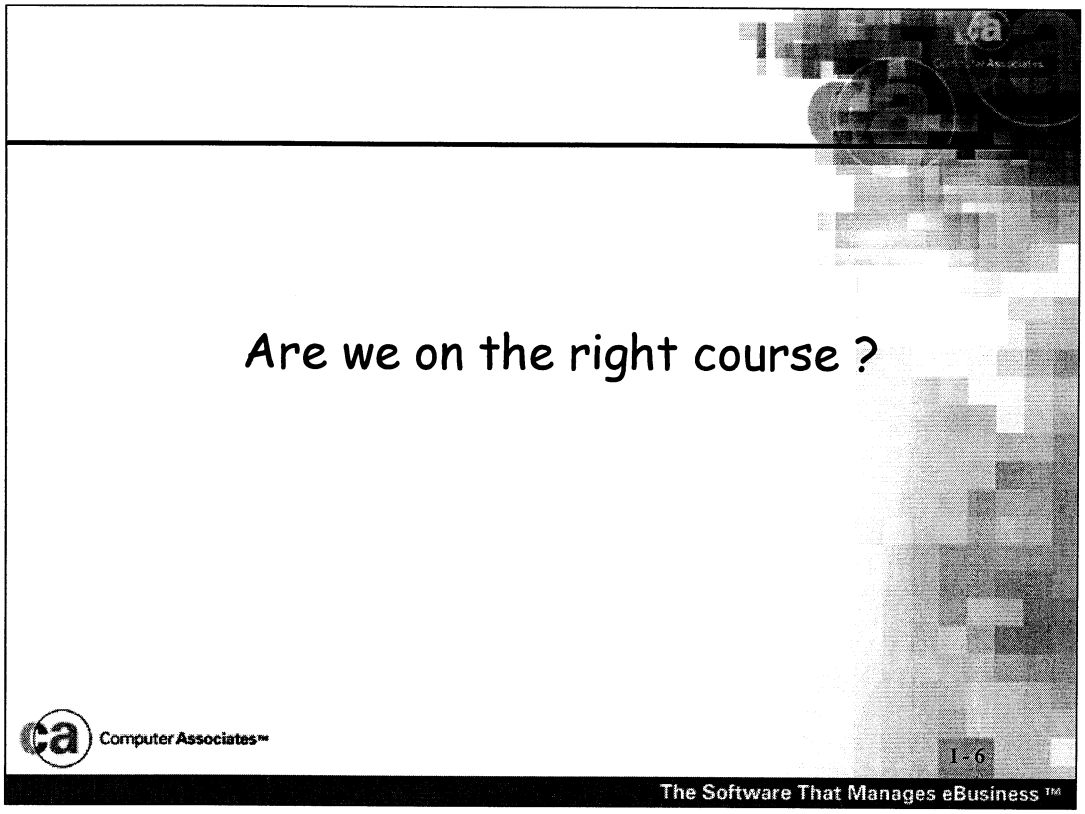
Outside North America, contact your local Computer Associates technical support center for assistance.

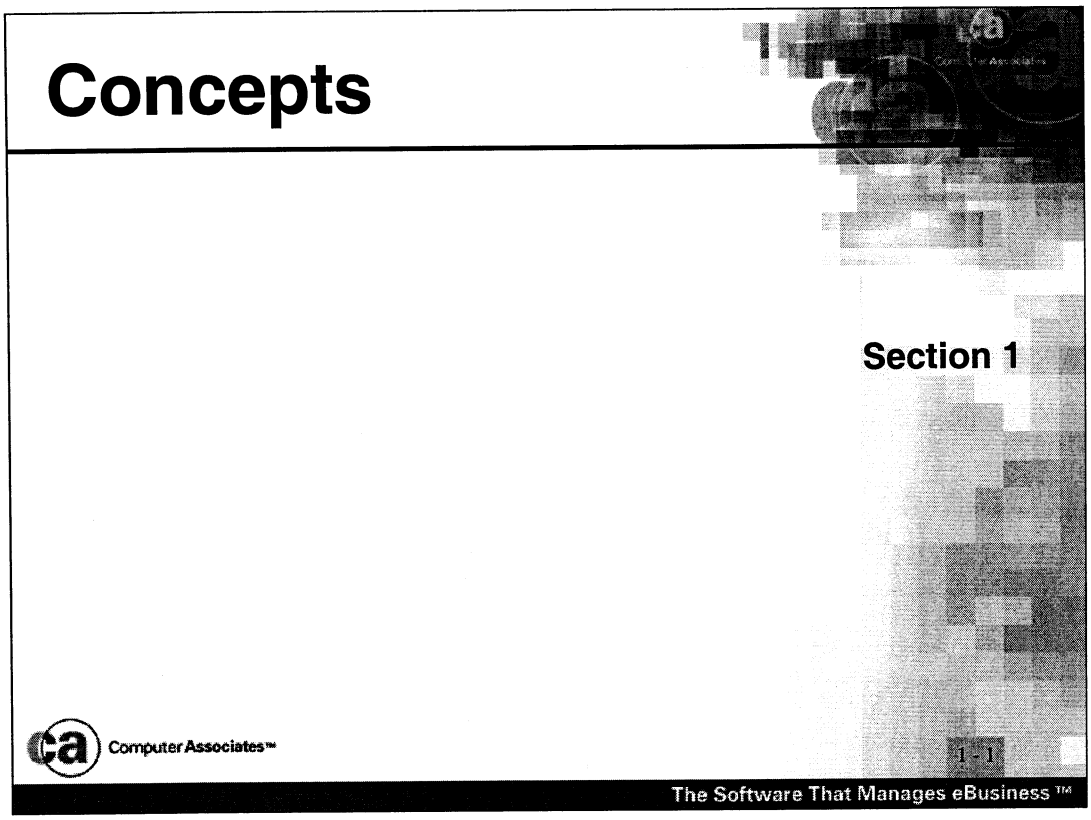

Introduction
- RX201/E
• 4 days
- Audience
• The course will benefit anyone building applications in REXX under the TSO environment..
- Prerequisites
• ISPF editor and utilities
• QSAM concepts
• Programming concepts and techniques
- Description
• This hands-on course is designed to provide TSO REXX programming skills. It focuses on the structure and format of the REXX language and utilising the features available under TSO.
Computer Associates™
I - 3
The Software That Manages eBusiness ™

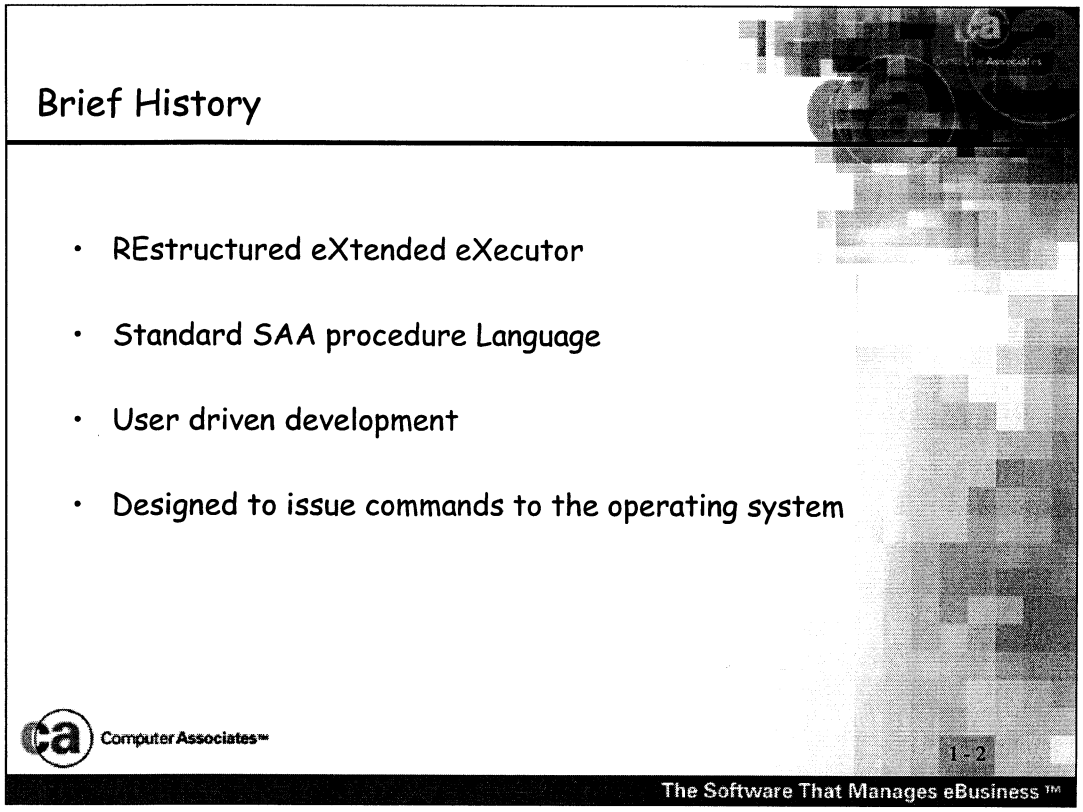
Course Objectives

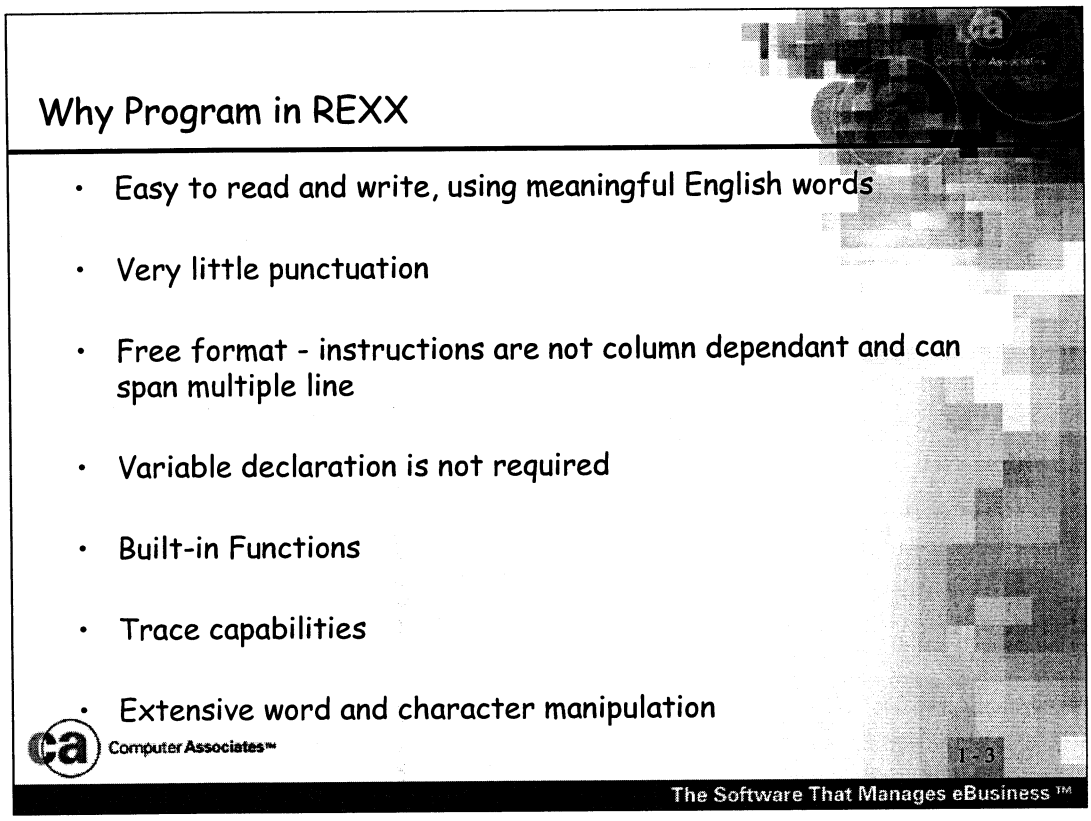
- Use proper syntax and demonstrating the fundamentals of good programming
- Use REXX functions as well as DO/WHILE loops and IF/THEN/ESE constructs
- Use Built-in functions such as SUBSTR, DATATYPE and POS
- Code basic user-written functions
- Write REXX program to pass commands to TSO
- Use the results of TSO commands in a REXX program.
- Read and write to QSAM datasets.
- Use REXX with Dialog manager panels.
- Use the TSO program stacks for storing and retrieving data.
- Capture TSO messages.

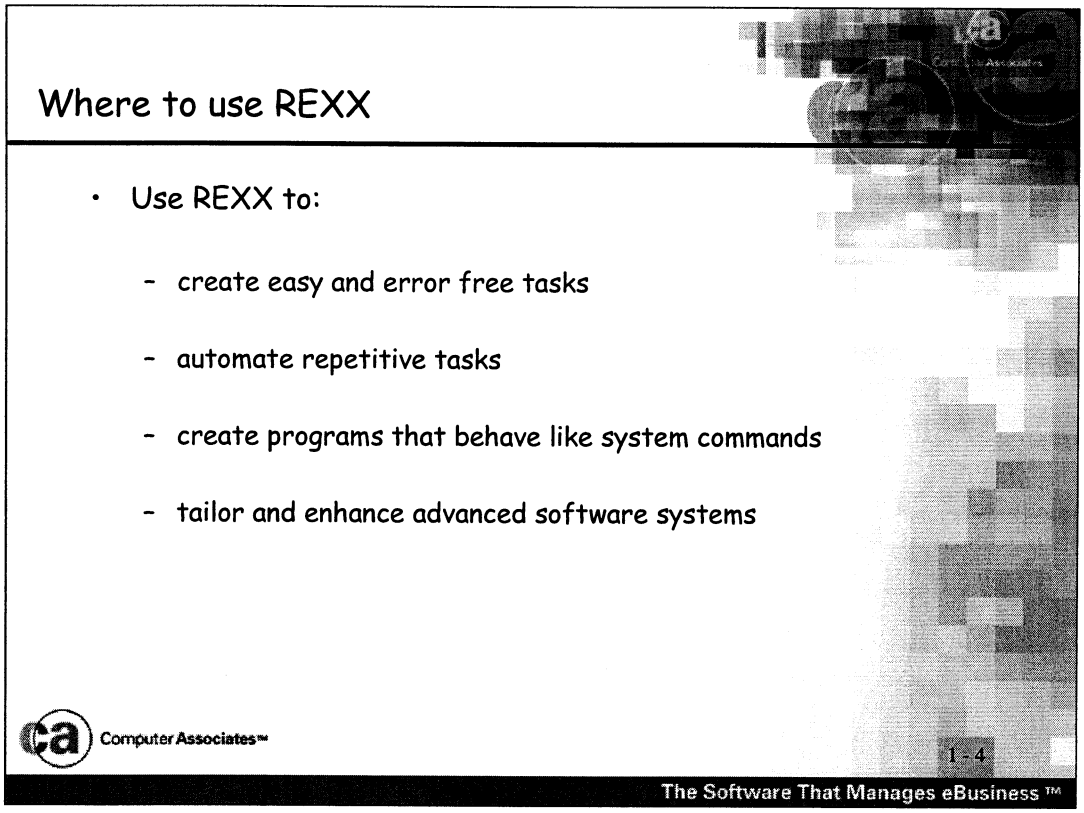

Course Outline
• Preliminary concepts
• Basic Components
• Flow Control Instructions
• Looping Instructions
• Built-in functions
• PARSE instructions
• ADDRESSing
• OUTTRAP
• EXECIO
• Data stacks
• Dialog Panels
Computer Associates™
I - 5
The Software That Manages eBusiness ™

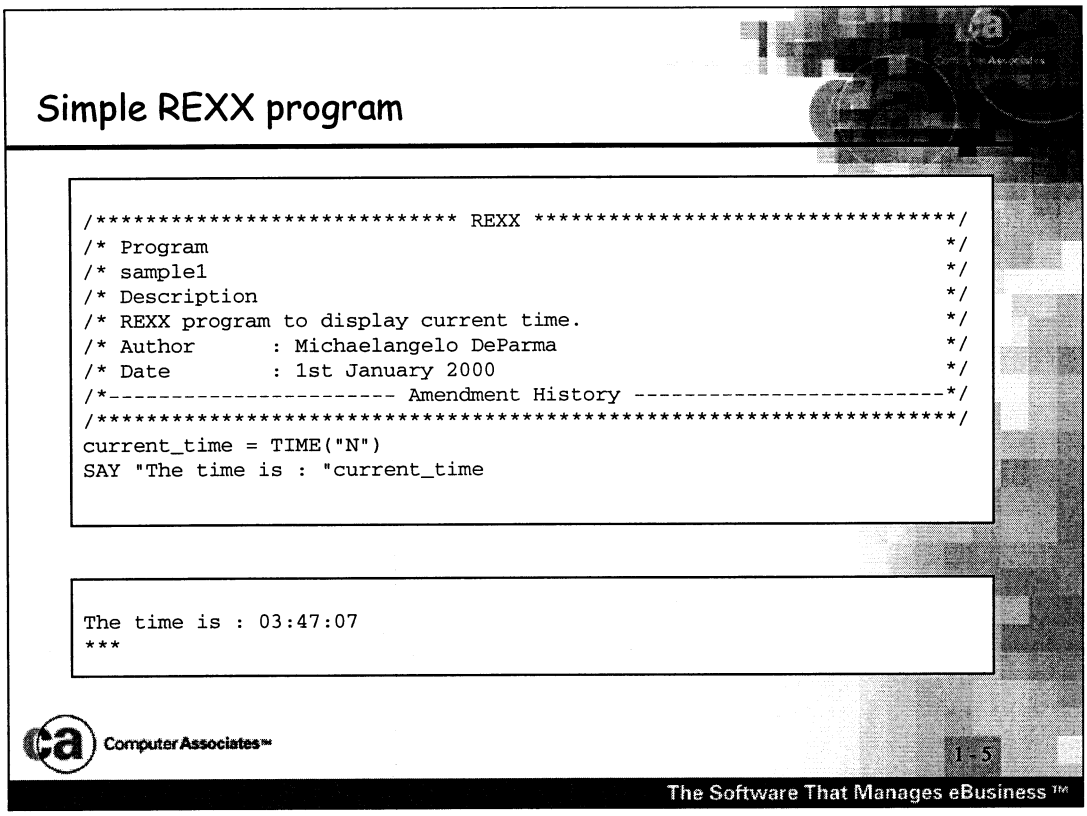

Are we on the right course ?
ca Computer Associates™
I - 6
The Software That Manages eBusiness ™

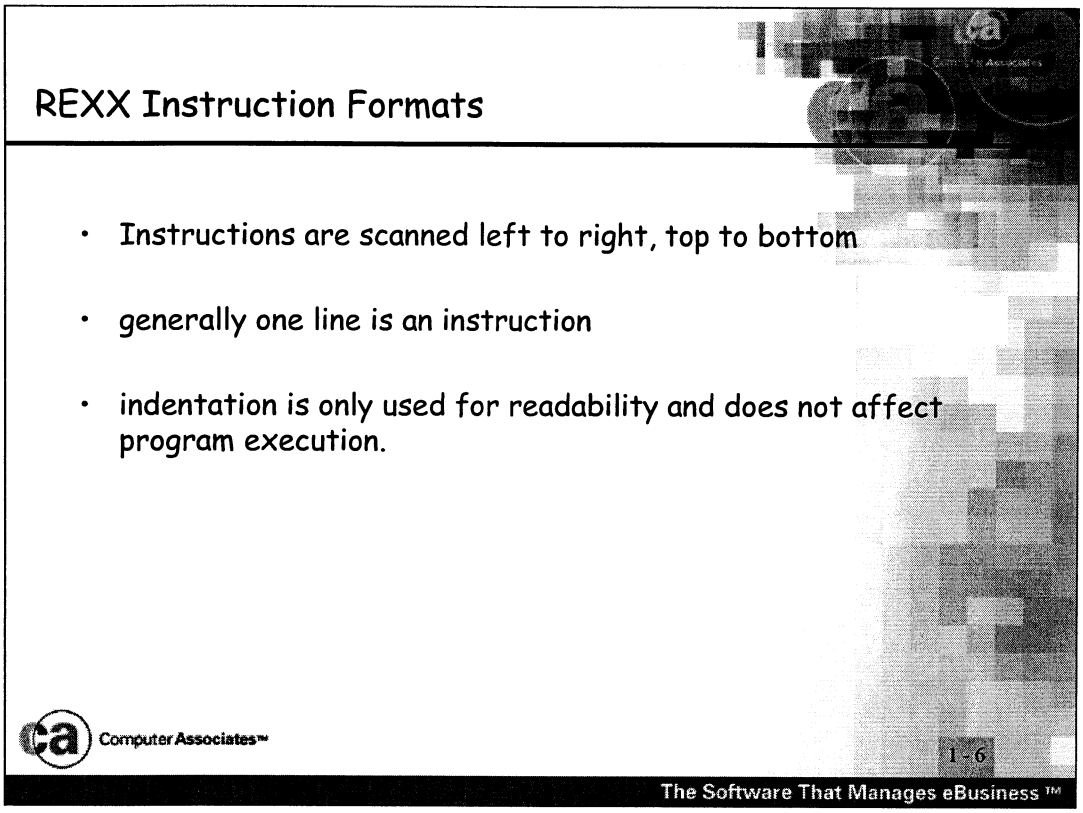

Concepts
Section 1
Computer Associates™
1 - 1
The Software That Manages eBusiness ™

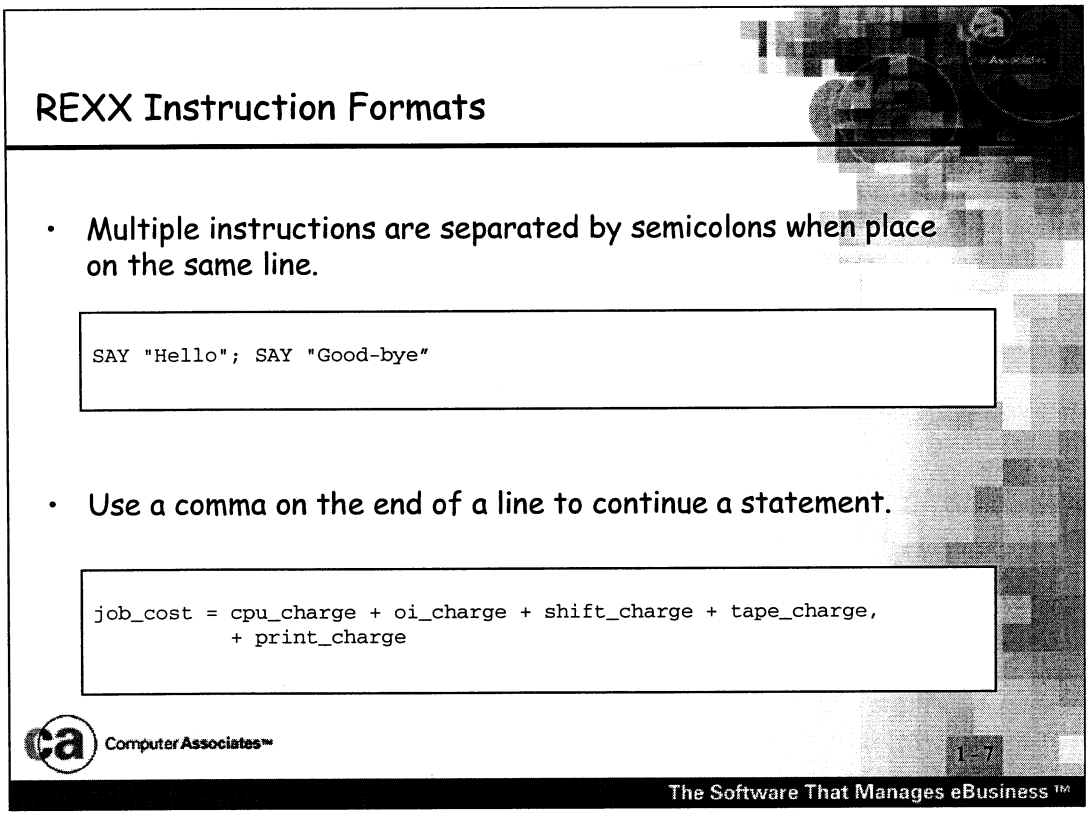

Brief History
• REstructured eXtended eXecutor
• Standard SAA procedure Language
• User driven development
• Designed to issue commands to the operating system
Computer Associates™
1 - 2
The Software That Manages eBusiness ™

