TSO REXX Basic course

This course has been prepared by Milos Forman for MCoE needs only!

TSO REXX

- **TSO REXX Programming**
  - 5 days
- **Audience**
  - The course was prepared for MCOE new hires
- **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 utilizing the features available under TSO.
    To be able to use REXX in various environments and with various API's

Presentations contain extractions from a relevant IBM documentation. Especially syntax diagrams and definitions. These extractions are used to focus students to mentioned documents. Students are asked to study particular topics from the documentation independently.

1) Introduction and Concepts

1) Introduction + Concepts:
Course objectives.
Documentation and resources.
Concepts:
How, why and where REXX.
TSO command ALTLIB, EXEC...
REXX instruction format.
Instructions: SAY, PULL...

2) Basic components:
Clauses, Statements, Comments, Literal strings, Numbers, Symbols, Variables, Labels...
Arithmetic and Concatenation operators.
Instructions: Assignment, DROP, UPPER, NUMERIC...

3) Parsing:
Instructions: PARSE, ARG...
Paterns.

4) Flow control:
Logical operators, comparative operators.
Instructions: IF/THEN/ELSE, SELECT, NOP...

Introduction

5) Debugging and error trapping:
Instructions: SIGNAL, TRACE...
TSO Immediate commands HT, RT, HE, HI, TE, TS.

6) Looping:
Instructions: DO, LEAVE and ITERATE...

7) Subroutines and functions:
Internal, External.
Instructions: CALL, PROCEDURE. EXPOSE, RETURN, EXIT, INTERPRET...
Addressing: ADDRESS, OUTTRAP instruction.

8) Built-in functions:
DATATYPE, POS, LEFT/RIGHT, STRIP, ABBREV, FORMAT, COMPARE, arithmetic functions.
TSO Built-in functions.

Introduction

9) **Data stacks:**

QUEUE, PUSH instructions.

TSO commands and functions for work with stacks.

10) Work with data:

Instruction: EXECIO, Read/Write instruction.

TSO commands to work with datasets.

11) TSO REXX in TSO:
Panels: ISPF.
EDIT macros.
REXX in Background TSO or Non TSO address space.
REXX from JCL.
REXX in USS.

12) REXX Edit Macros (extended version):

Introduction.

Examples of ISPF Macros.

ISPF Commands.

Practice.

Introduction

13) REXX & ISPF (extended version):

Dialog.

Dialog Elements and Structure.

Starting Dialog.

Dialog Variables.

Table services.

File tailoring services.

Panel Definition.

Panel Definition Sections.

Panel Definition Procedural Sections.

Panel Definition Control variables.

14) Compiler and Library (extended version):

Compiler inputs/outputs.

Compiler invocation.

Compiler Option and Control directives.

Other considerations.

Introduction

15) REXX in Different Address Spaces (extended version):

REXX in Batch.
Invocation of REXX from Programming languages.
Manipulating with current generation of REXX variables.

16) REXX in USS (extended version):

Available environments.
SYSCALL.
Built-in Functions.

17) REXX Sockets API (extended version):

Socket concept.
Client / Server.
REXX Socket API.

Introduction - Documentation

TSO/E REXX User’s Guide
TSO/E REXX Reference

IBM Compiler and Library for REXX on zSeries Diagnosis Guide
IBM Compiler and Library for REXX on zSeries User’s Guide and Reference

Using REXX and z/OS UNIX System Services

DB2 Universal Database for z/OS Application Programming and SQL Guide

z/OS Communications Server IP Sockets Application Programming Interface Guide and Reference

REXX Samples

REXX samples: (You should first copy them under your pmfID)

MCOE.REXA.REXX	->	pmfid.REXA.REXX
MCOE.REXA.JCL	->	pmfid.REXA.JCL
MCOE.REXA.SKEL	->	pmfid.REXA.SKEL
MCOE.REXA.PANEL	->	pmfid.REXA.PANEL
/u/users/kotmi01/REXX	->	/u/users/pmfid

Concepts – Brief History

- REstructured eXtended eXecutor.
- Standard SAA procedure Language.
- User driven Development.
- Born on operating system VM/SP - current z/VM.
- Available on many platforms.
- Interpret, (compiler is available on some platforms).
- Designed to issue commands to the operating system.