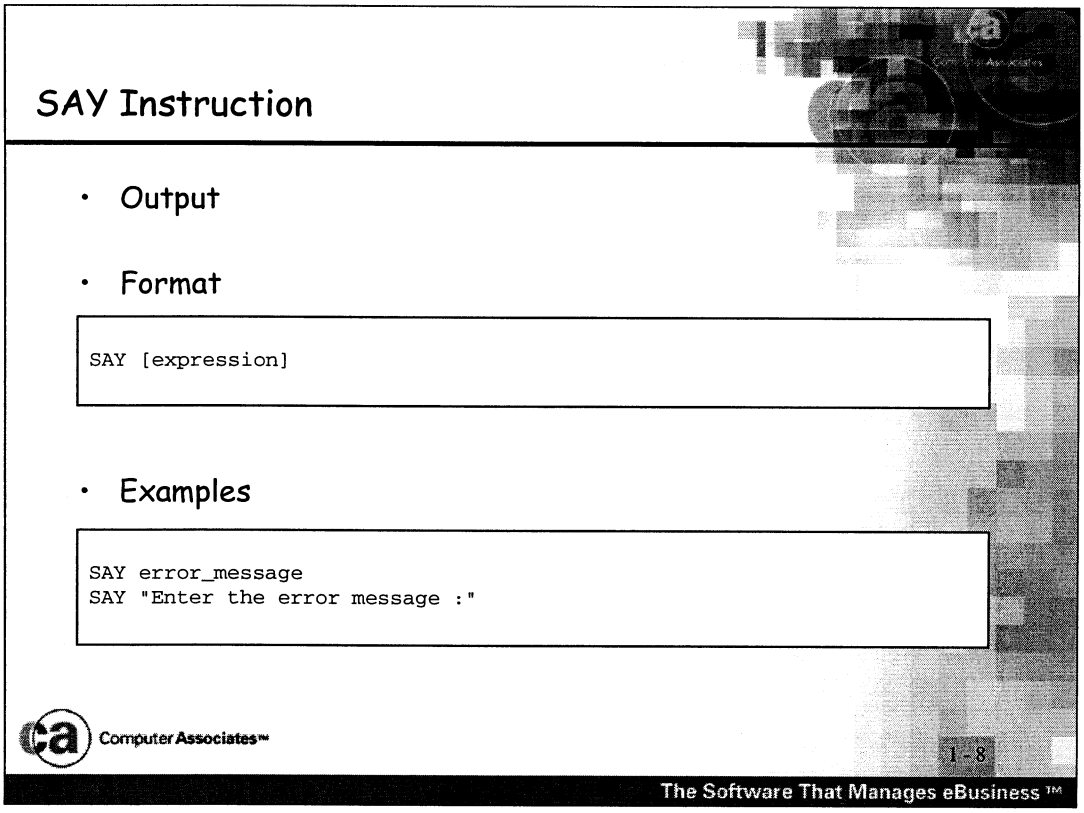

Why Program in REXX
• Easy to read and write, using meaningful English words
• Very little punctuation
• Free format - instructions are not column dependant and can span multiple line
• Variable declaration is not required
• Built-in Functions
• Trace capabilities
• Extensive word and character manipulation
Computer Associates™
1 - 3
The Software That Manages eBusiness ™

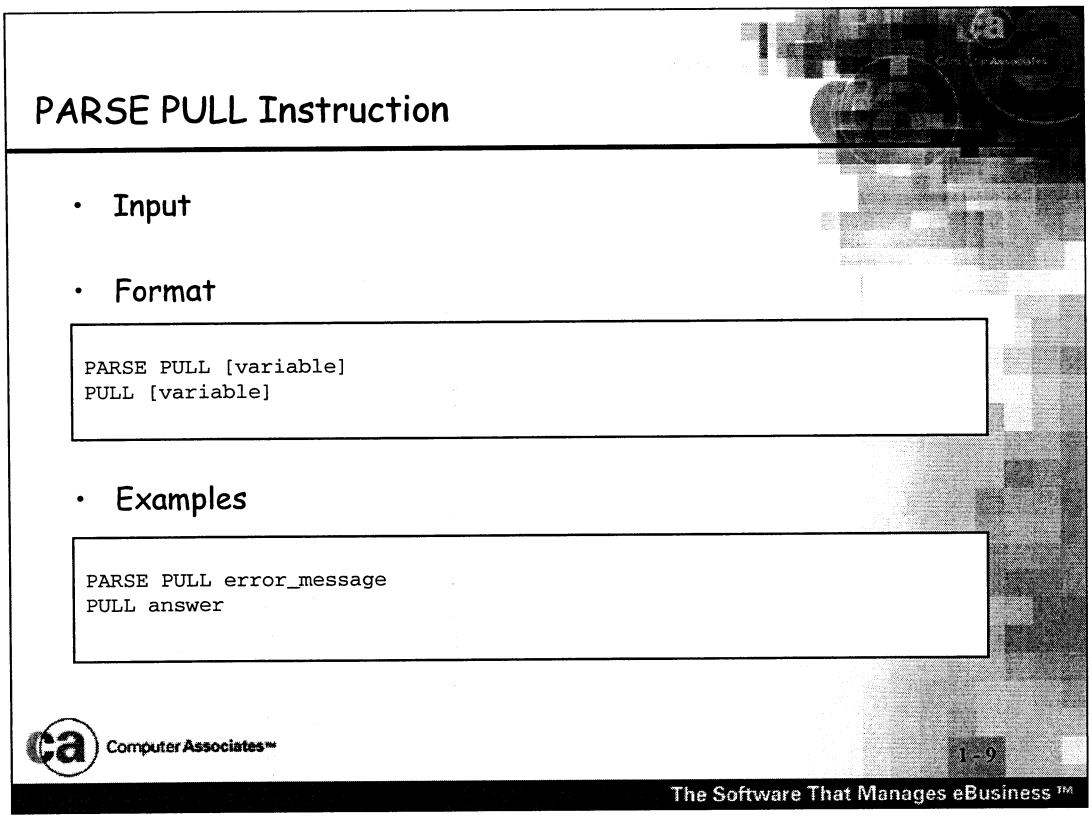

Where to use REXX
• Use REXX to:
- create easy and error free tasks
- automate repetitive tasks
- create programs that behave like system commands
- tailor and enhance advanced software systems
Computer Associates™
1-4
The Software That Manages eBusiness ™

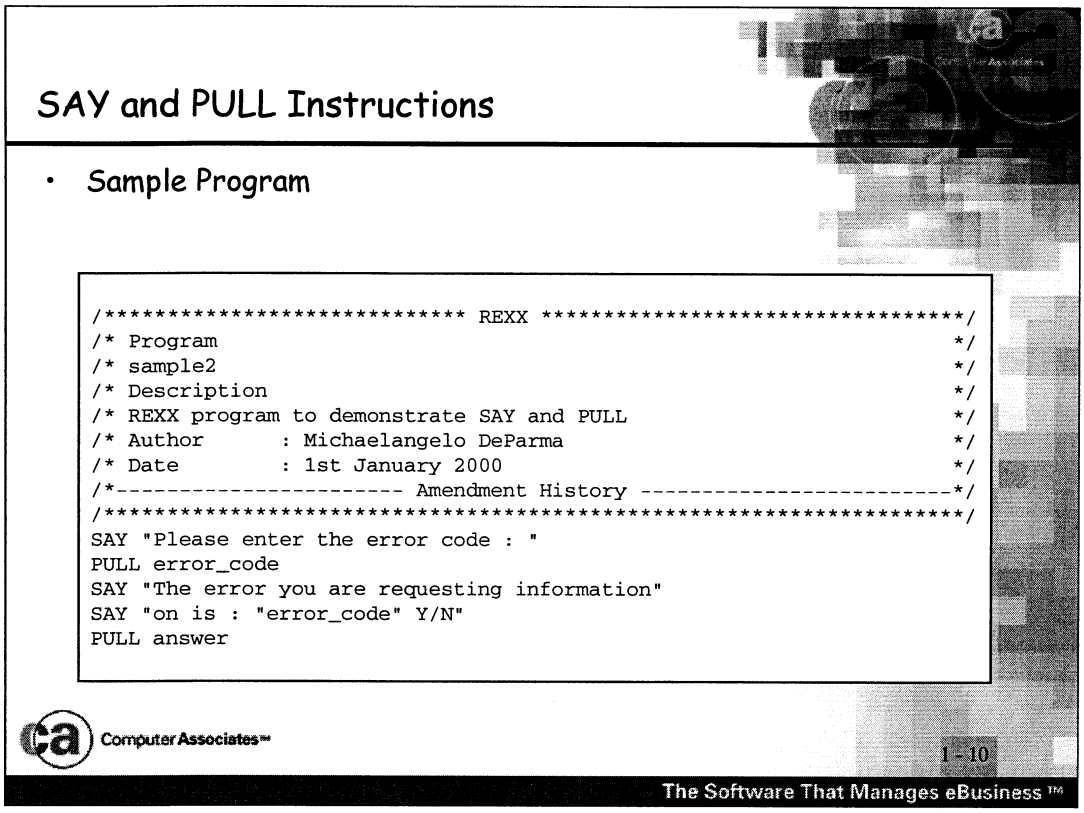

Simple REXX program
/****************************** REXX ***********************************/
/* Program   */
/* sample1   */
/* Description   */
/* REXX program to display current time.   */
/* Author   : Michaelangelo DeParma   */
/* Date   : 1st January 2000   */
/*---------------------- Amendment History -------------------------*/
/***/
current_time = TIME("N")
SAY "The time is : "current_time
The time is : 03:47:07
***
Computer Associates™
1 - 5
The Software That Manages eBusiness ™

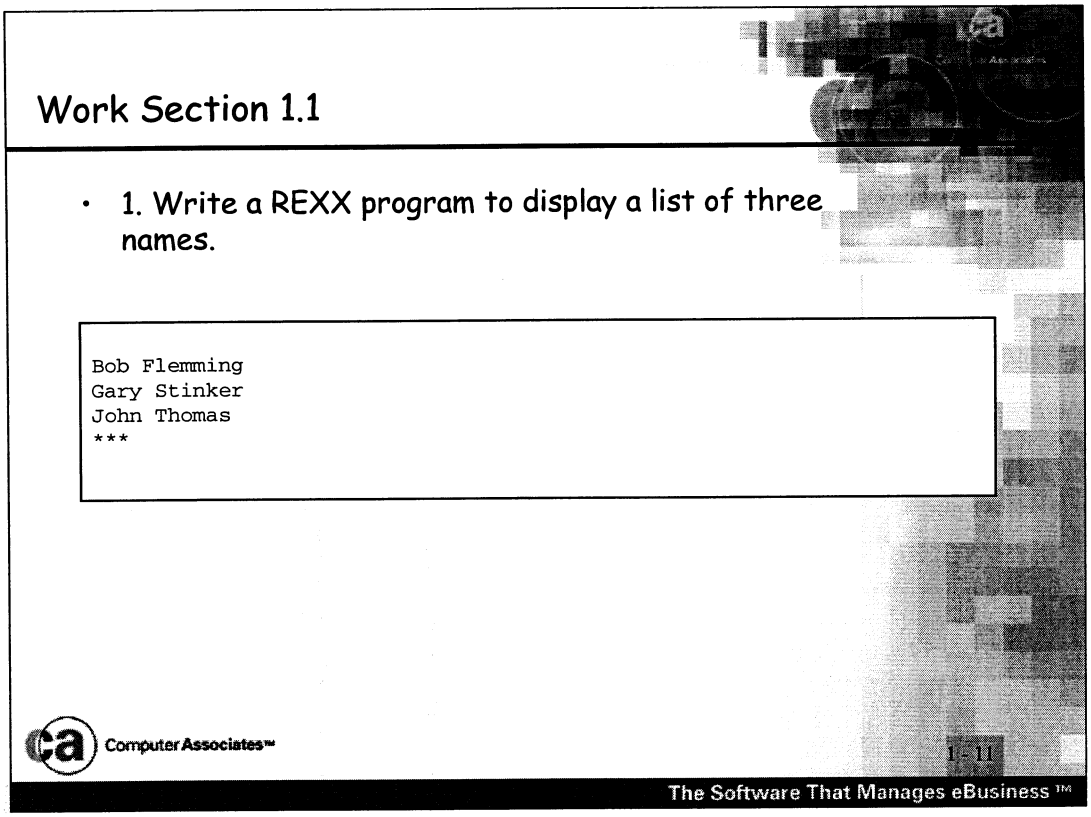

REXX Instruction Formats
• Instructions are scanned left to right, top to bottom
• generally one line is an instruction
• indentation is only used for readability and does not affect program execution.
Computer Associates™
1-6
The Software That Manages eBusiness ™


REXX Instruction Formats
• Multiple instructions are separated by semicolons when place on the same line.
SAY "Hello"; SAY "Good-bye"
• Use a comma on the end of a line to continue a statement.
job_cost = cpu_charge + oi_charge + shift_charge + tape_charge,
+ print_charge
Computer Associates™
1-7
The Software That Manages eBusiness ™


SAY Instruction
• Output
• Format
SAY [expression]
• Examples
SAY error_message
SAY "Enter the error message :"
Computer Associates™
1 - 8
The Software That Manages eBusiness ™


PARSE PULL Instruction
• Input
• Format
PARSE PULL [variable]
PULL [variable]
• Examples
PARSE PULL error_message
PULL answer
Computer Associates™
1-9
The Software That Manages eBusiness ™


SAY and PULL Instructions
• Sample Program
/****************************** REXX ***********************************/
/* Program   */
/* sample2   */
/* Description   */
/* REXX program to demonstrate SAY and PULL   */
/* Author   : Michaelangelo DeParma   */
/* Date   : 1st January 2000   */
/*----------------------- Amendment History --------------------------*/
/***/
SAY "Please enter the error code : "
PULL error_code
SAY "The error you are requesting information"
SAY "on is : "error_code" Y/N"
PULL answer
Computer Associates™
1-10
The Software That Manages eBusiness ™



Use your dataset for all the REXX programs.

CLCS.IULCnn.REXX

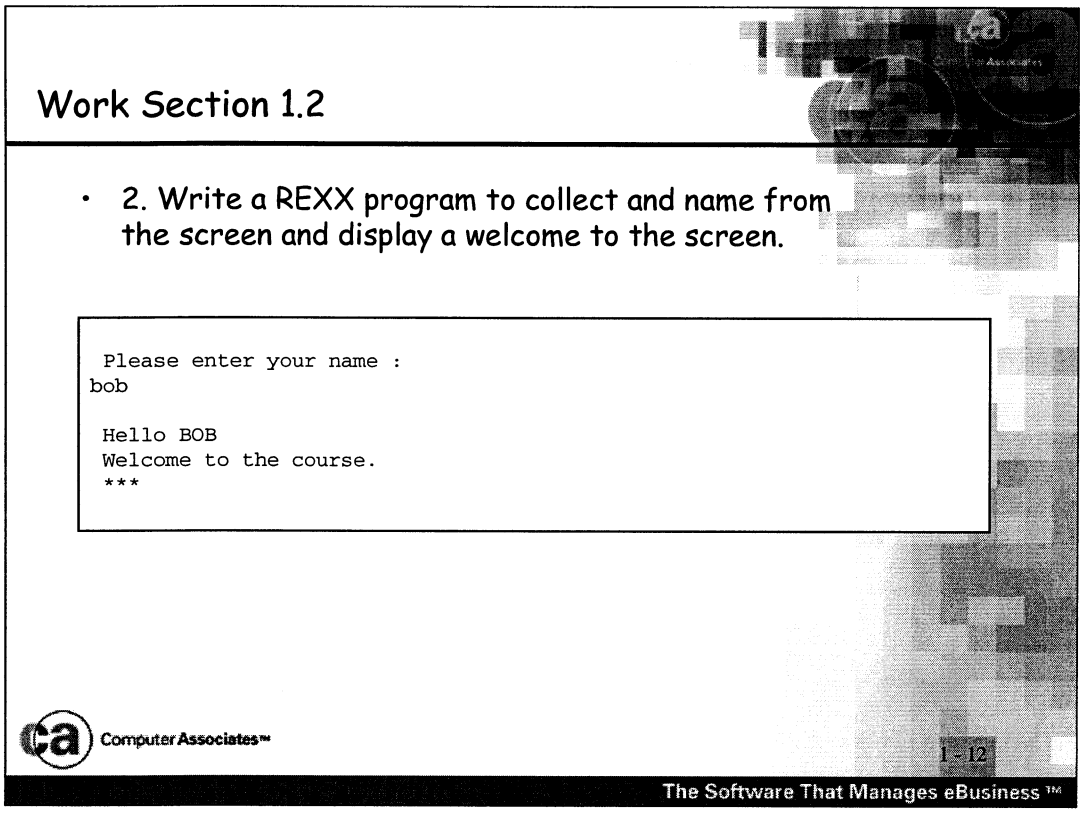
Execute the program from ispf option 6

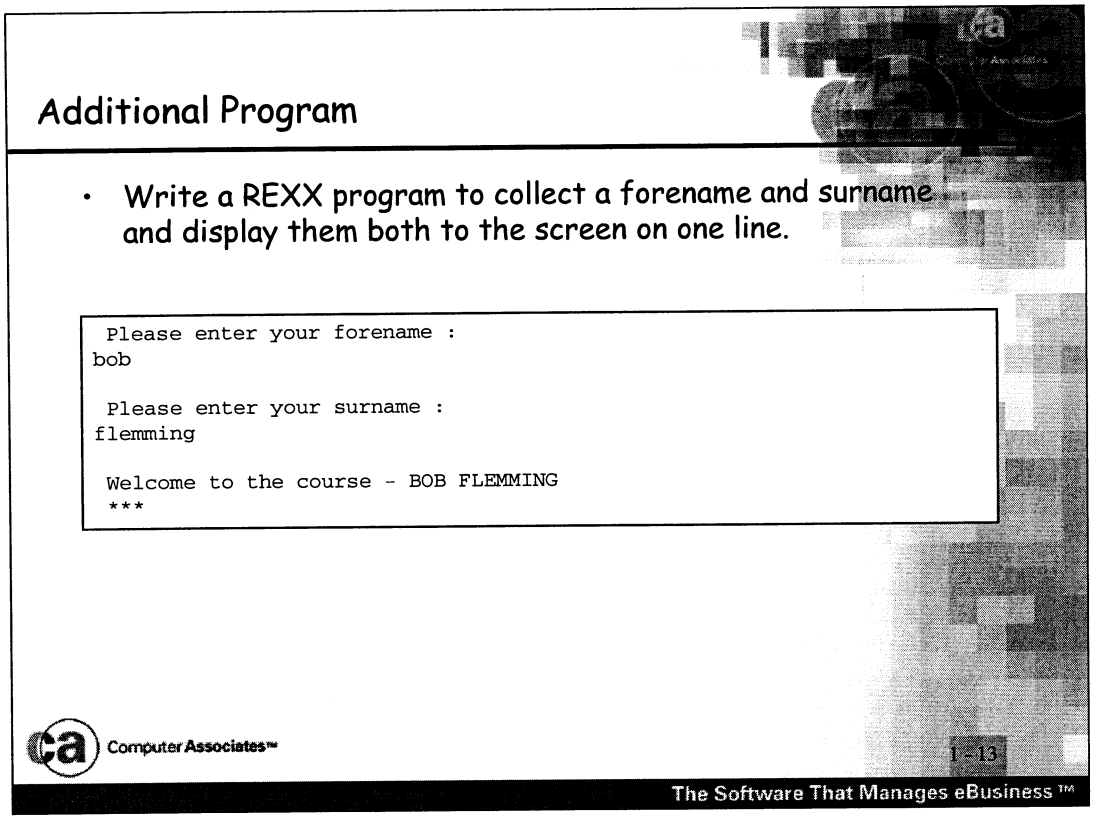
```
Menu  List  Mode  Functions  Utilities  Help
 ------------------------------------------------------------------------------
                              ISPF Command Shell
 Enter TSO or Workstation commands below:

 ===> ex 'clcs.iulc01.rexx(test)'
```

Replace nn with your user number.

E.g. for id IULC35

CLCS.IULC35.REXX


Work Section 1.2
• 2. Write a REXX program to collect and name from the screen and display a welcome to the screen.
Please enter your name :
bob
Hello BOB
Welcome to the course.
***
Computer Associates™
1 - 12
The Software That Manages eBusiness ™


Additional Program
• Write a REXX program to collect a forename and surname and display them both to the screen on one line.
Please enter your forename :
bob
Please enter your surname :
flemming
Welcome to the course - BOB FLEMMING
***
Computer Associates™
1-13
The Software That Manages eBusiness ™

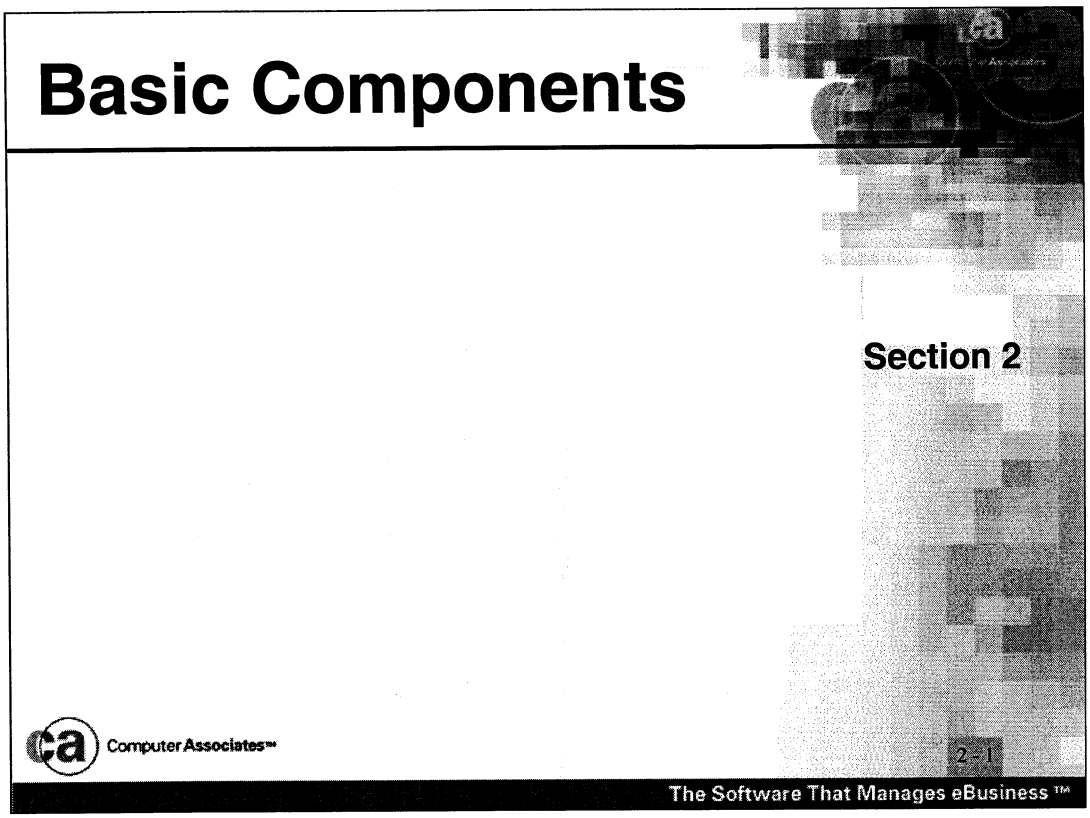
Additional Program

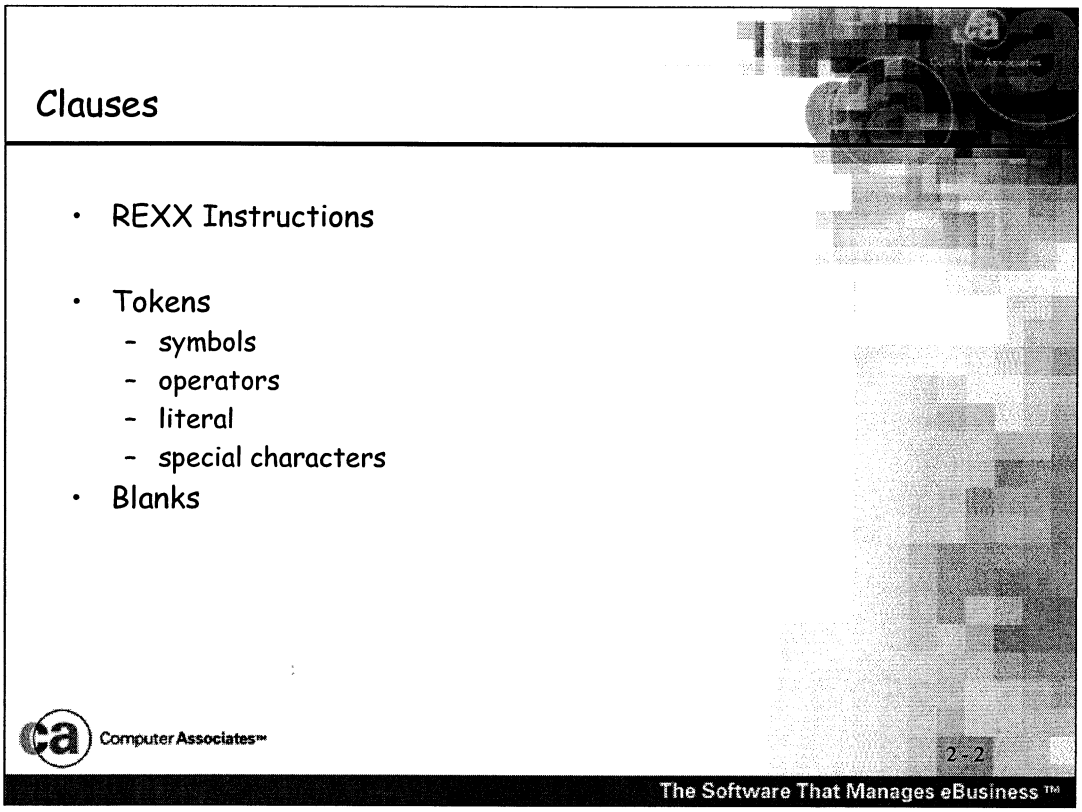
- Write a REXX program to collect a forename and surname on one line and display them both to the screen in reverse order..

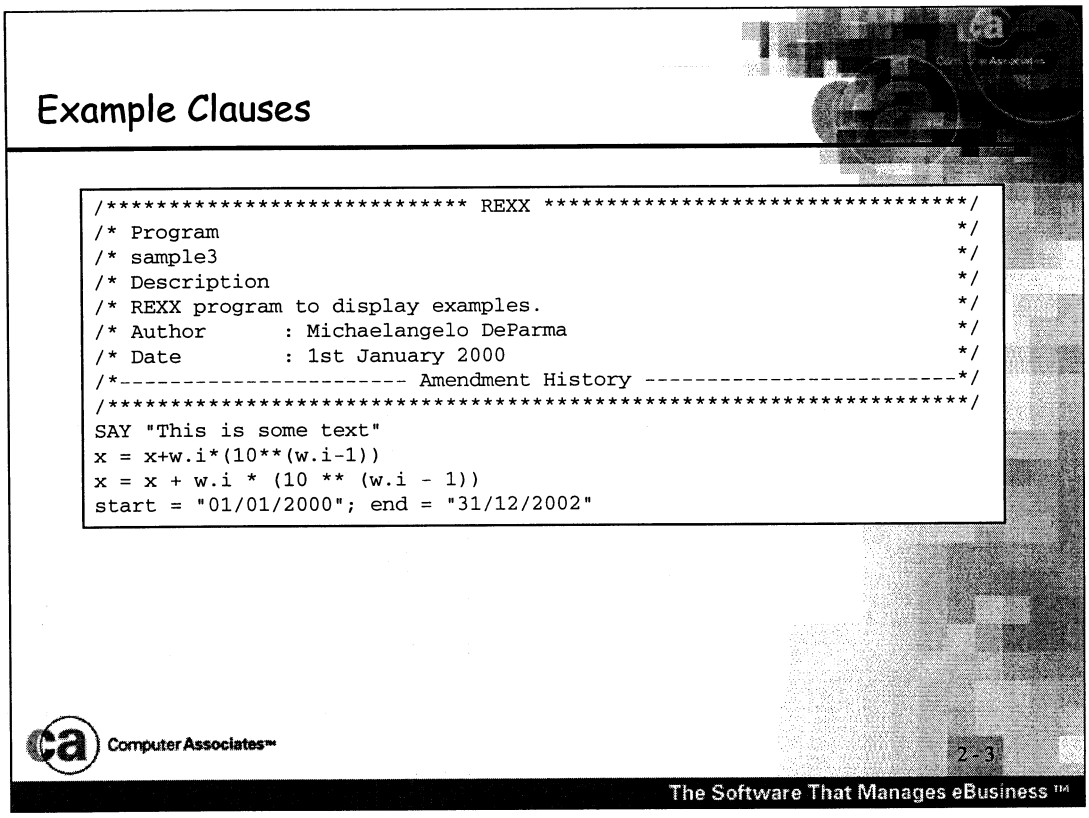
```
  Please enter your Forename and Surname :
bob flemming

 Welcome to the course - FLEMMING BOB
 ***
```

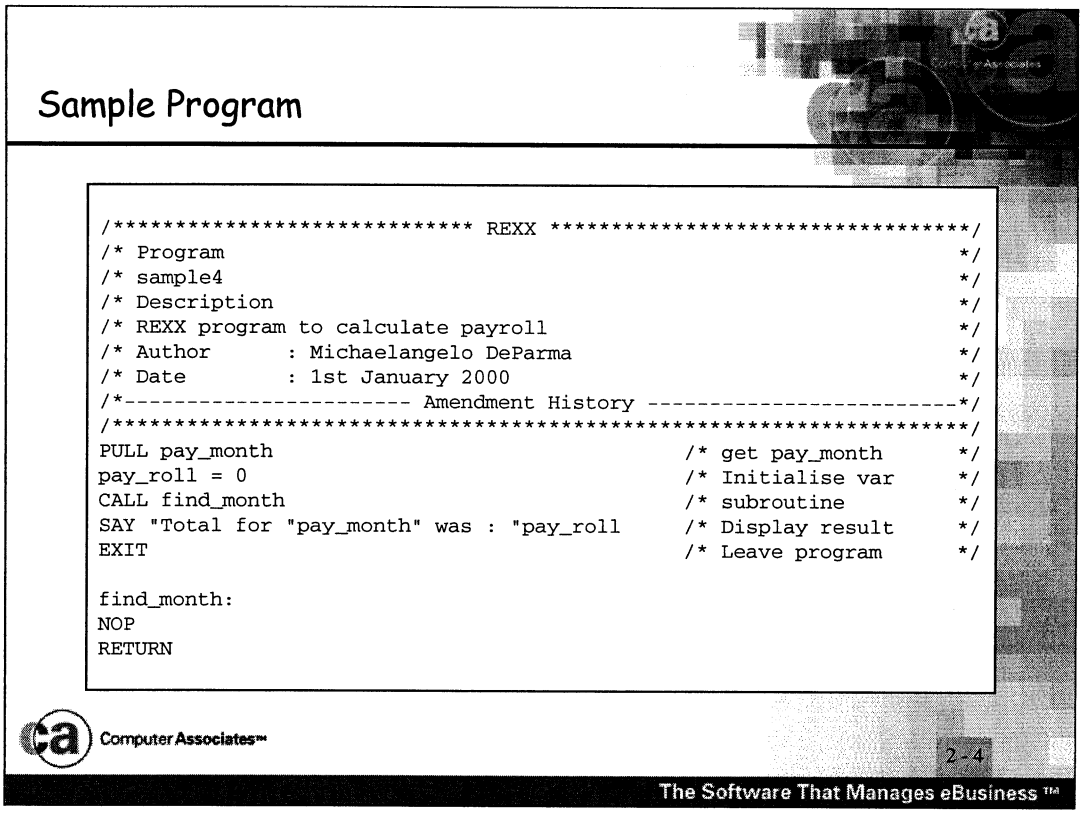
1 - 14

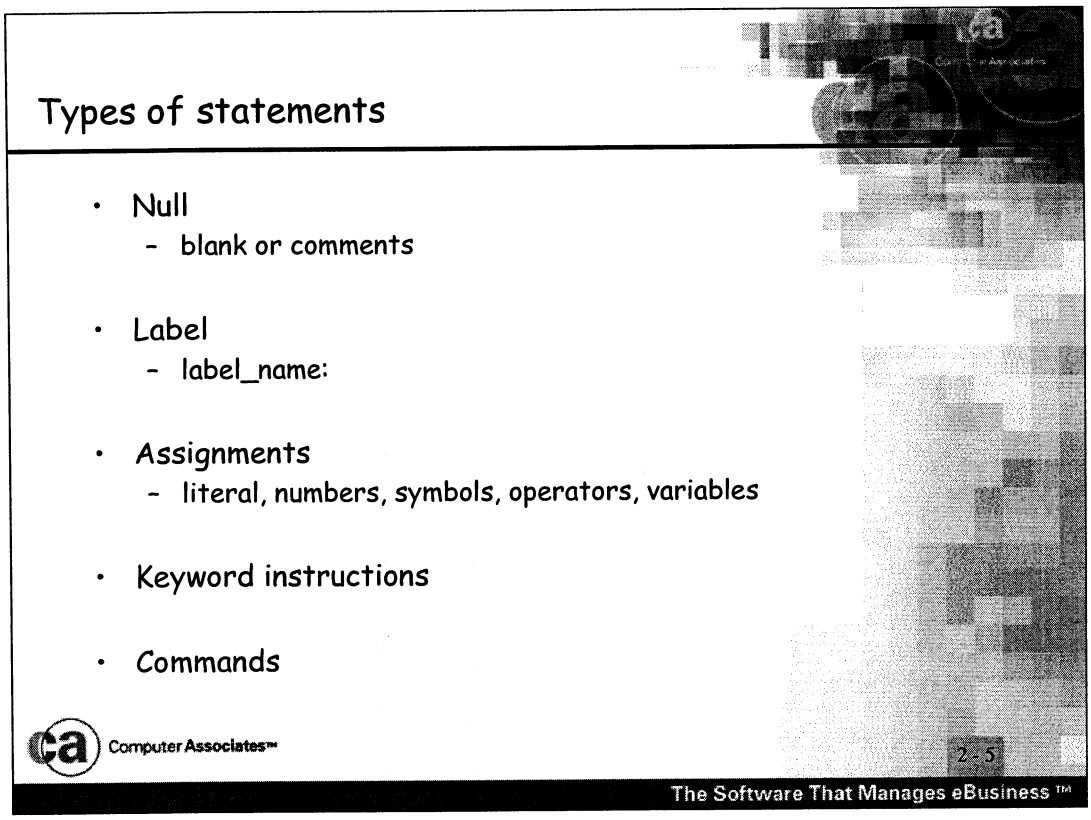
The Software That Manages eBusiness ™

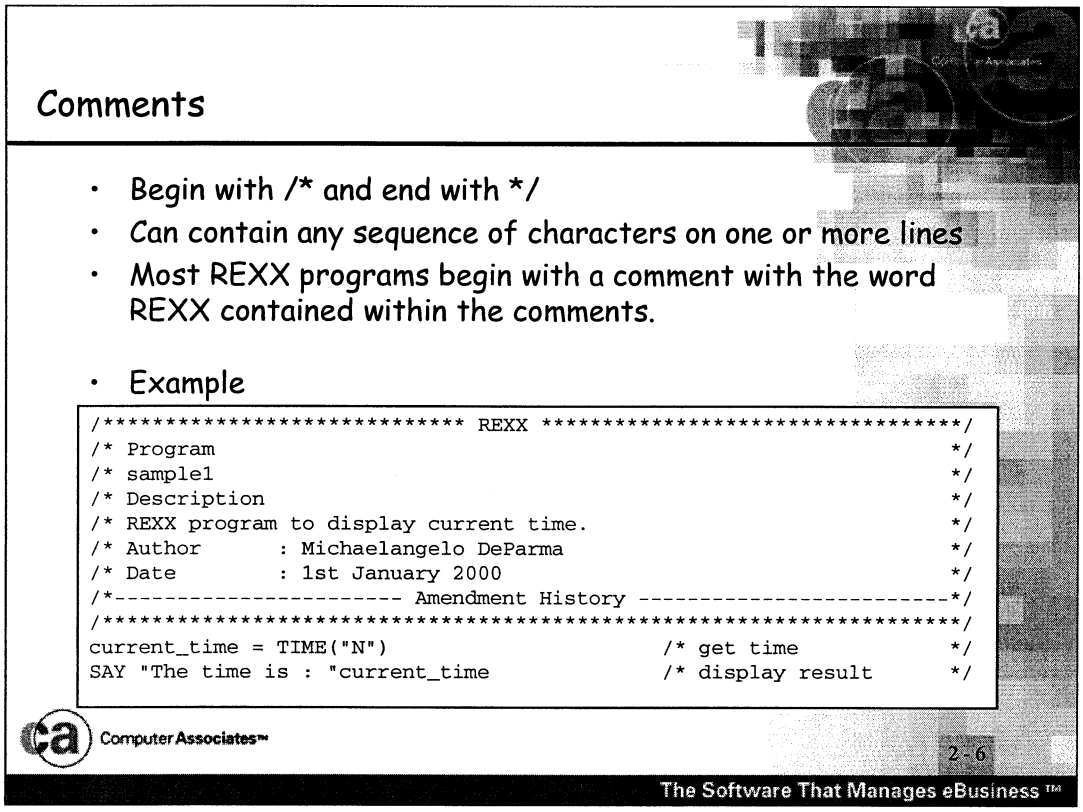

Basic Components
Section 2
Computer Associates™
2 - 1
The Software That Manages eBusiness ™

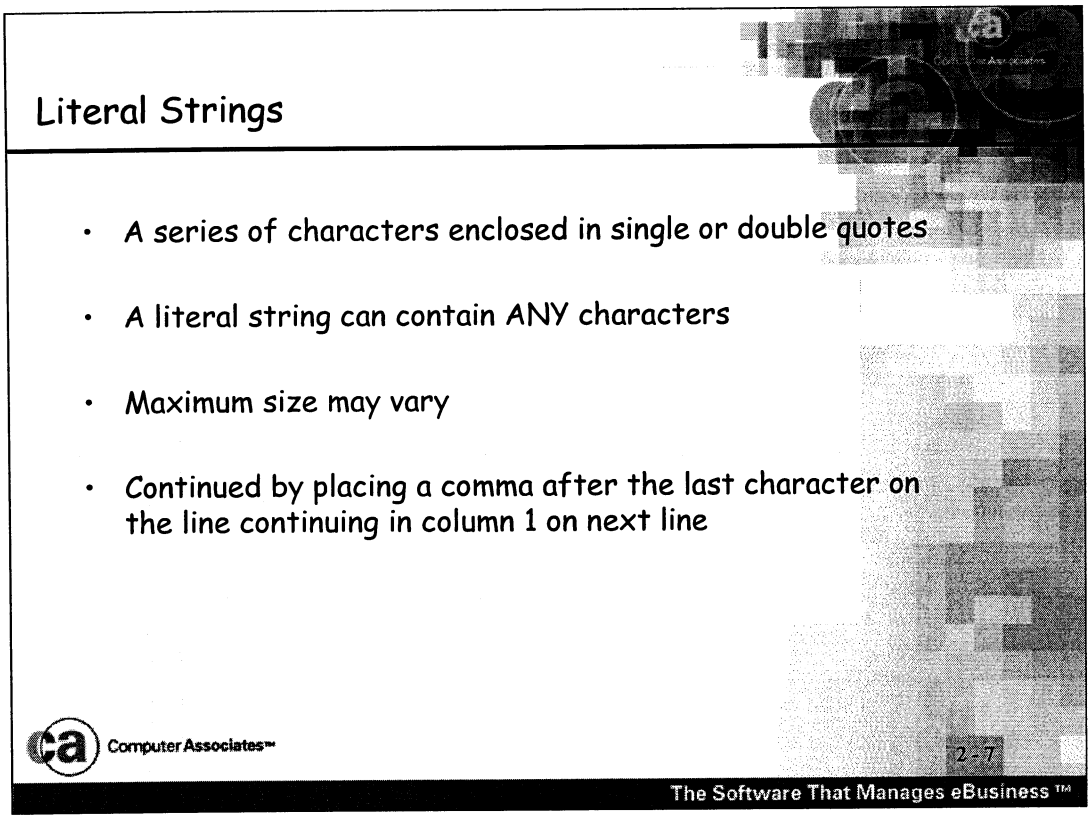

Clauses
• REXX Instructions
• Tokens
- symbols
- operators
- literal
- special characters
• Blanks
Computer Associates™
2-2
The Software That Manages eBusiness ™

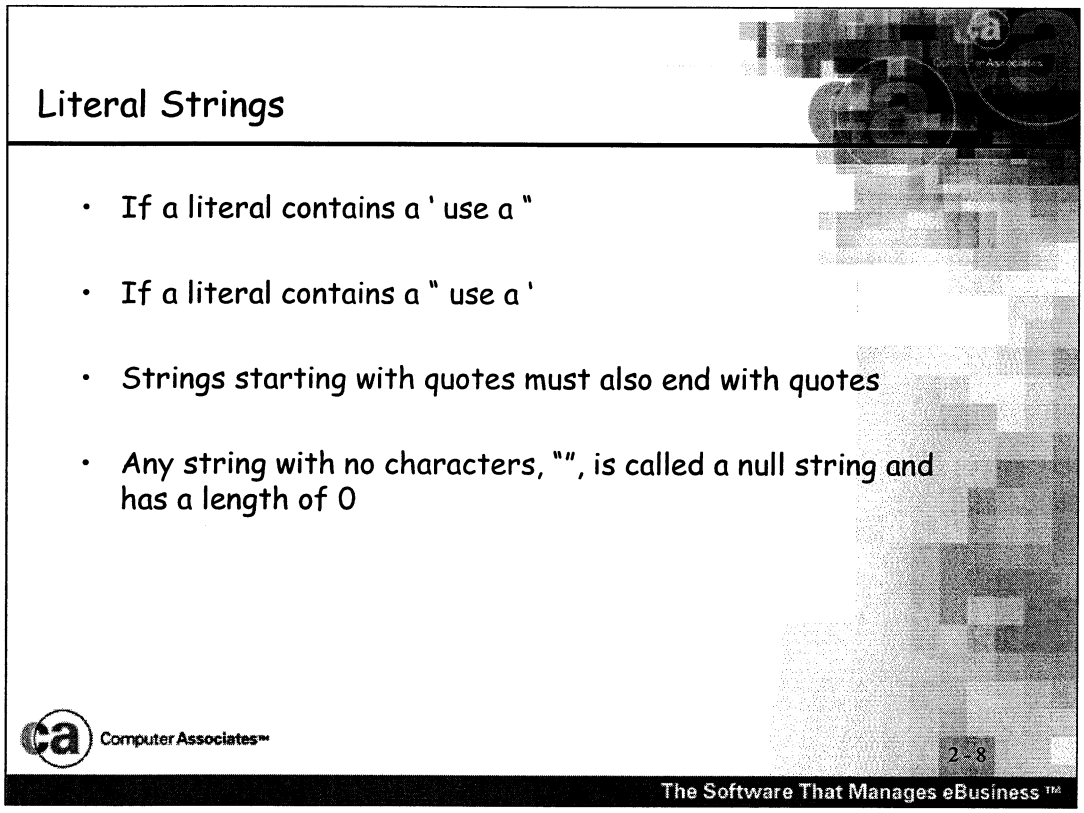

Example Clauses
/***************************** REXX ********************************/
/* Program   */
/* sample3   */
/* Description   */
/* REXX program to display examples.   */
/* Author   : Michaelangelo DeParma   */
/* Date   : 1st January 2000   */
/*----------------------- Amendment History -------------------------*/
/***/
SAY "This is some text"
x = x+w.i*(10**(w.i-1))
x = x + w.i * (10 ** (w.i - 1))
start = "01/01/2000"; end = "31/12/2002"
Computer Associates™
2-3
The Software That Manages eBusiness ™

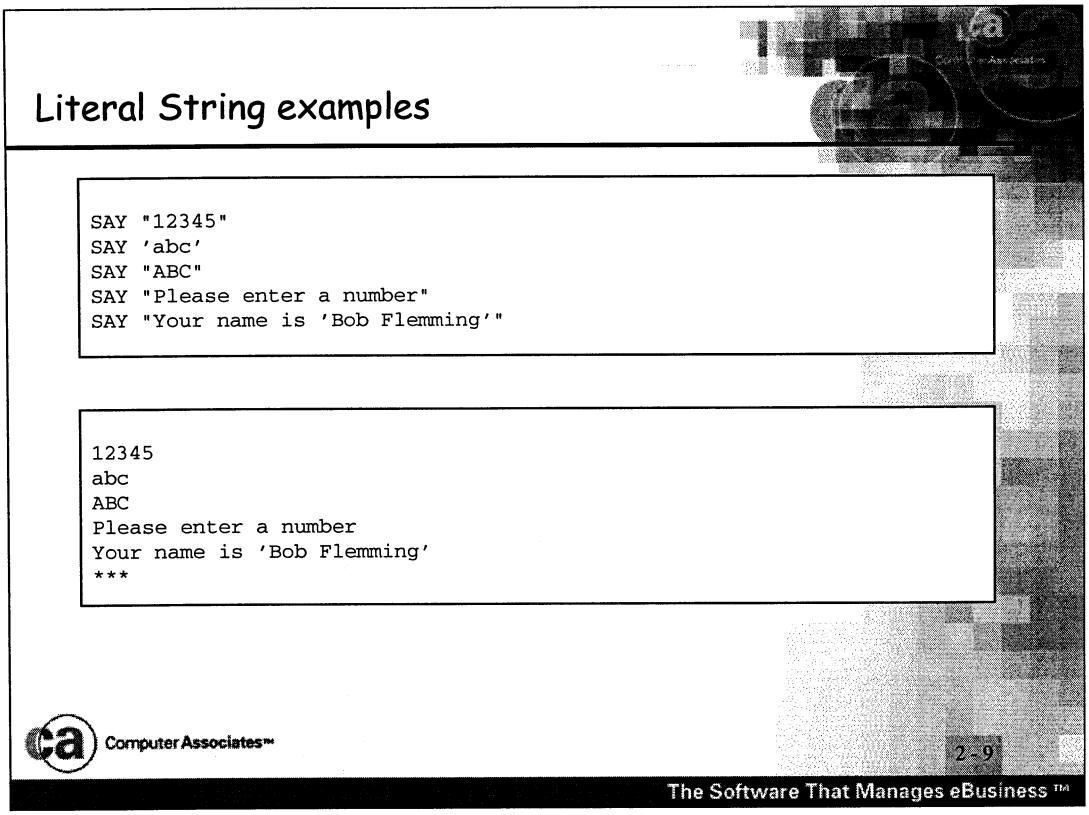

Sample Program
/***************************** REXX ********************************/
/* Program   */
/* sample4   */
/* Description   */
/* REXX program to calculate payroll   */
/* Author   : Michaelangelo DeParma   */
/* Date   : 1st January 2000   */
/*---------------------- Amendment History -------------------------*/
/***/
PULL pay_month   /* get pay_month   */
pay_roll = 0   /* Initialise var   */
CALL find_month   /* subroutine   */
SAY "Total for "pay_month" was : "pay_roll /* Display result   */
EXIT   /* Leave program   */
find_month:
NOP
RETURN
Computer Associates™
2-4
The Software That Manages eBusiness ™

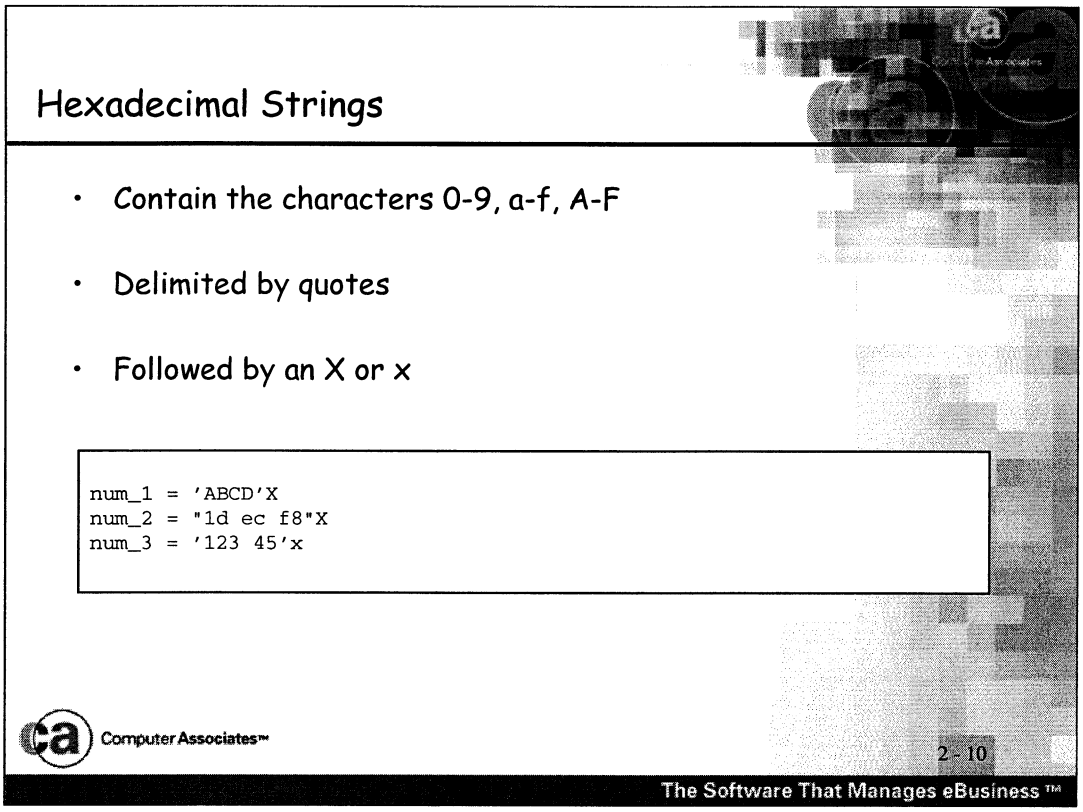

Types of statements
• Null
- blank or comments
• Label
- label_name:
• Assignments
- literal, numbers, symbols, operators, variables
• Keyword instructions
• Commands
Computer Associates™
2-5
The Software That Manages eBusiness ™