Concepts – Why Program in REXX

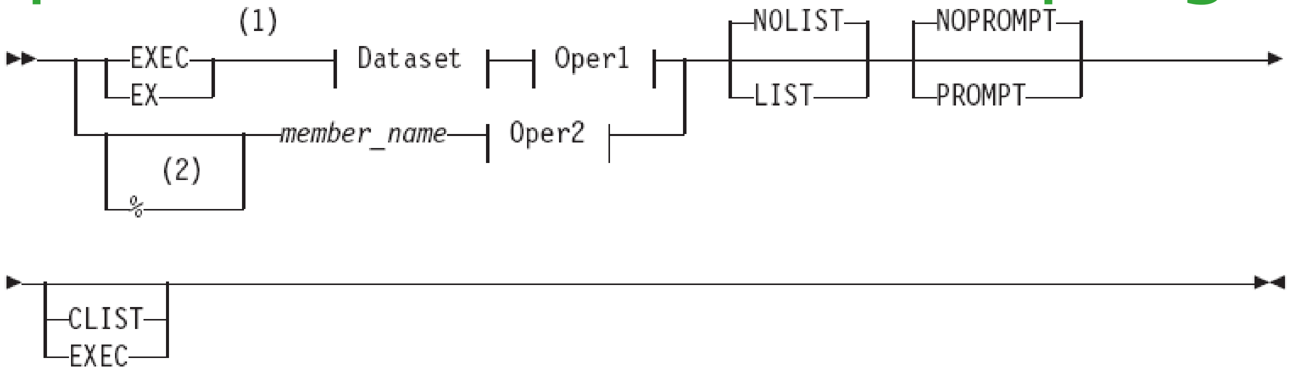
- Easy to read and write, using meaningful English words.
- Very litle punctuation.
- Free format – instruction are not column dependant and can span multiple line.
- Variable declaration is not required.
- Includes built-in functions.
- Trace capabilities.
- Extensive word and character manipulation.

Concepts – Where to use REXX

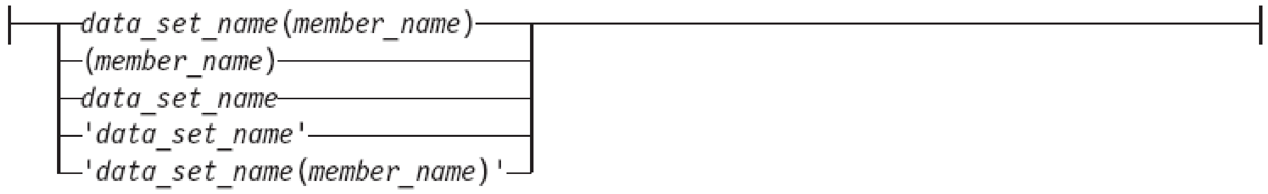
Use REXX to:

- Create easy and error free tasks.
- Automate repetitive tasks.
- Create program that behave like system commands.
- Tailor and enhance software systems.
- Create complex application.
- Ad hoc applications.