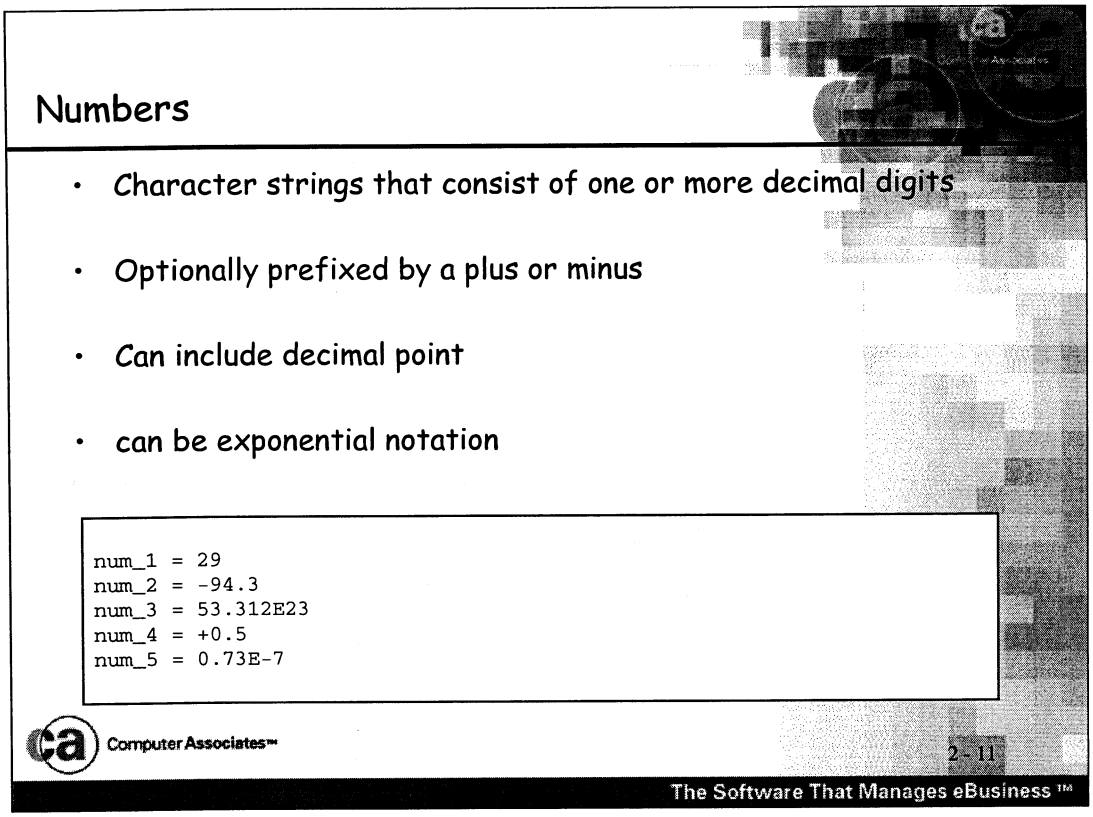

Comments
• Begin with /* and end with */
• Can contain any sequence of characters on one or more lines
• Most REXX programs begin with a comment with the word REXX contained within the comments.
• Example
/***************************** REXX ********************************/
/* Program   */
/* sample1   */
/* Description   */
/* REXX program to display current time.   */
/* Author   : Michaelangelo DeParma   */
/* Date   : 1st January 2000   */
/*----------------------- Amendment History -------------------------*/
/***/
current_time = TIME("N")   /* get time   */
SAY "The time is : "current_time   /* display result   */
ComputerAssociates™
2-6
The Software That Manages eBusiness ™


Literal Strings
• A series of characters enclosed in single or double quotes
• A literal string can contain ANY characters
• Maximum size may vary
• Continued by placing a comma after the last character on the line continuing in column 1 on next line
Computer Associates™
2-7
The Software That Manages eBusiness ™


Literal Strings
• If a literal contains a ' use a "
• If a literal contains a " use a '
• Strings starting with quotes must also end with quotes
• Any string with no characters, "", is called a null string and has a length of 0
ca Computer Associates™
2-8
The Software That Manages eBusiness ™


Literal String examples
SAY "12345"
SAY 'abc'
SAY "ABC"
SAY "Please enter a number"
SAY "Your name is 'Bob Flemming'"
12345
abc
ABC
Please enter a number
Your name is 'Bob Flemming'
***
Computer Associates™
2-9
The Software That Manages eBusiness ™


Hexadecimal Strings
• Contain the characters 0-9, a-f, A-F
• Delimited by quotes
• Followed by an X or x
num_1 = 'ABCD'X
num_2 = "1d ec f8"X
num_3 = '123 45'x
Computer Associates™
2 - 10
The Software That Manages eBusiness ™

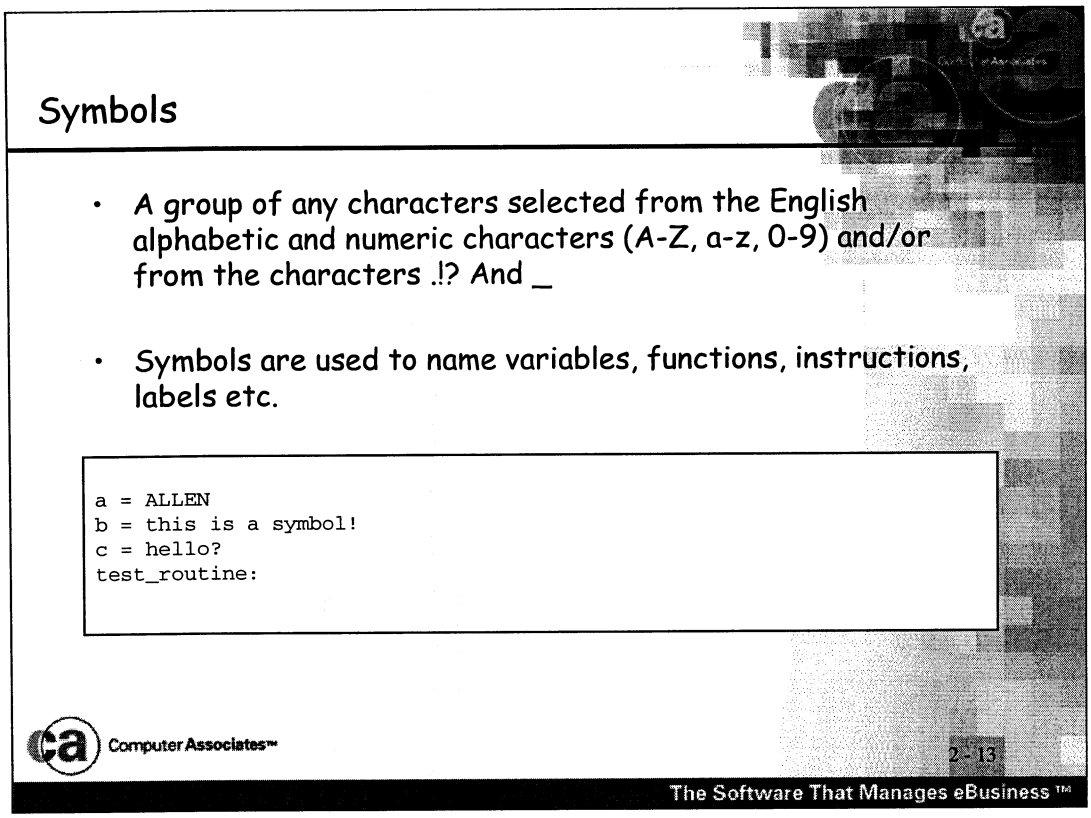

Numbers
• Character strings that consist of one or more decimal digits
• Optionally prefixed by a plus or minus
• Can include decimal point
• can be exponential notation
num_1 = 29
num_2 = -94.3
num_3 = 53.312E23
num_4 = +0.5
num_5 = 0.73E-7
Computer Associates™
2-11
The Software That Manages eBusiness ™

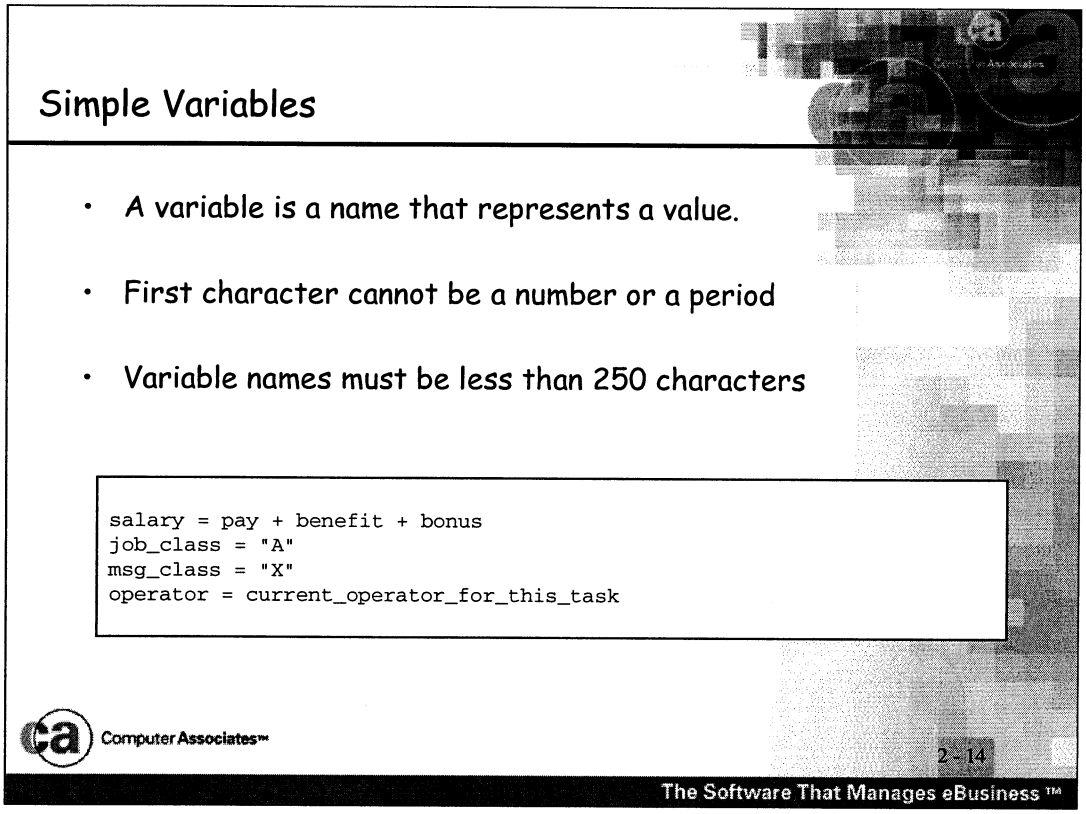
Test Exercise 21

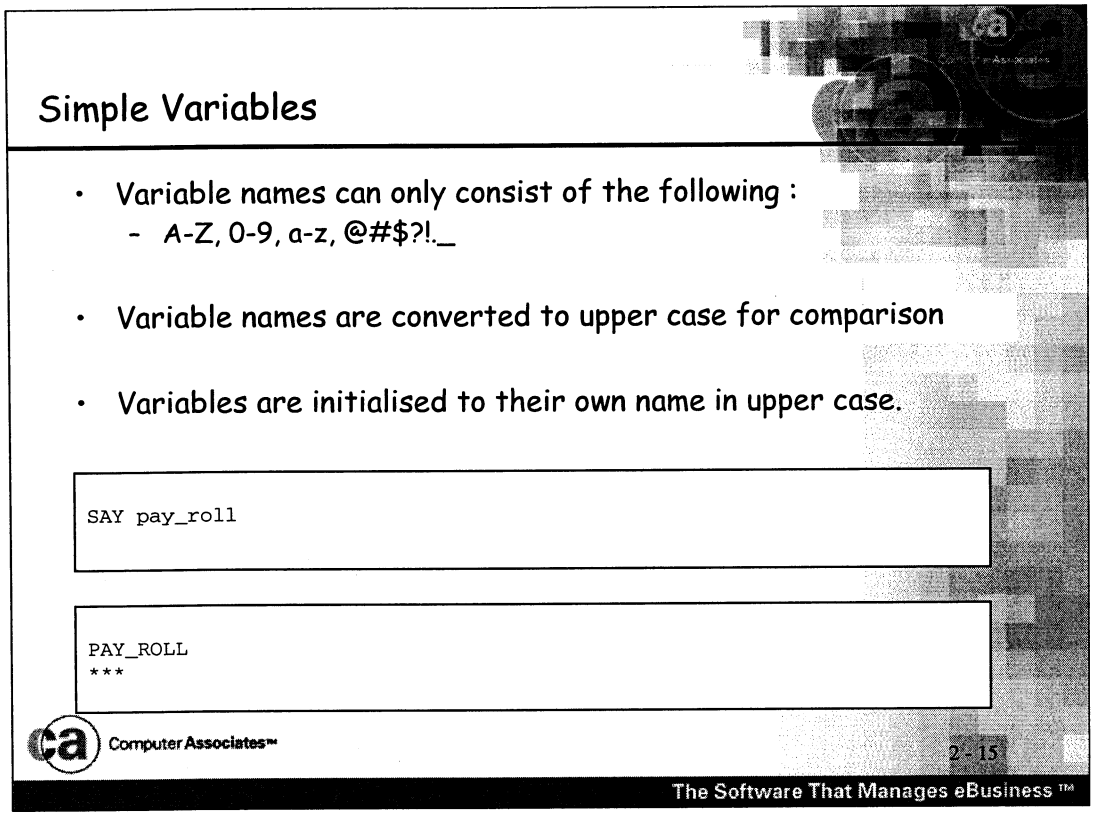
- Write a REXX program using the following code. Test and identify the type of statements. Which statement is :
  - Comment
  - Literal
  - Hexadecimal
  - number
  - error

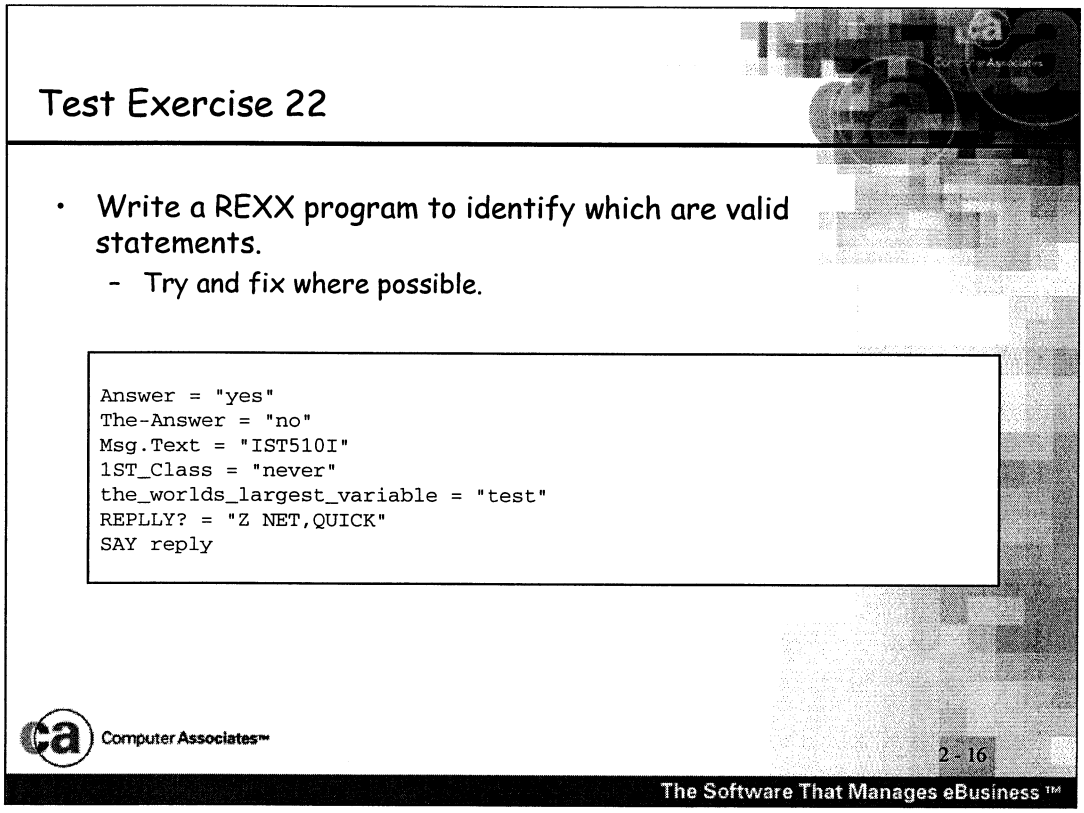
```
/* REXX EXEC */
a = "REXX EXEC"
c = 94 4F
d = '94 4F'
e = 32
f = '32'
g = "32"x
b = "REXX EXEC"x
REXX EXEC
```

The Software That Manages eBusiness ™


Symbols
• A group of any characters selected from the English alphabetic and numeric characters (A-Z, a-z, 0-9) and/or from the characters .!? And _
• Symbols are used to name variables, functions, instructions, labels etc.
a = ALLEN
b = this is a symbol!
c = hello?
test_routine:
Computer Associates™
2-13
The Software That Manages eBusiness ™


Simple Variables
• A variable is a name that represents a value.
• First character cannot be a number or a period
• Variable names must be less than 250 characters
salary = pay + benefit + bonus
job_class = "A"
msg_class = "X"
operator = current_operator_for_this_task
Computer Associates™
2-14
The Software That Manages eBusiness ™


Simple Variables
• Variable names can only consist of the following :
- A-Z, 0-9, a-z, @#$?!._
• Variable names are converted to upper case for comparison
• Variables are initialised to their own name in upper case.
SAY pay_roll
PAY_ROLL
***
Computer Associates™
2-15
The Software That Manages eBusiness ™


Test Exercise 22
• Write a REXX program to identify which are valid statements.
- Try and fix where possible.
Answer = "yes"
The-Answer = "no"
Msg.Text = "IST510I"
1ST_Class = "never"
the_worlds_largest_variable = "test"
REPLLY? = "Z NET,QUICK"
SAY reply
Computer Associates™
2 - 16
The Software That Manages eBusiness ™

Compound Variables

- A technique for grouping variables
- Start with a stem which is a valid variable name followed by a period (.)
- Characters after the first period are called tail

```
CPUTIME.ASID
Stem    Tail

new_rate.shift.type.weekly
Stem     Tail  Tail Tail
```

Computer Associates™

2 - 17

The Software That Manages eBusiness ™

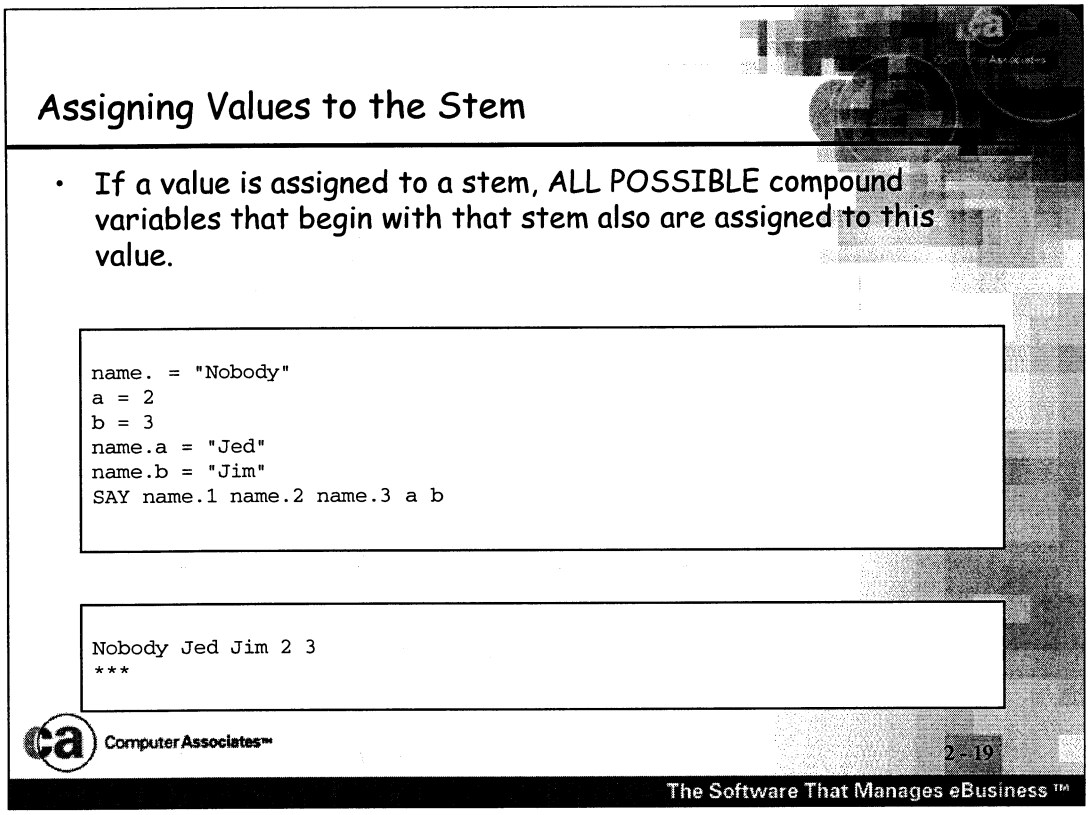
Compound Variables

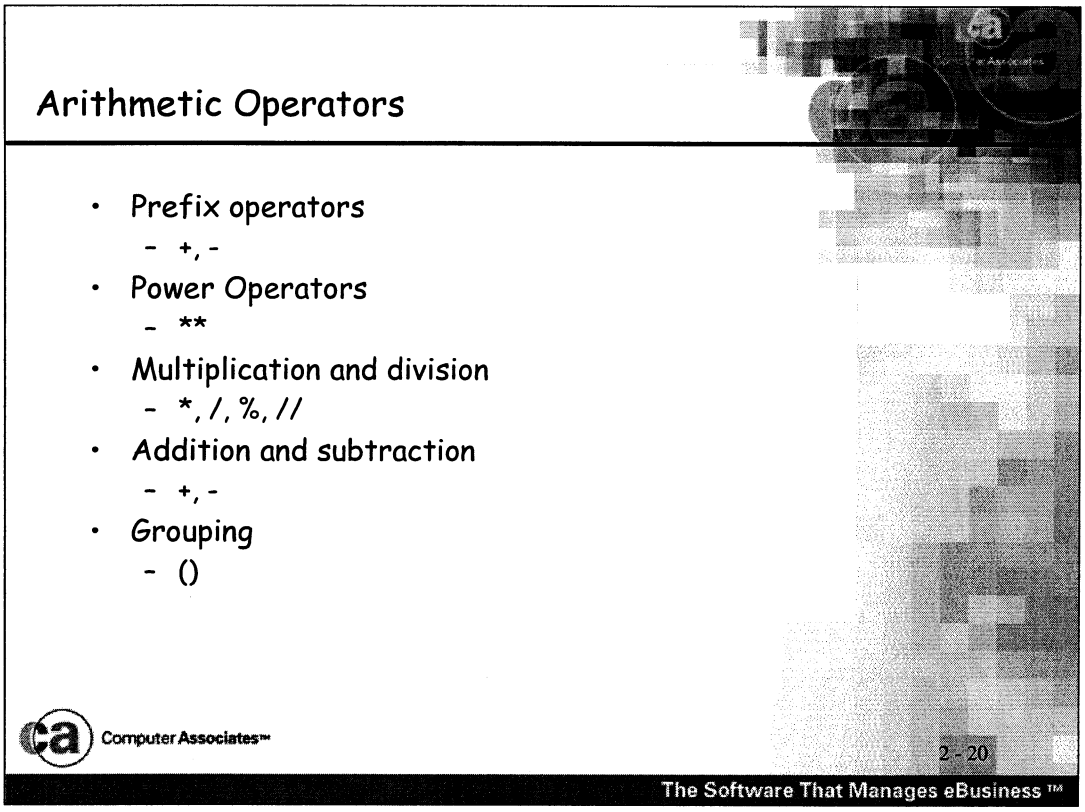
- Tails are treated as a series of simple variable names separated by periods
- At execution time, each variable is replaced by its current value to generate the derived name.

```
/***************************** REXX ********************************/
/* Program                                                         */
/* test                                                            */
/* Description                                                     */
/* REXX program to test statements                                 */
/* Author       : Michaelangelo DeParma                            */
/* Date         : 1st January 2000                                 */
/*----------------------- Amendment History -------------------------*/
/***************************************************************************/
tgif = 3
day.1 = "Mon"
day.2 = "Tue"
day.3 = "Wed"
SAY day.tgif
```

2 - 18

The Software That Manages eBusiness ™


Assigning Values to the Stem
• If a value is assigned to a stem, ALL POSSIBLE compound variables that begin with that stem also are assigned to this value.
name. = "Nobody"
a = 2
b = 3
name.a = "Jed"
name.b = "Jim"
SAY name.1 name.2 name.3 a b
Nobody Jed Jim 2 3
***
Computer Associates™
2-19
The Software That Manages eBusiness ™



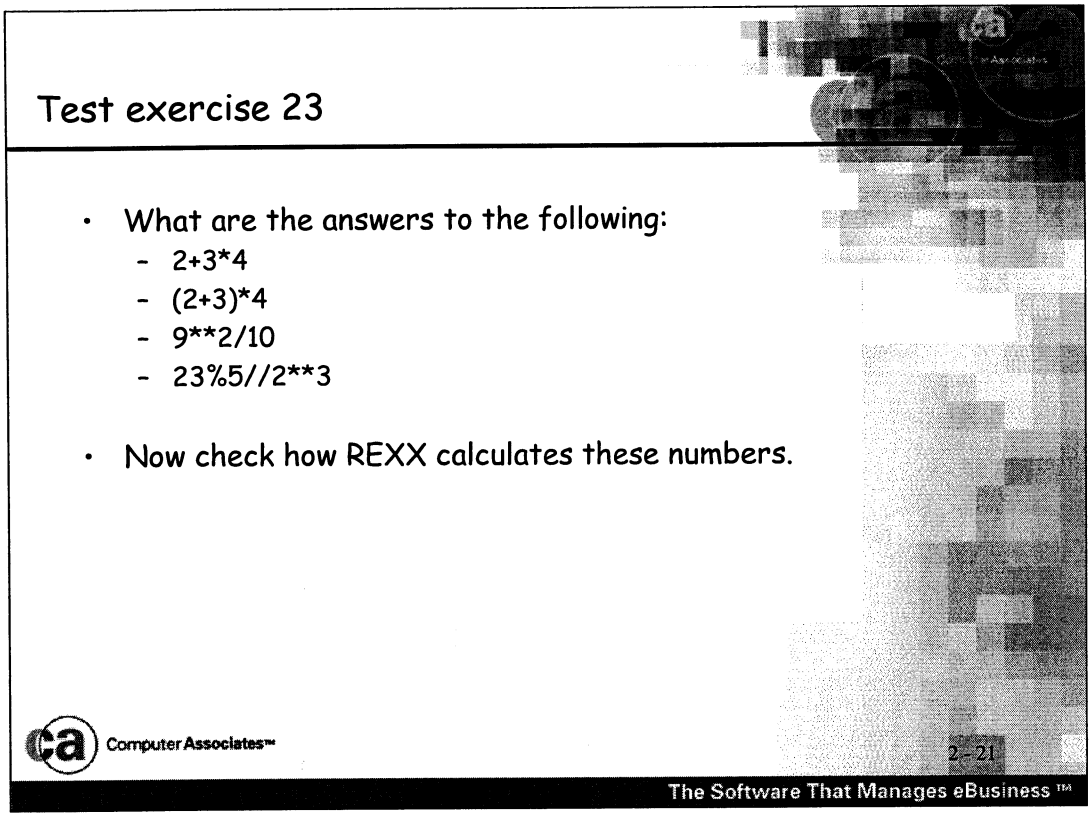
1) Expressions in parenthesis are evaluated first

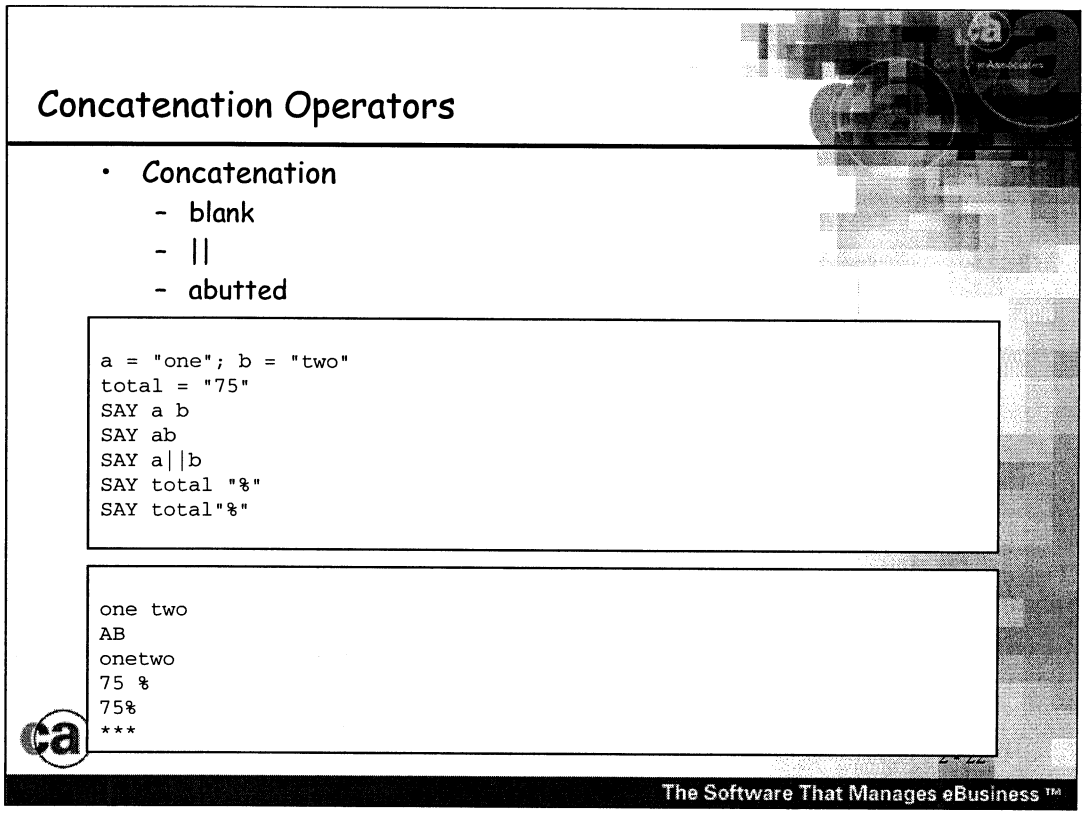
2) prefix operators - and + if present for numbers in expression

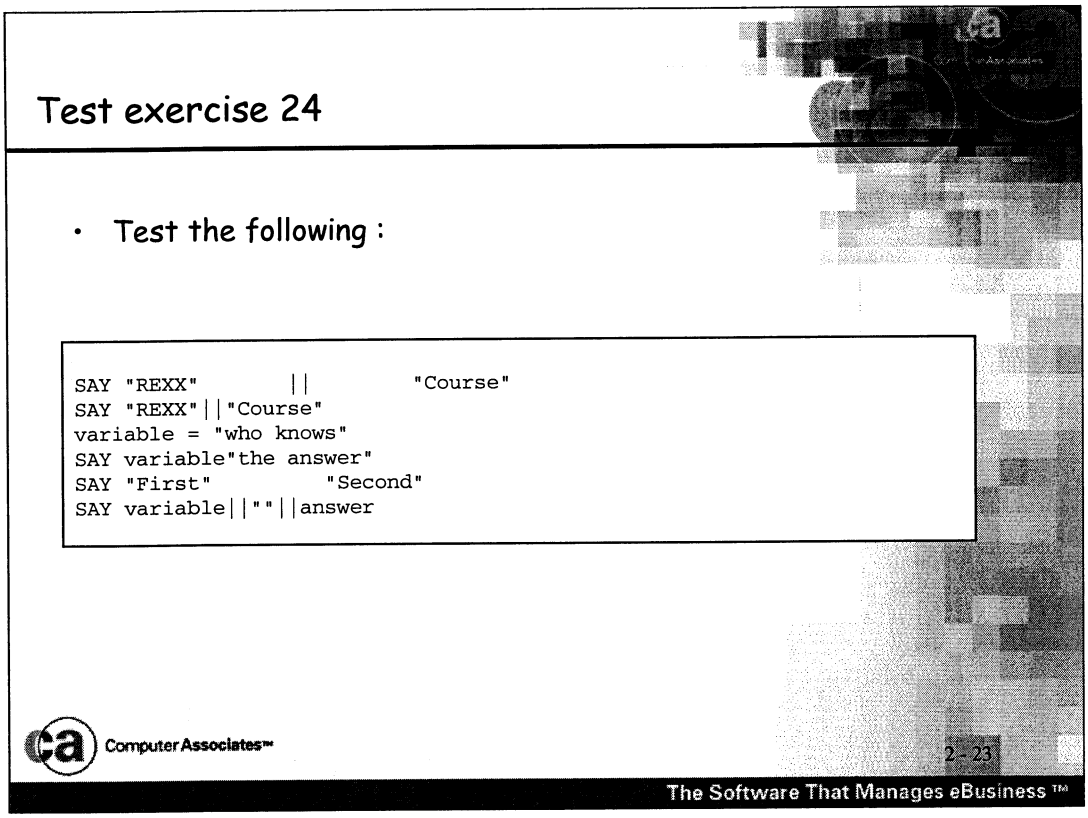
3) exponentiation **

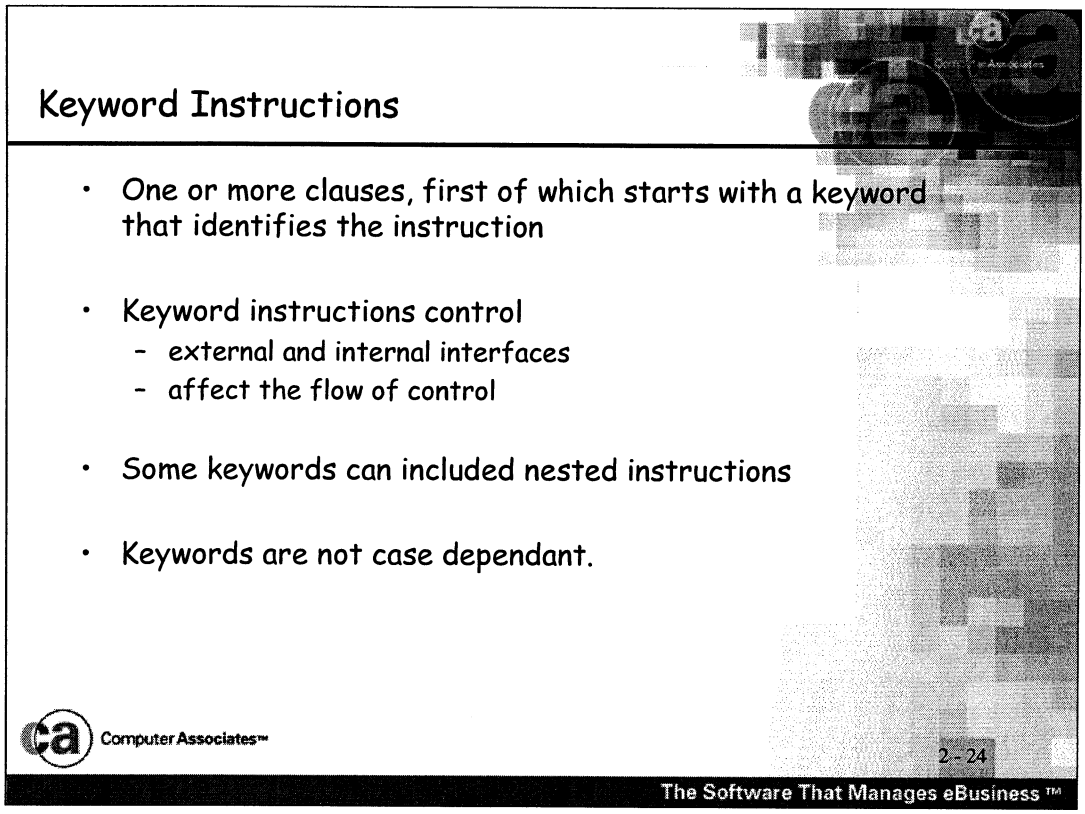
4) Multiplication and division in this order *,/,%,//

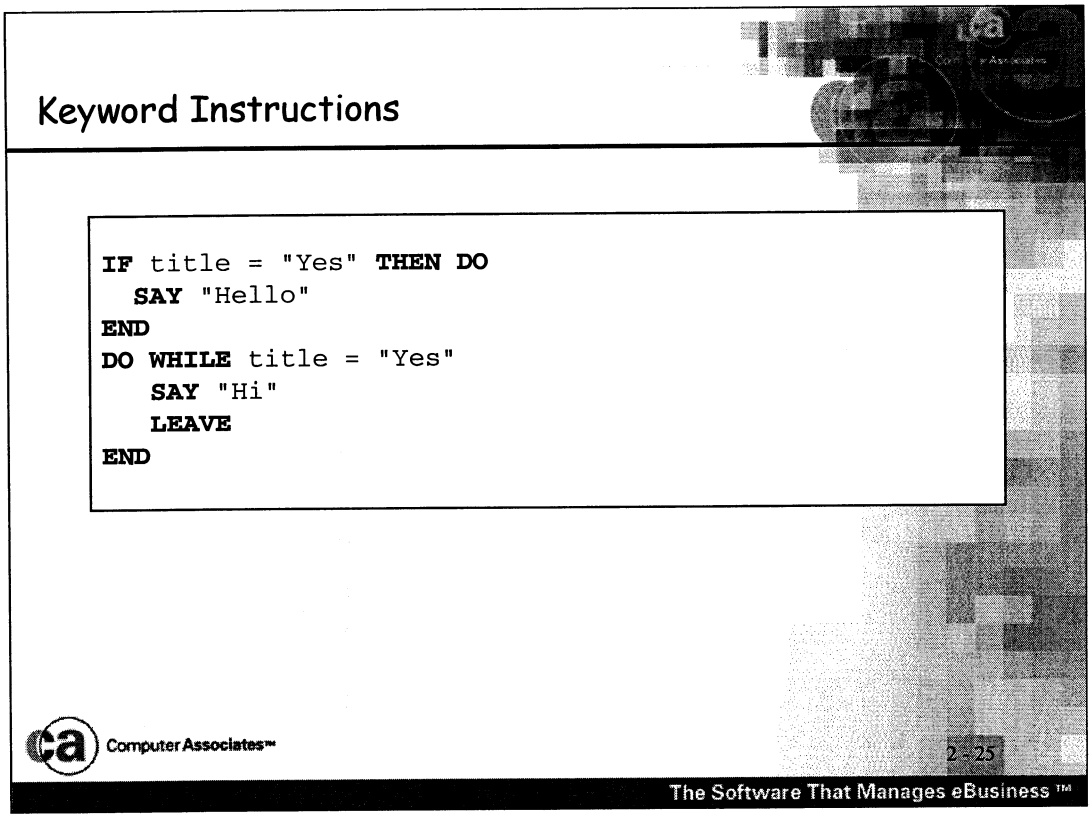
5) Addition and Subtraction + and -

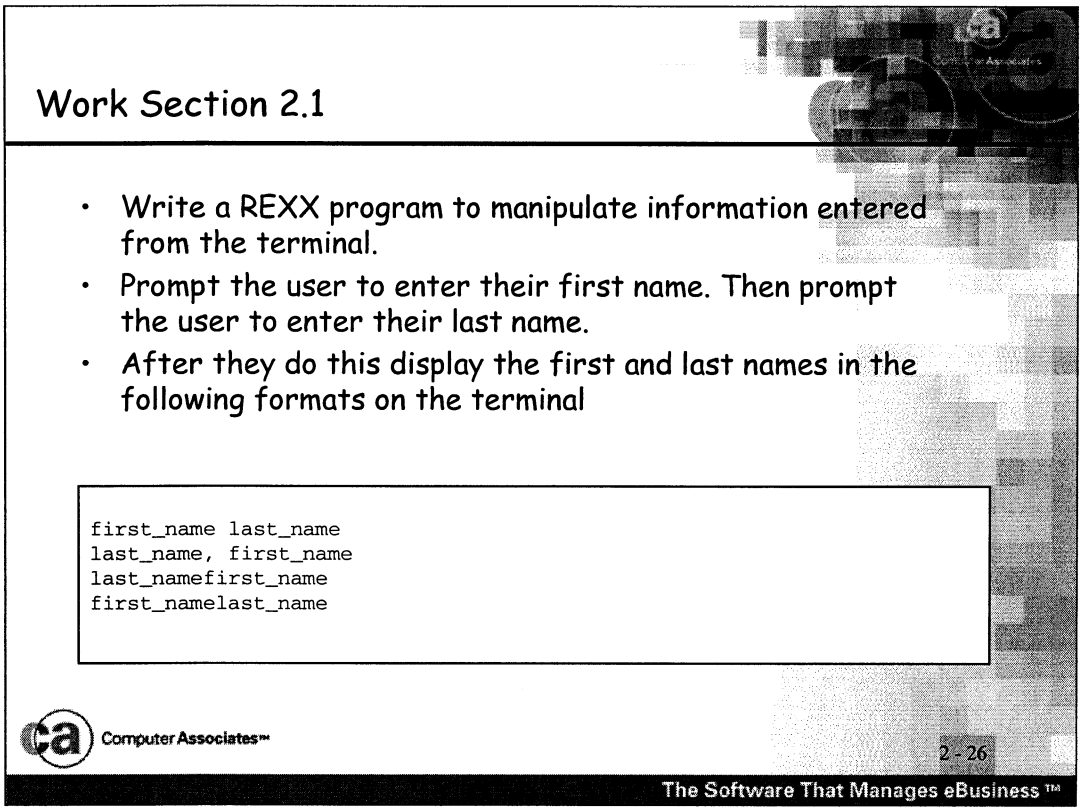

Test exercise 23
• What are the answers to the following:
- 2+3*4
- (2+3)*4
- 9**2/10
- 23%5//2**3
• Now check how REXX calculates these numbers.
Computer Associates™
2-21
The Software That Manages eBusiness ™


Concatenation Operators
• Concatenation
- blank
- ||
- abutted
a = "one"; b = "two"
total = "75"
SAY a b
SAY ab
SAY a||b
SAY total "%"
SAY total"%"
one two
AB
onetwo
75 %
75%
***
The Software That Manages eBusiness ™


Test exercise 24
• Test the following :
SAY "REXX" || "Course"
SAY "REXX"||"Course"
variable = "who knows"
SAY variable"the answer"
SAY "First" "Second"
SAY variable||""||answer
Computer Associates™
2-23
The Software That Manages eBusiness ™


Keyword Instructions
• One or more clauses, first of which starts with a keyword that identifies the instruction
• Keyword instructions control
- external and internal interfaces
- affect the flow of control
• Some keywords can included nested instructions
• Keywords are not case dependant.
Computer Associates™
2-24
The Software That Manages eBusiness ™

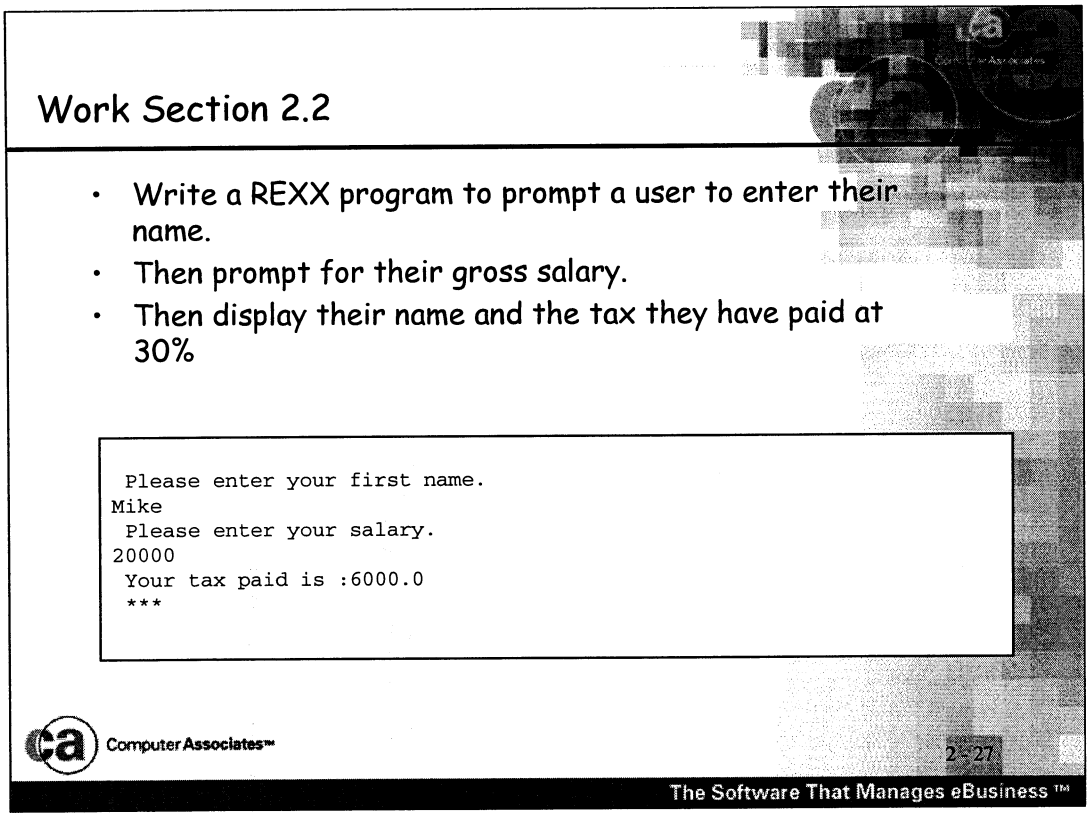

Keyword Instructions
IF title = "Yes" THEN DO
SAY "Hello"
END
DO WHILE title = "Yes"
SAY "Hi"
LEAVE
END
Computer Associates™
2-25
The Software That Manages eBusiness ™



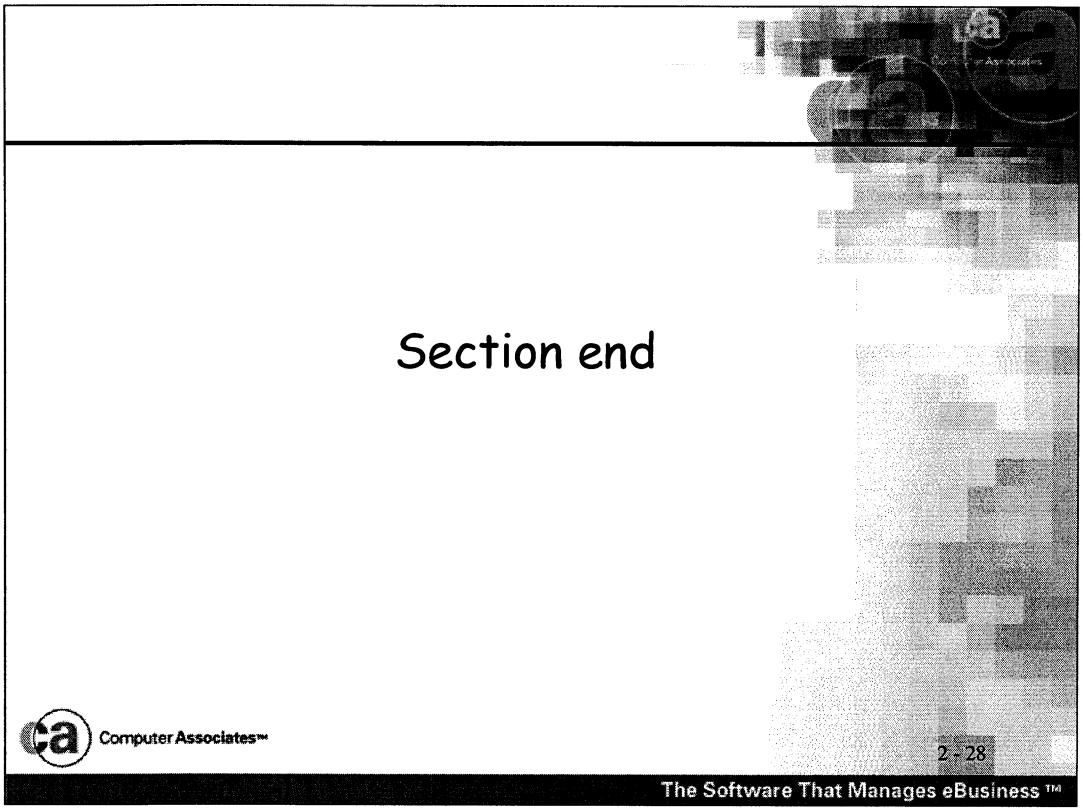
What is the result from this?

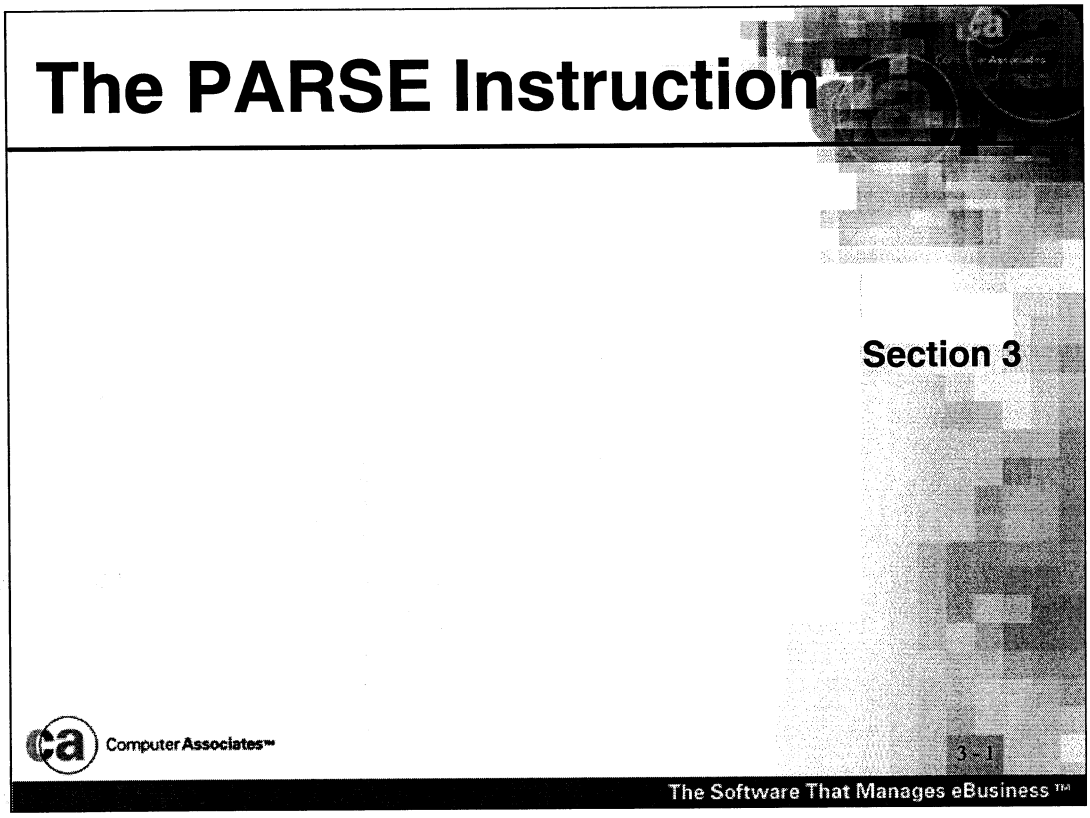
Use your dataset for all the REXX programs.

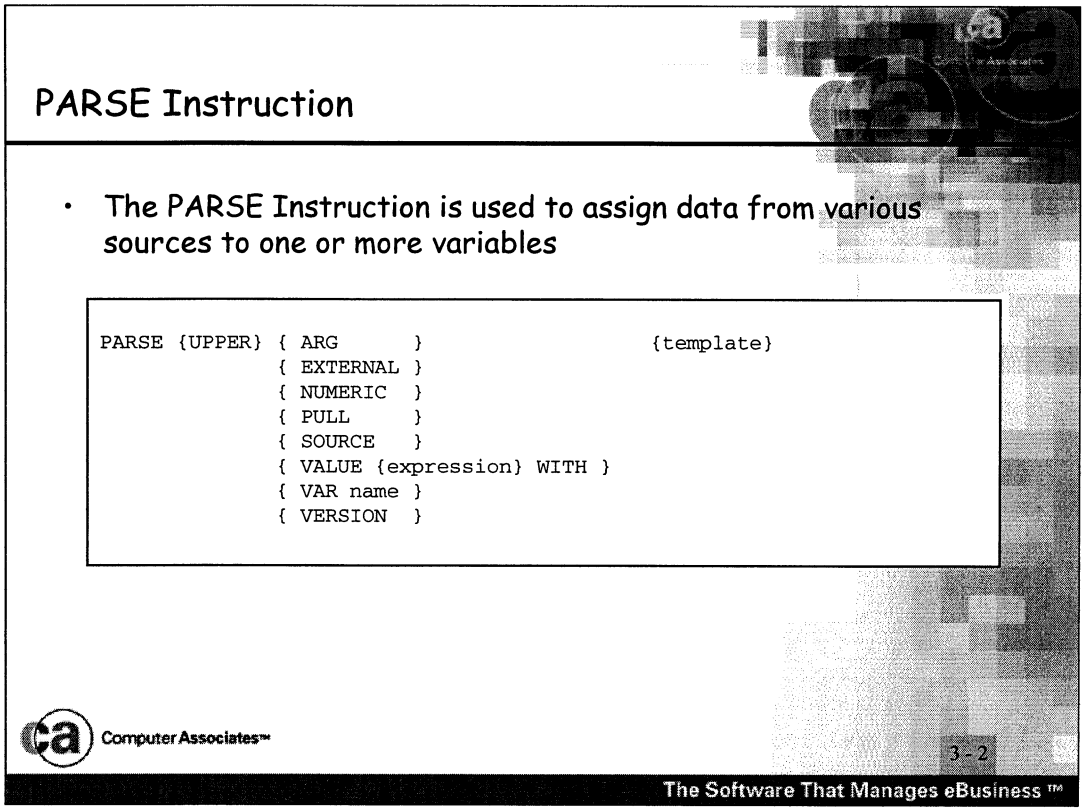
CLCS.IULCnn.REX



What is the format of the result ?


Section end
Computer Associates™
2-28
The Software That Manages eBusiness ™

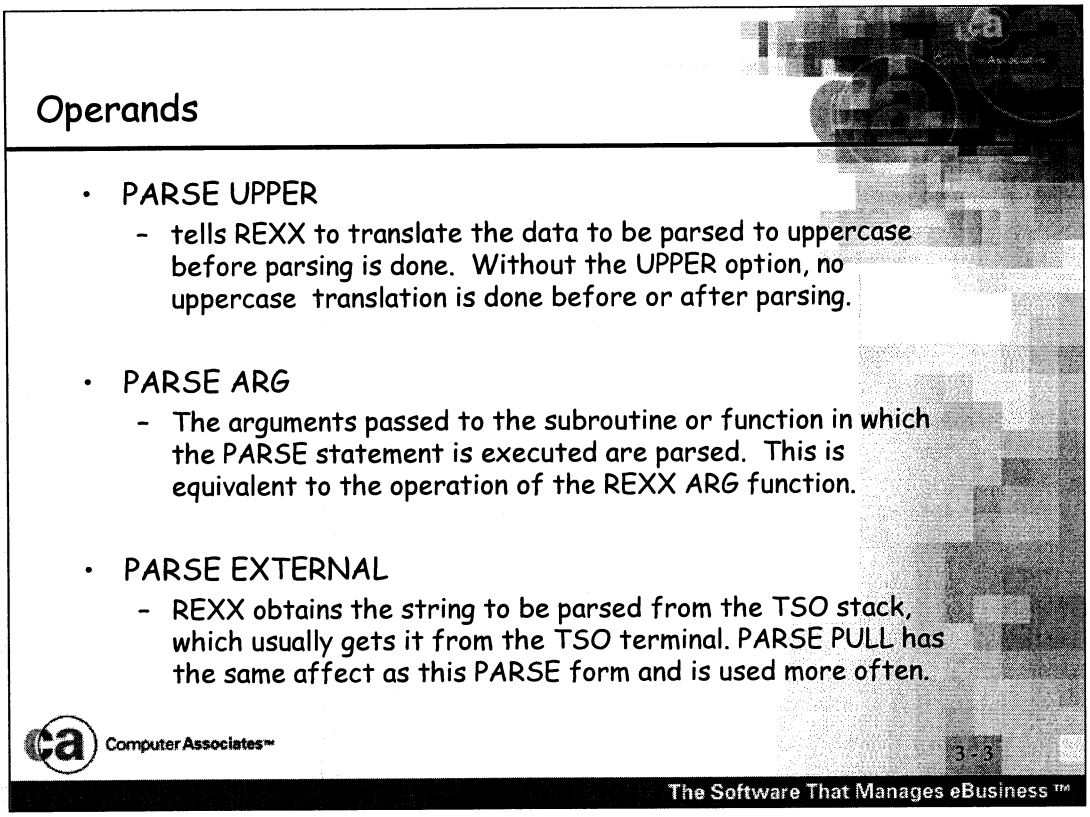

The PARSE Instruction
Section 3
Computer Associates™
3 - 1
The Software That Manages eBusiness ™

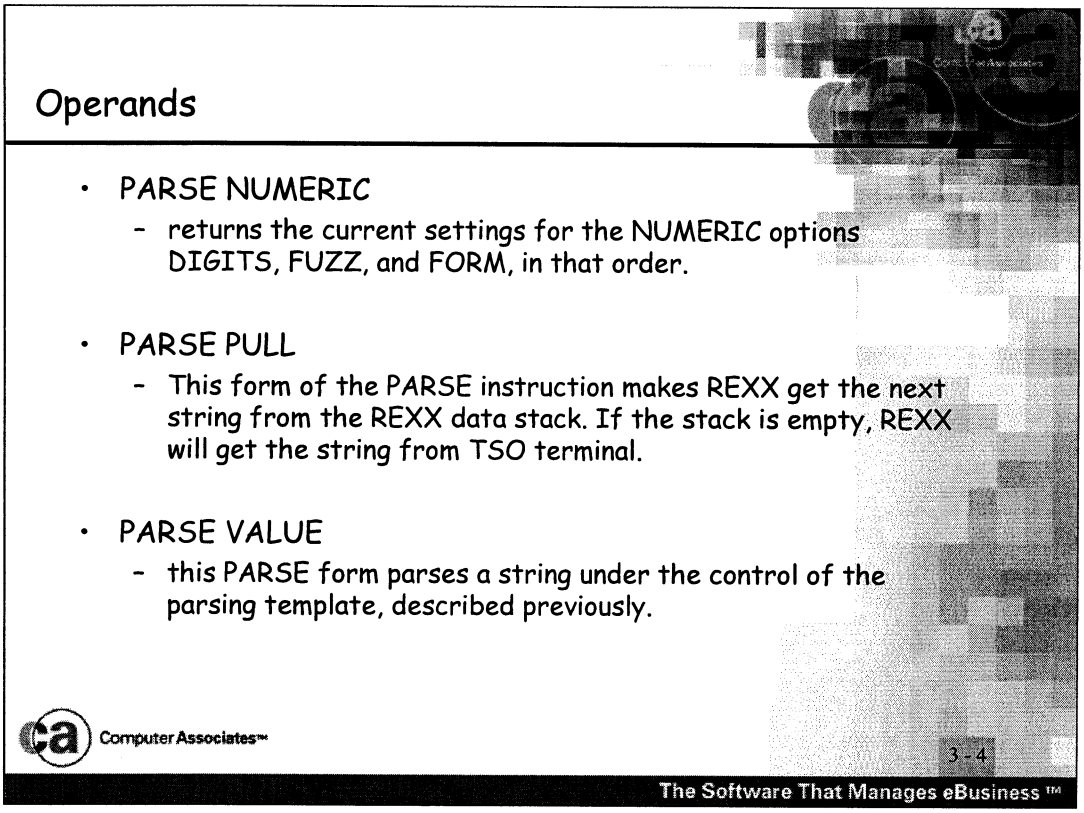


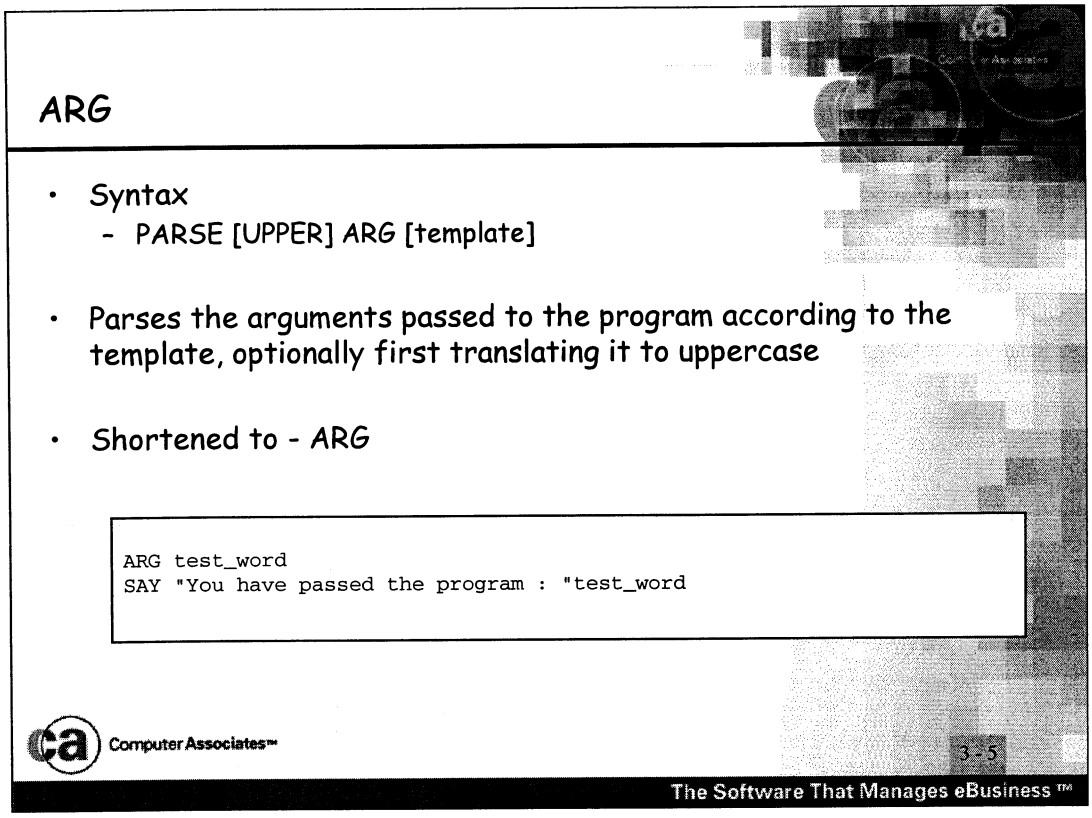
The PARSE instruction tells REXX how to assign data to one or more variables. The data to assign can be from the terminal, the data stack, or from arguments passed to a subroutine or function. The way in which REXX assigns data to a variable is governed by what is known as a 'parsing template', discussed below.

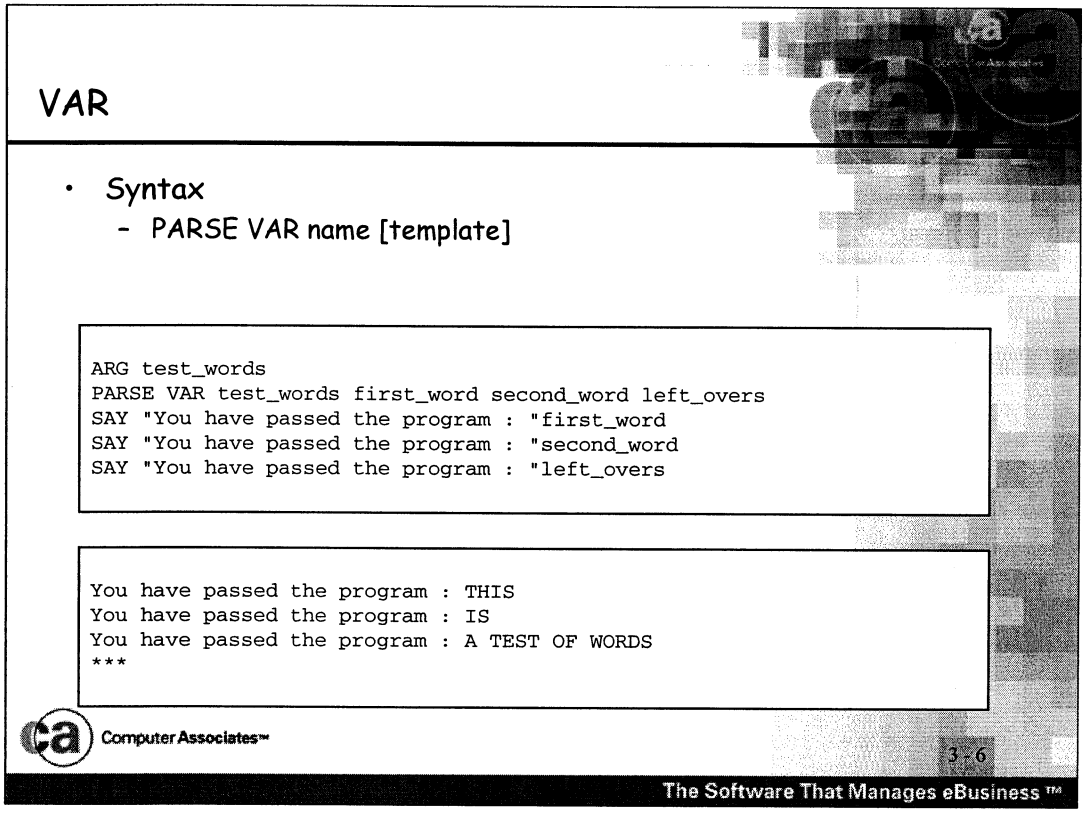
'template' is made up of alternating optional "patterns" and variable names. "patterns" are of two types: those that cause parsing to search for a matching string (variable patterns and literal patterns) and numeric patterns that supply a string position number in the data from which parsing is to extract data. Any number of "patterns" and

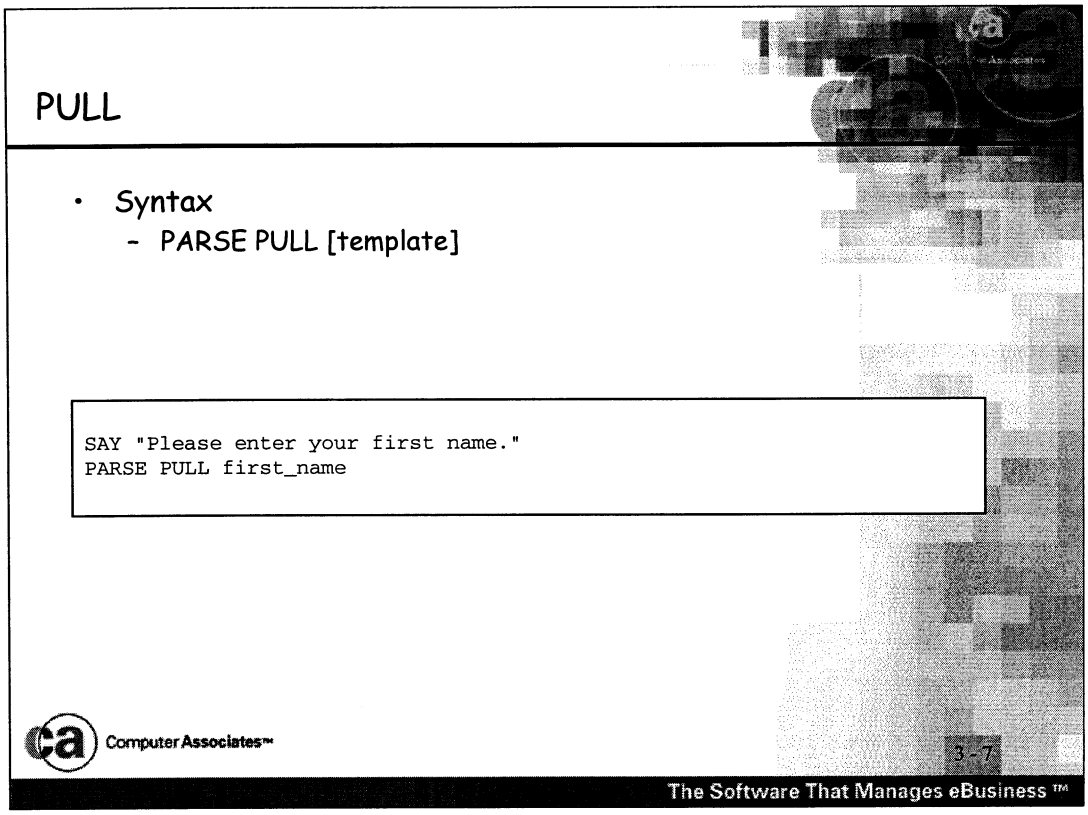
variables can be intermixed.

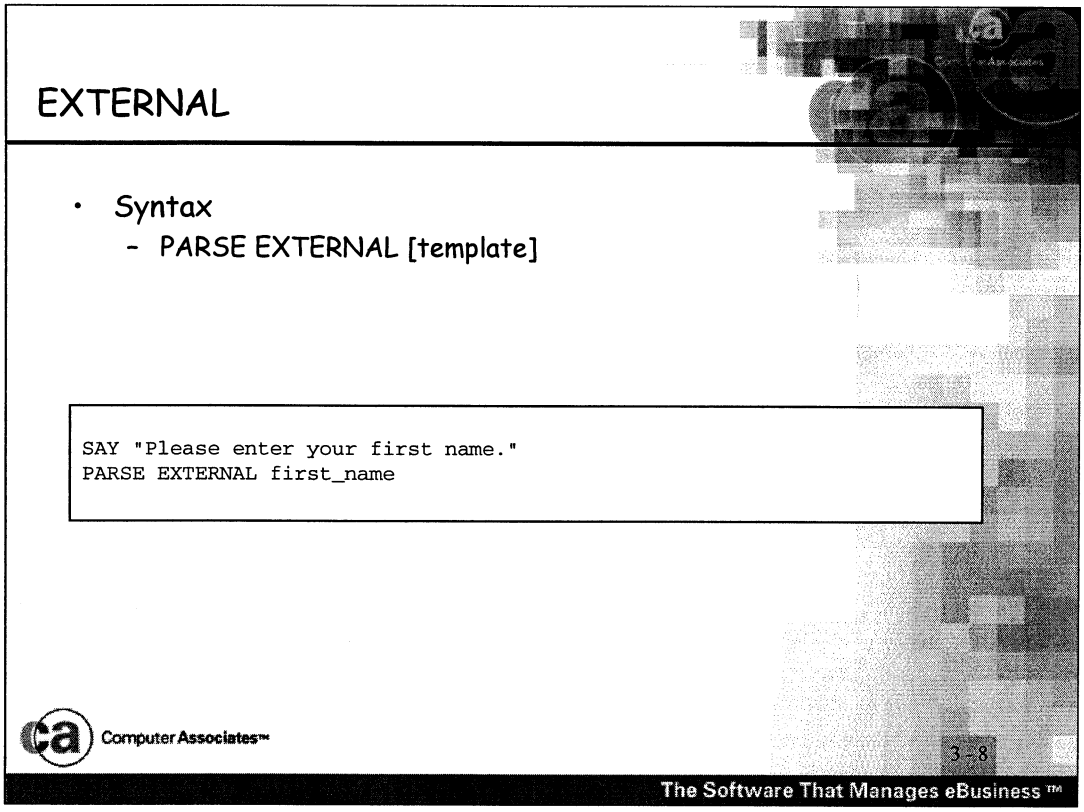

Operands
• PARSE UPPER
- tells REXX to translate the data to be parsed to uppercase before parsing is done. Without the UPPER option, no uppercase translation is done before or after parsing.
• PARSE ARG
- The arguments passed to the subroutine or function in which the PARSE statement is executed are parsed. This is equivalent to the operation of the REXX ARG function.
• PARSE EXTERNAL
- REXX obtains the string to be parsed from the TSO stack, which usually gets it from the TSO terminal. PARSE PULL has the same affect as this PARSE form and is used more often.
Computer Associates™
3-3
The Software That Manages eBusiness ™


Operands
• PARSE NUMERIC
- returns the current settings for the NUMERIC options DIGITS, FUZZ, and FORM, in that order.
• PARSE PULL
- This form of the PARSE instruction makes REXX get the next string from the REXX data stack. If the stack is empty, REXX will get the string from TSO terminal.
• PARSE VALUE
- this PARSE form parses a string under the control of the parsing template, described previously.
Computer Associates™
3 - 4
The Software That Manages eBusiness ™

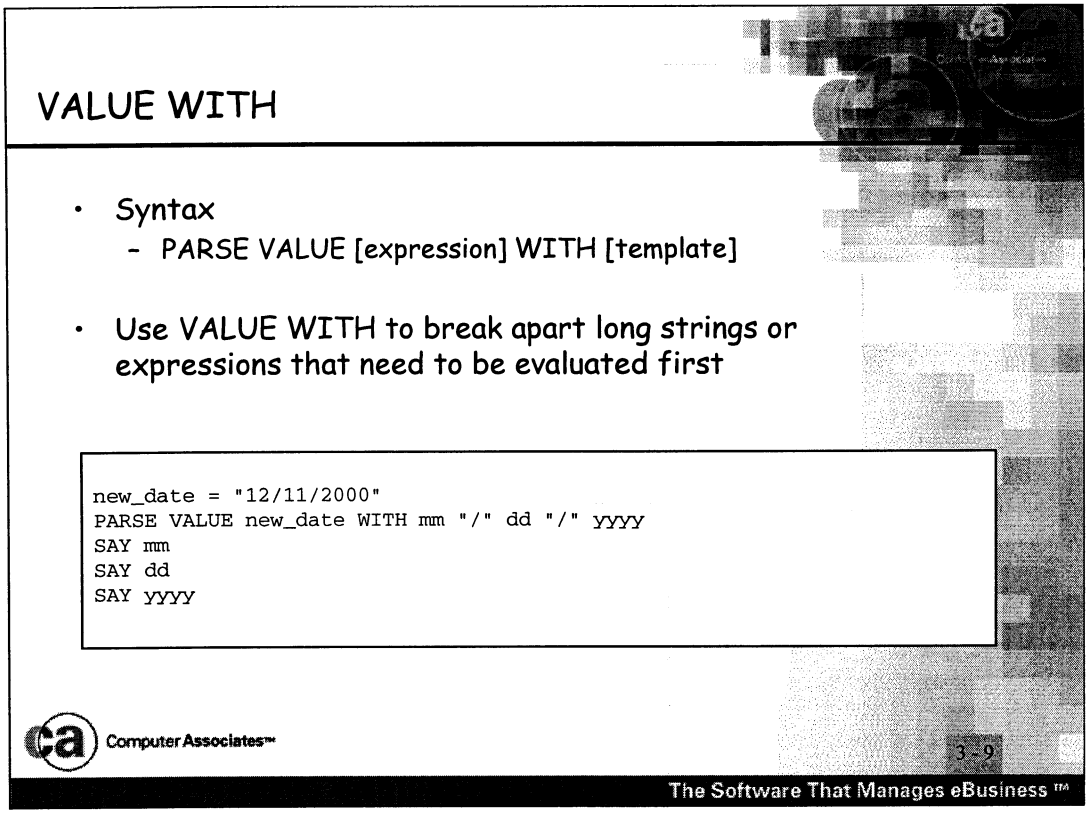

ARG
• Syntax
- PARSE [UPPER] ARG [template]
• Parses the arguments passed to the program according to the template, optionally first translating it to uppercase
• Shortened to - ARG
ARG test_word
SAY "You have passed the program : "test_word
Computer Associates™
3-5
The Software That Manages eBusiness ™

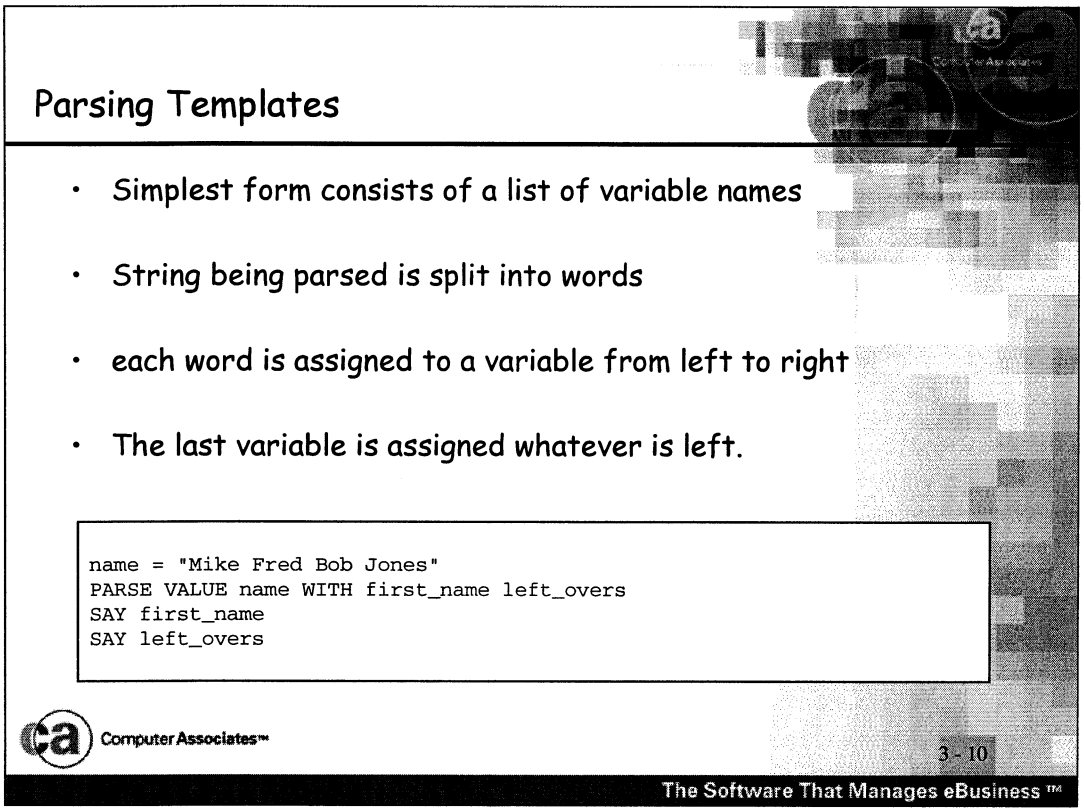

VAR
• Syntax
- PARSE VAR name [template]
ARG test_words
PARSE VAR test_words first_word second_word left_overs
SAY "You have passed the program : "first_word
SAY "You have passed the program : "second_word
SAY "You have passed the program : "left_overs
You have passed the program : THIS
You have passed the program : IS
You have passed the program : A TEST OF WORDS
***
Computer Associates™
3-6
The Software That Manages eBusiness ™

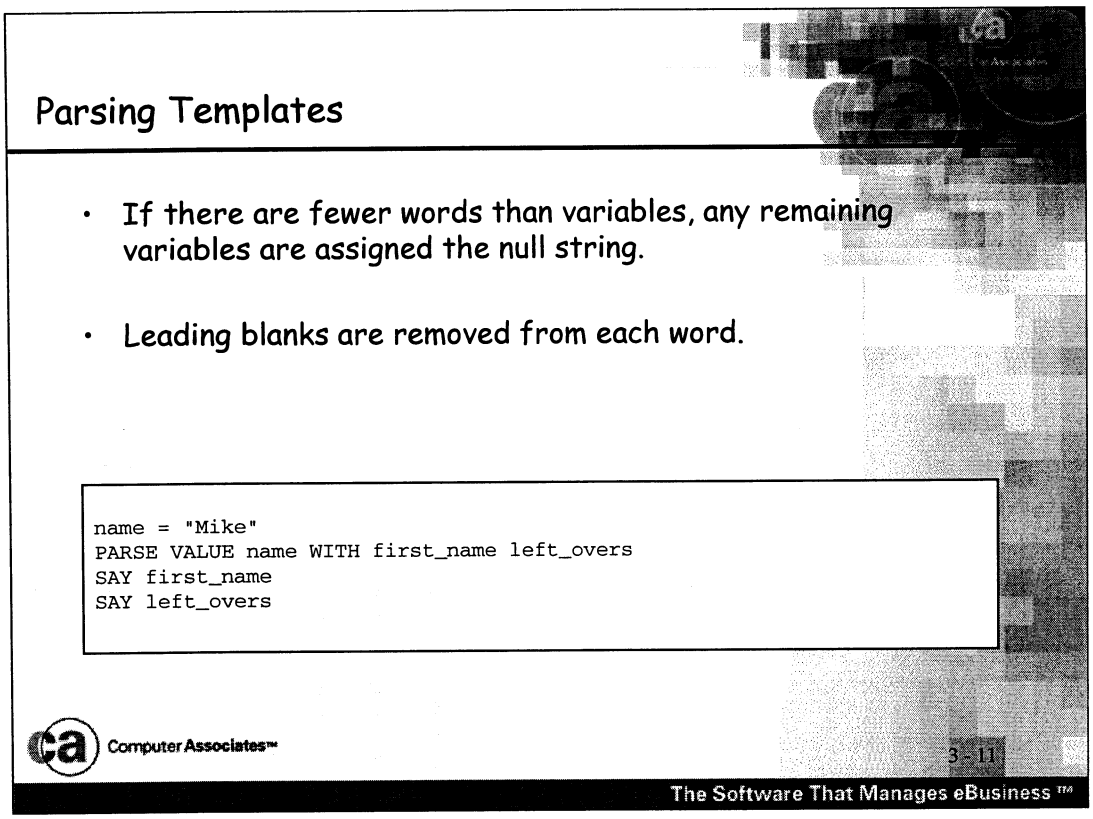

PULL
• Syntax
- PARSE PULL [template]
SAY "Please enter your first name."
PARSE PULL first_name
Computer Associates™
3 - 7
The Software That Manages eBusiness ™

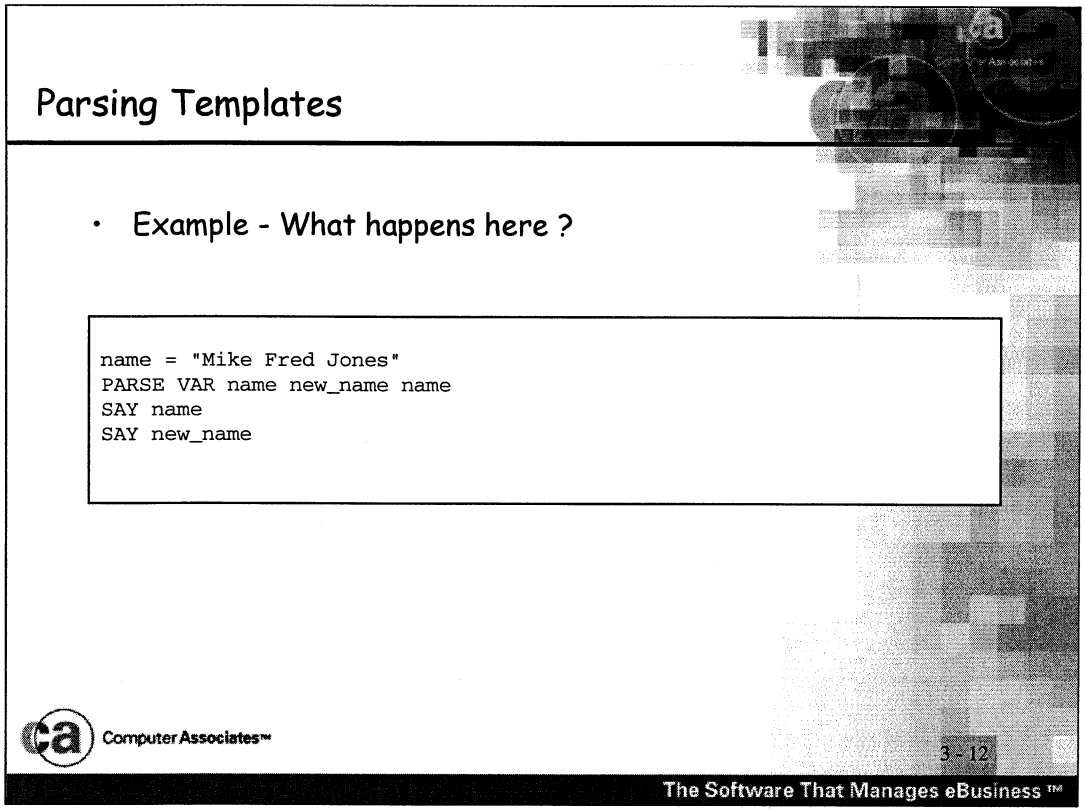


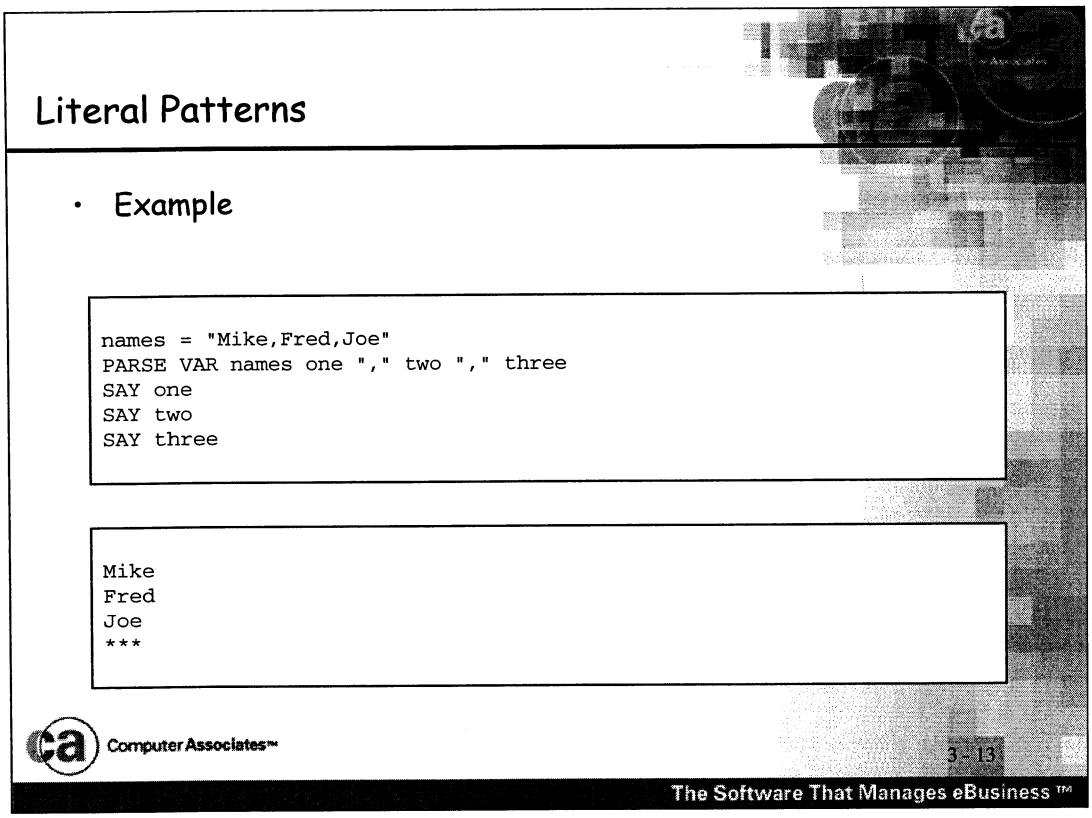
This is TSO only.

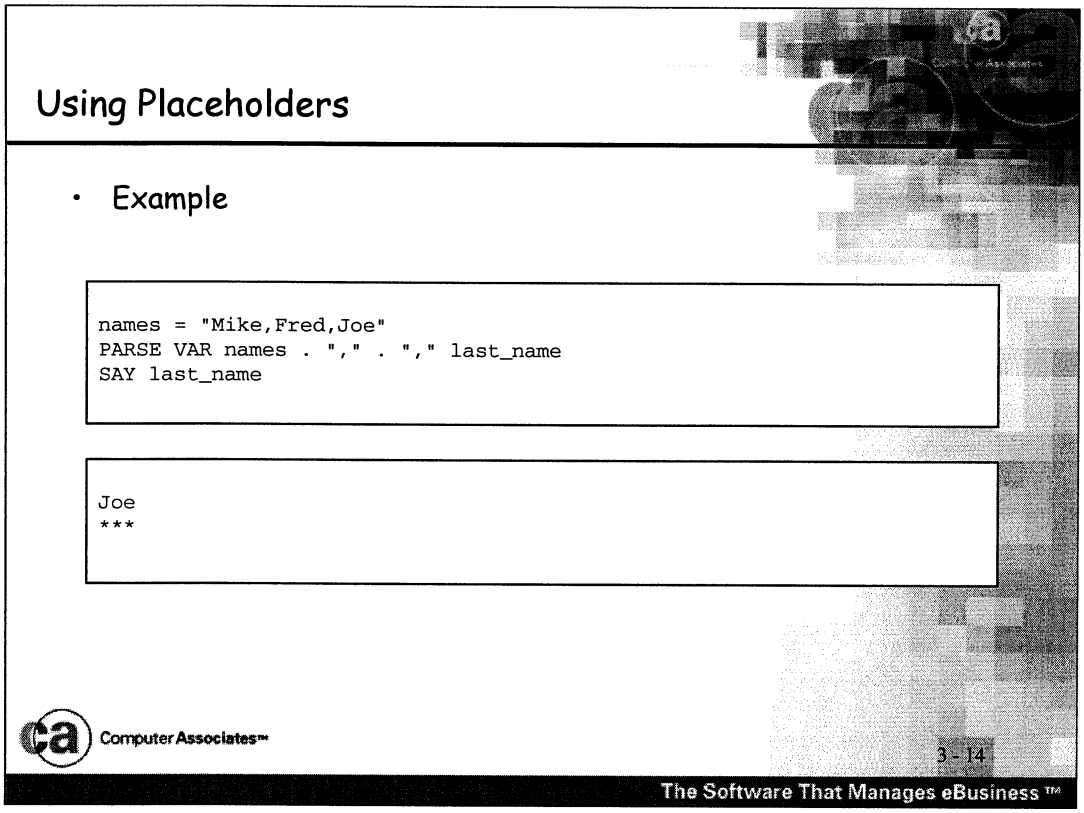

VALUE WITH
• Syntax
- PARSE VALUE [expression] WITH [template]
• Use VALUE WITH to break apart long strings or expressions that need to be evaluated first
new_date = "12/11/2000"
PARSE VALUE new_date WITH mm "/" dd "/" yyyy
SAY mm
SAY dd
SAY yyyy
Computer Associates™
3 - 9
The Software That Manages eBusiness ™

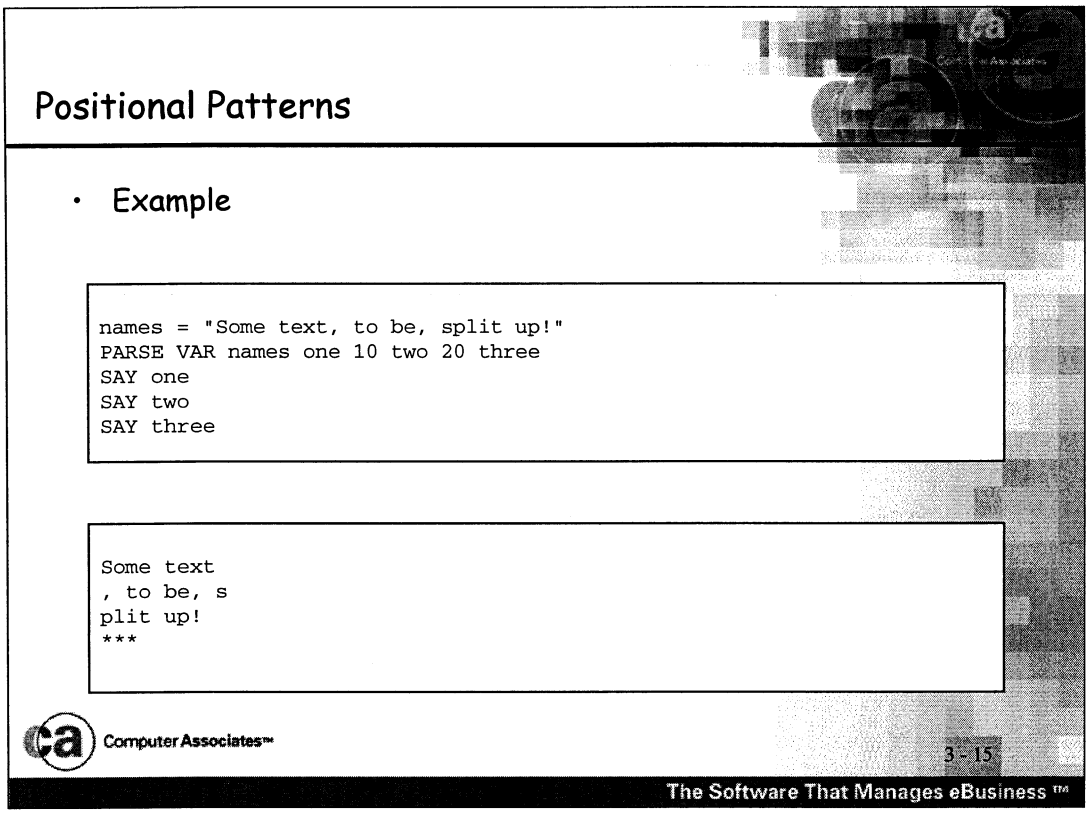

Parsing Templates
• Simplest form consists of a list of variable names
• String being parsed is split into words
• each word is assigned to a variable from left to right
• The last variable is assigned whatever is left.
name = "Mike Fred Bob Jones"
PARSE VALUE name WITH first_name left_overs
SAY first_name
SAY left_overs
Computer Associates™
3 - 10
The Software That Manages eBusiness ™

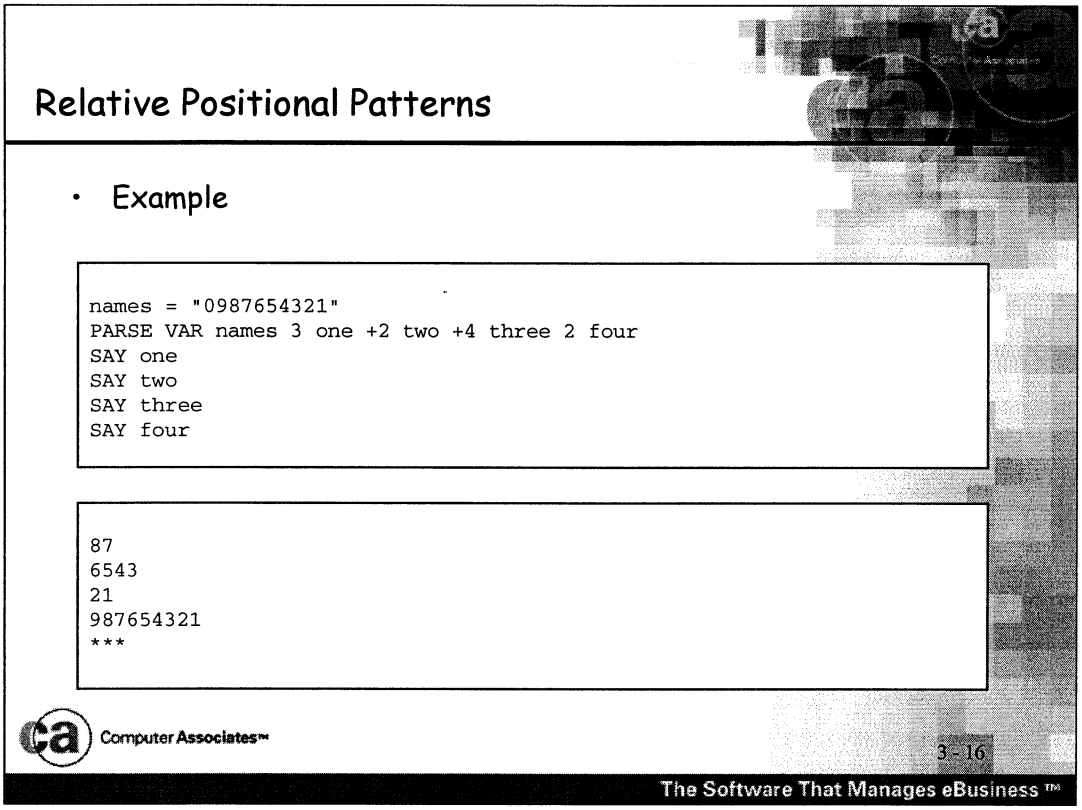

Parsing Templates
• If there are fewer words than variables, any remaining variables are assigned the null string.
• Leading blanks are removed from each word.
name = "Mike"
PARSE VALUE name WITH first_name left_overs
SAY first_name
SAY left_overs
Computer Associates™
3-11
The Software That Manages eBusiness ™

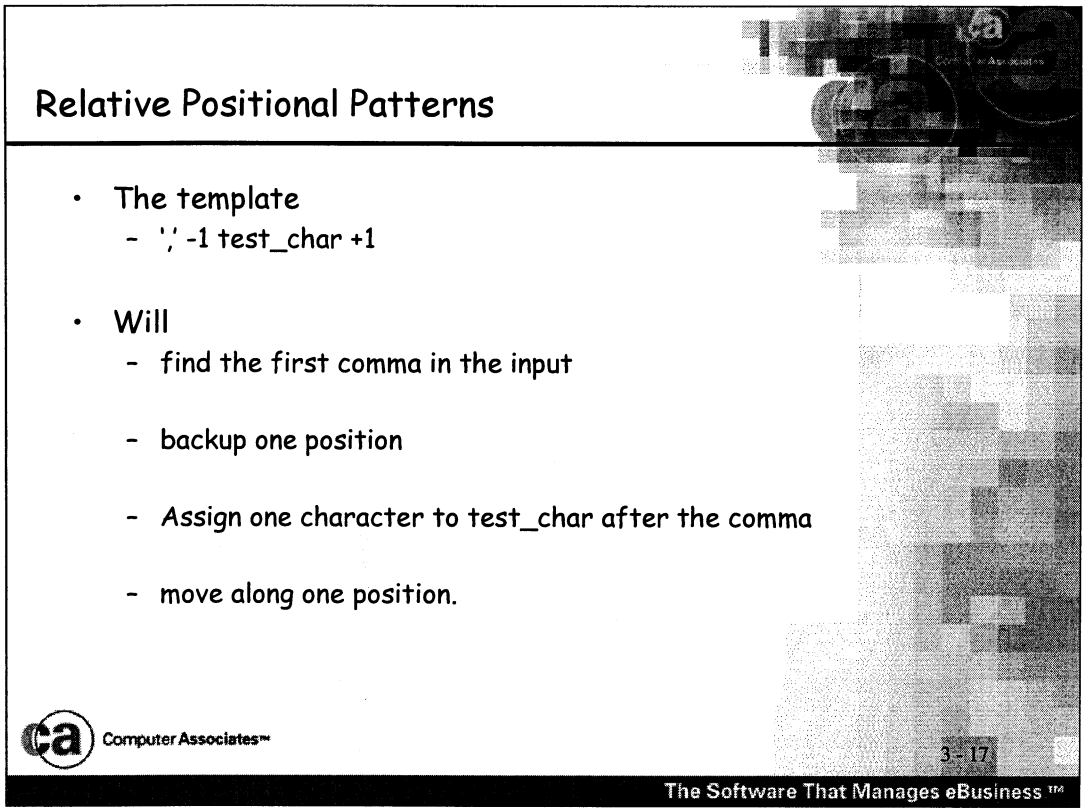

Parsing Templates
• Example - What happens here ?
name = "Mike Fred Jones"
PARSE VAR name new_name name
SAY name
SAY new_name
Computer Associates™
3 - 12
The Software That Manages eBusiness ™

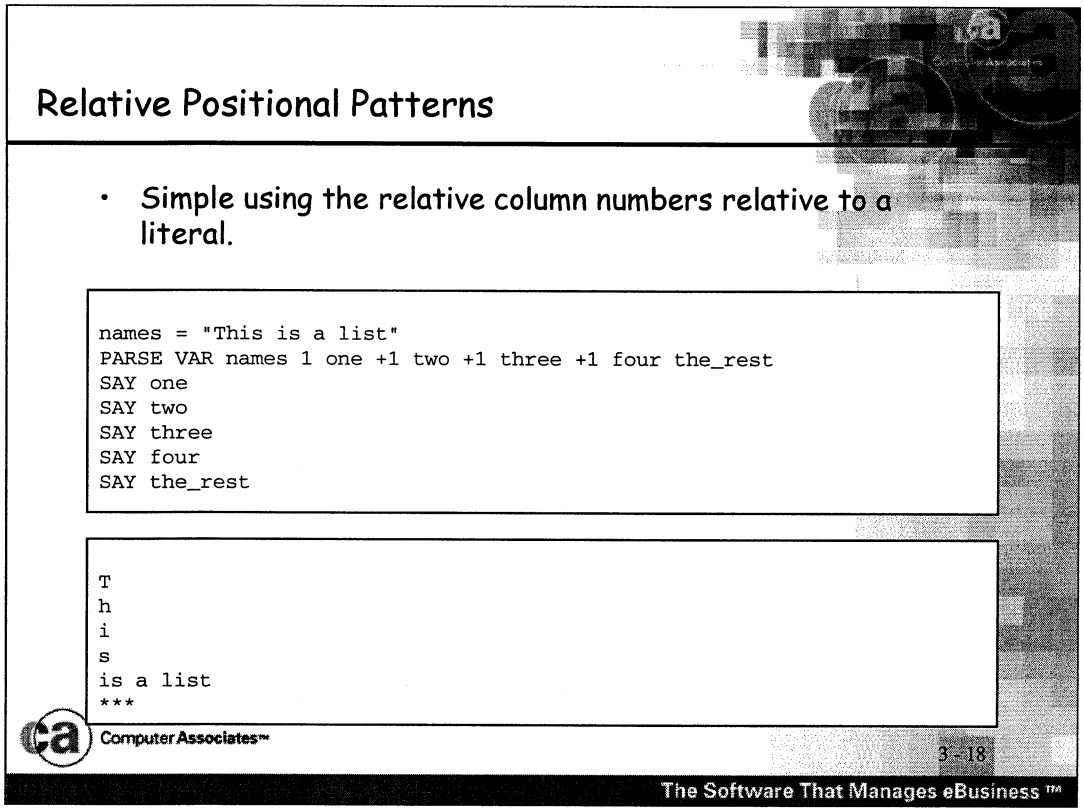

Literal Patterns
• Example
names = "Mike,Fred,Joe"
PARSE VAR names one "," two "," three
SAY one
SAY two
SAY three
Mike
Fred
Joe
***
Computer Associates™
3 - 13
The Software That Manages eBusiness ™

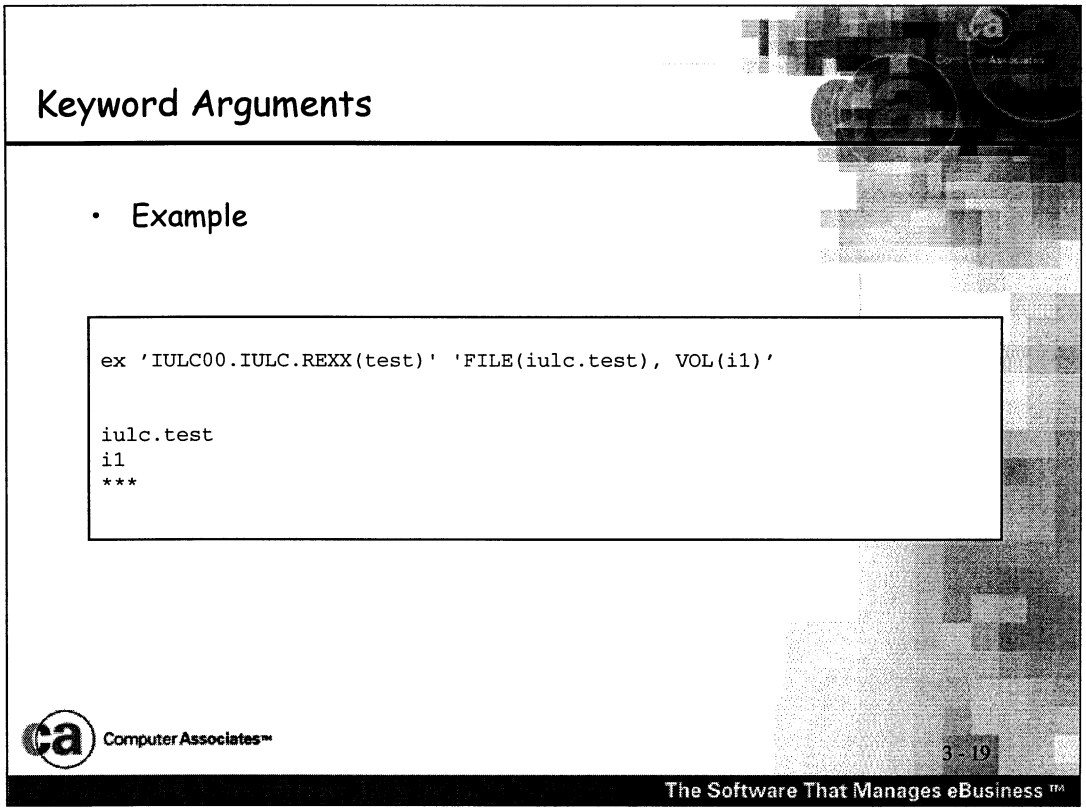

Using Placeholders
• Example
names = "Mike,Fred,Joe"
PARSE VAR names . "," . "," last_name
SAY last_name
Joe
***
Computer Associates™
3 - 14
The Software That Manages eBusiness ™


Positional Patterns
• Example
names = "Some text, to be, split up!"
PARSE VAR names one 10 two 20 three
SAY one
SAY two
SAY three
Some text
, to be, s
plit up!
***
Computer Associates™
3 - 15
The Software That Manages eBusiness ™

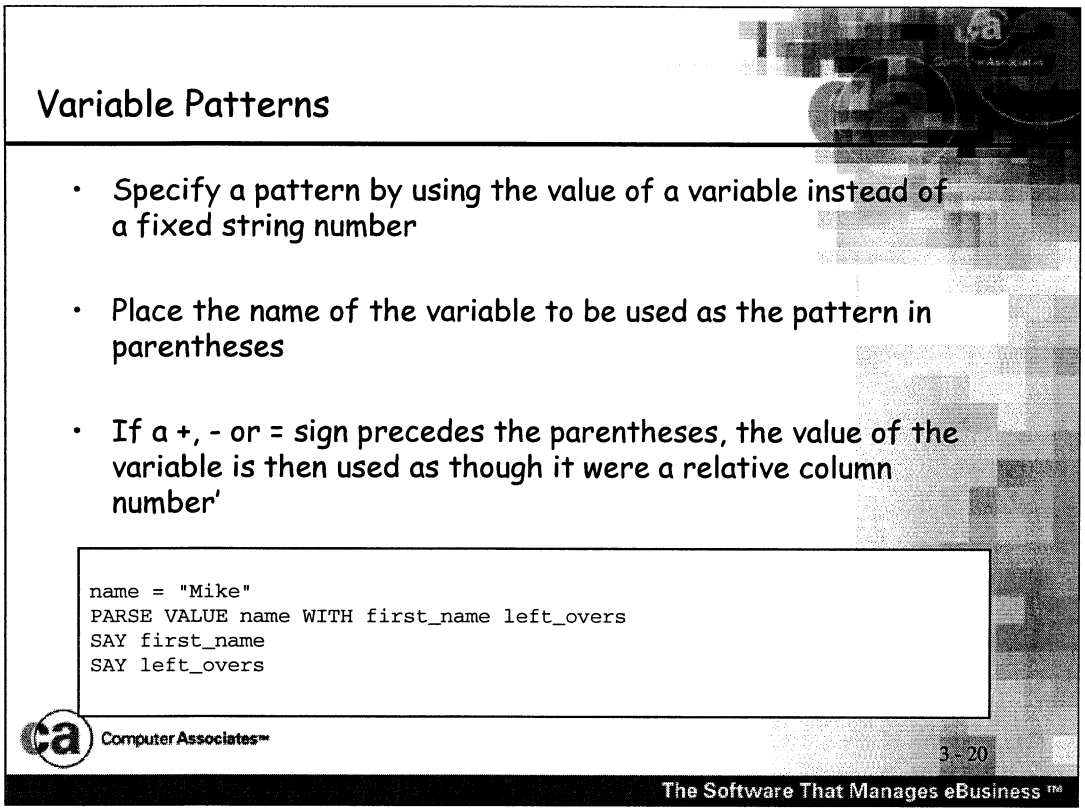

Relative Positional Patterns
• Example
names = "0987654321"
PARSE VAR names 3 one +2 two +4 three 2 four
SAY one
SAY two
SAY three
SAY four
87
6543
21
987654321
***
Computer Associates™
3-16
The Software That Manages eBusiness ™

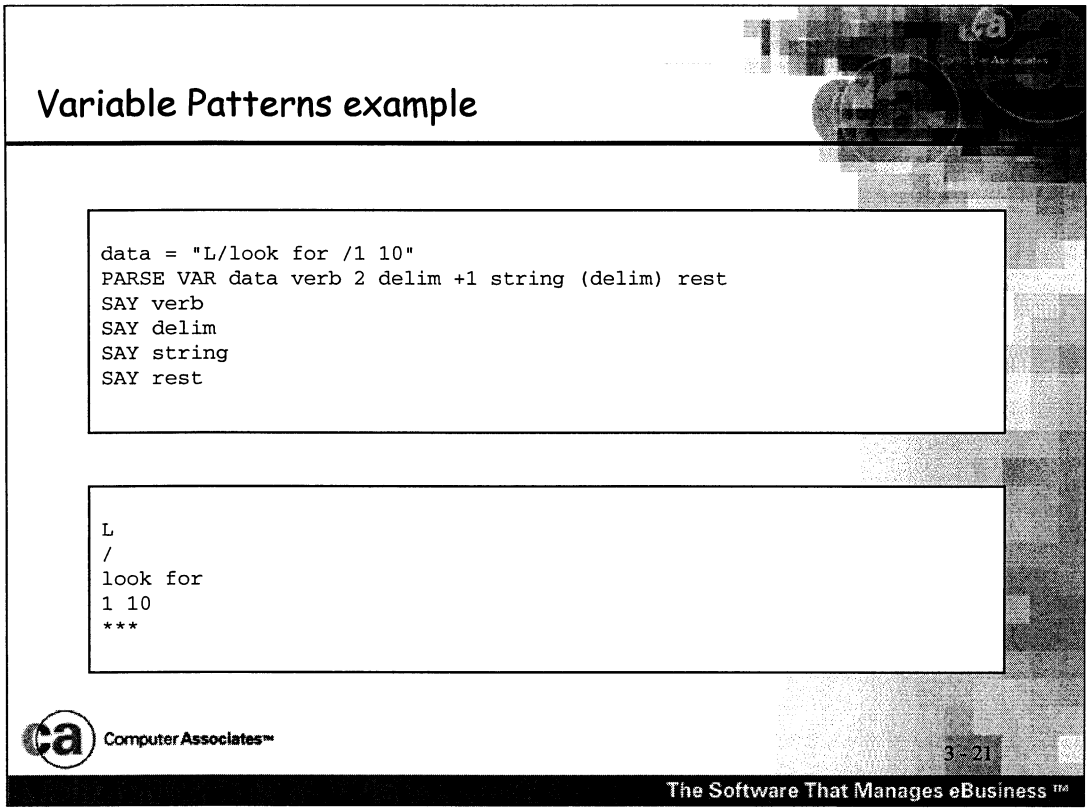

Relative Positional Patterns
• The template
- ',' -1 test_char +1
• Will
- find the first comma in the input
- backup one position
- Assign one character to test_char after the comma
- move along one position.
Computer Associates™
3-17
The Software That Manages eBusiness ™

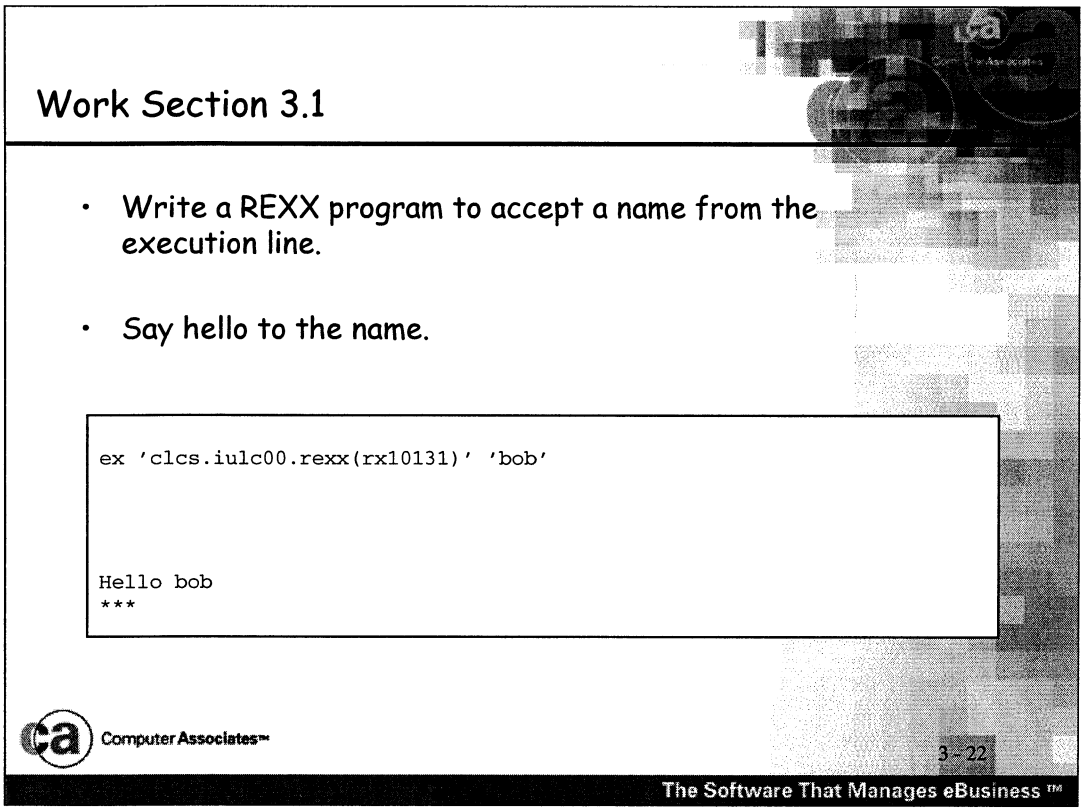

Relative Positional Patterns
• Simple using the relative column numbers relative to a literal.
names = "This is a list"
PARSE VAR names 1 one +1 two +1 three +1 four the_rest
SAY one
SAY two
SAY three
SAY four
SAY the_rest
T
h
i
s
is a list
***
Computer Associates™
3 - 18
The Software That Manages eBusiness ™

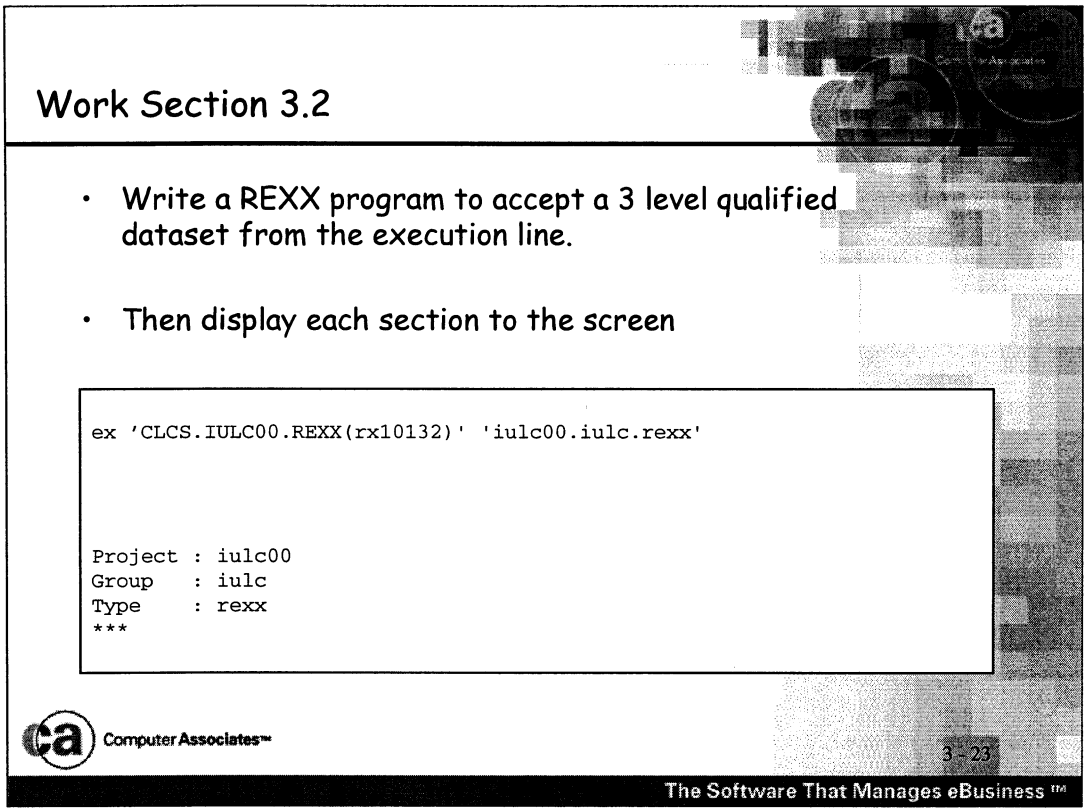


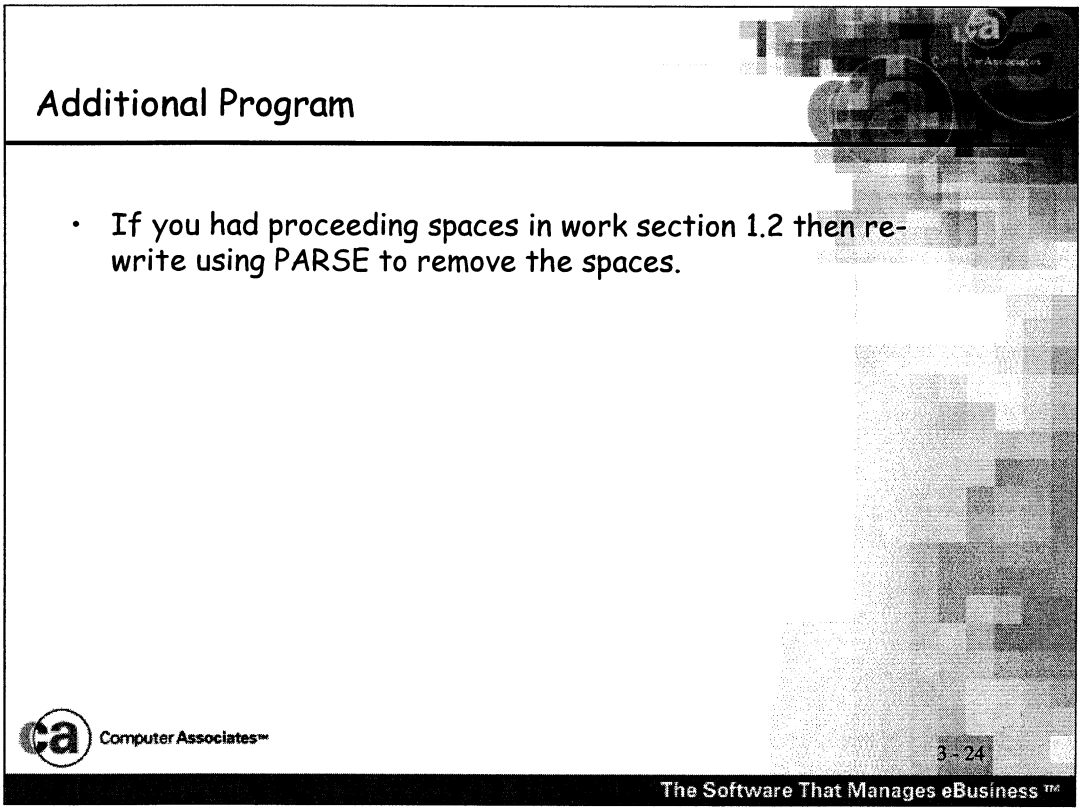

```
ARG data_list
PARSE VAR data_list "FILE(" data_set "), " "VOL(" volume ")"
SAY data_set
SAY volume
```


Variable Patterns
• Specify a pattern by using the value of a variable instead of a fixed string number
• Place the name of the variable to be used as the pattern in parentheses
• If a +, - or = sign precedes the parentheses, the value of the variable is then used as though it were a relative column number'
name = "Mike"
PARSE VALUE name WITH first_name left_overs
SAY first_name
SAY left_overs
Computer Associates™
3 - 20
The Software That Manages eBusiness ™


Variable Patterns example
data = "L/look for /1 10"
PARSE VAR data verb 2 delim +1 string (delim) rest
SAY verb
SAY delim
SAY string
SAY rest
L
/
look for
1 10
***
Computer Associates™
3 - 21
The Software That Manages eBusiness ™


Work Section 3.1
• Write a REXX program to accept a name from the execution line.
• Say hello to the name.
ex 'clcs.iulc00.rexx(rx10131)' 'bob'
Hello bob
***
Computer Associates™
3-22
The Software That Manages eBusiness ™


Work Section 3.2
• Write a REXX program to accept a 3 level qualified dataset from the execution line.
• Then display each section to the screen
ex 'CLCS.IULC00.REXX(rx10132)' 'iulc00.iulc.rexx'
Project : iulc00
Group : iulc
Type : rexx
***
Computer Associates™
3 - 23
The Software That Manages eBusiness ™


Additional Program
• If you had proceeding spaces in work section 1.2 then re-write using PARSE to remove the spaces.
Computer Associates™
3 - 24
The Software That Manages eBusiness ™