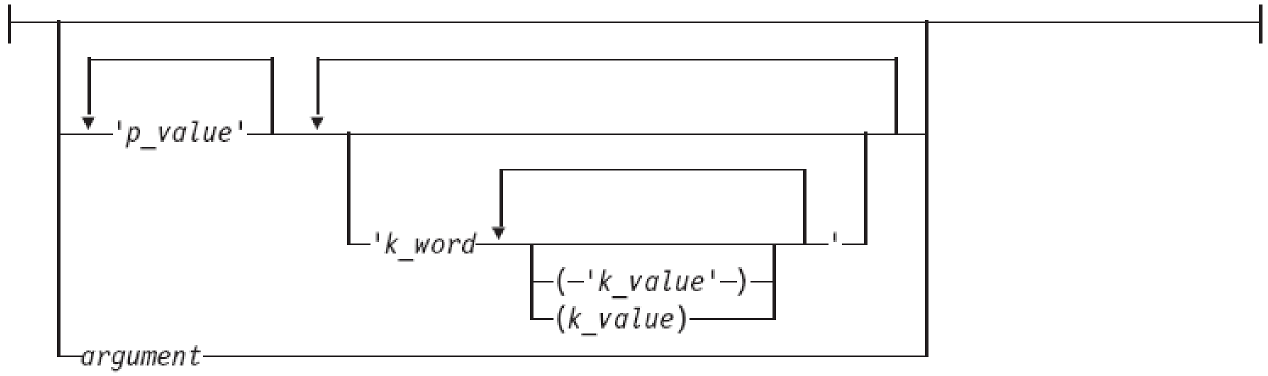
Concepts – How to run REXX program.



Dataset



Oper1



Concepts – How to run REXX programm

You can specify the EXEC command or the EXEC subcommand of EDIT and TEST in three ways:

- **Explicit form:** Enter EXEC or EX followed by the name of the data set that contains the CLIST or REXX exec. If you need prompting you should invoke EXEC explicitly with the PROMPT option.
- **Implicit form:** Do *not* enter EXEC or EX; enter only the name of the member to be found in a procedure library such as SYSEXEC or SYSPROC. A procedure library consists of partitioned data sets allocated to the specific file (SYSPROC or SYSEXEC) either dynamically by the ALLOCATE command or as part of the LOGON procedure. TSO/E determines if the member name is a system command before it searches the libraries.
- **Extended implicit form:** Enter a percent sign followed by the member name. TSO/E only searches the procedure library for the specified name. This form is faster because the system doesn't search for commands.

Some of the commands in a CLIST might have symbolic variables for operands. When you specify the EXEC command, you can supply actual values for the system to use in place of the symbolic variables. In addition, when you invoke a REXX exec you can pass arguments on the EXEC command. Specify the arguments in single quotes.

Concepts – How to run REXX program

Command Exec can be invoked:

TSO command line:

```
tso exec ‘mcoe.rexa.rexx(rx20115)’
```

From ISPF:

p=PDF
6=Command

```
exec 'mcoe.rexa.rexx(rx20115)'
```

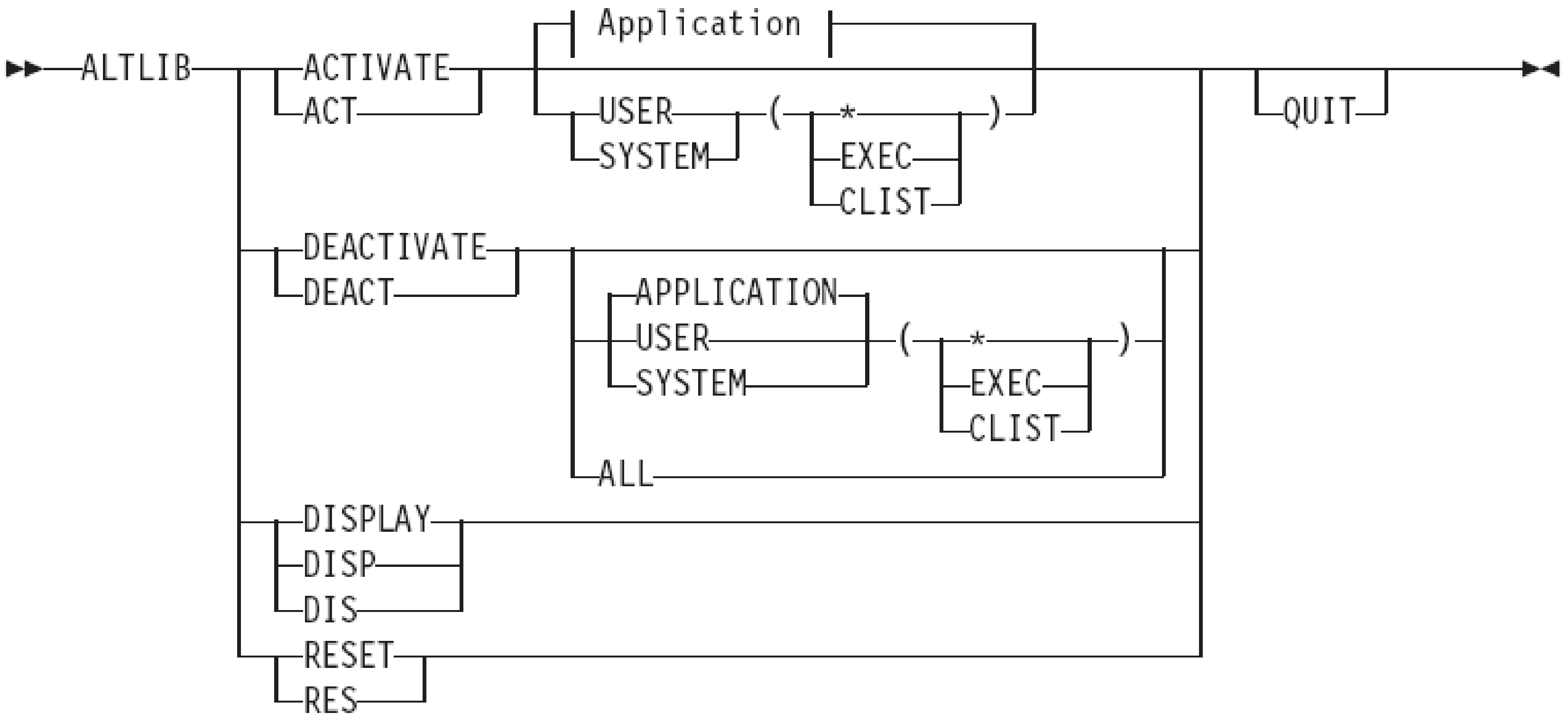
Type EXEC in front of member name.

```
exec__  RX20115  9  2005/08/12 2006/05/22 03:37:41 KOTMI01
```

Other kinds of invocation are mentioned in:

7) Subroutines and Functions
11) TSO REXX in TSO

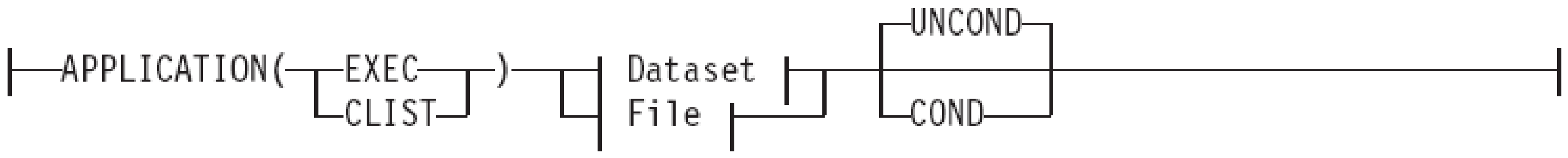
TSO Commands – ALTLIB



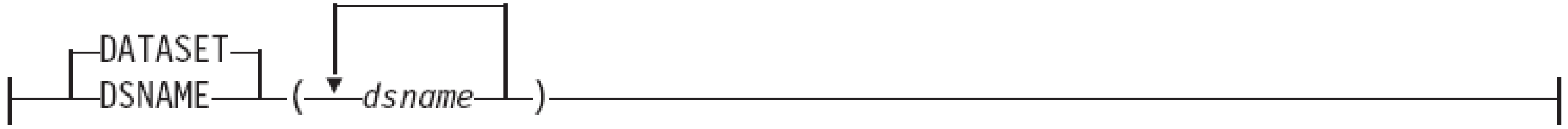
This course has been prepared by Milos Forman for MCoE needs only!

TSO Commands – ALTLIB (cont.)

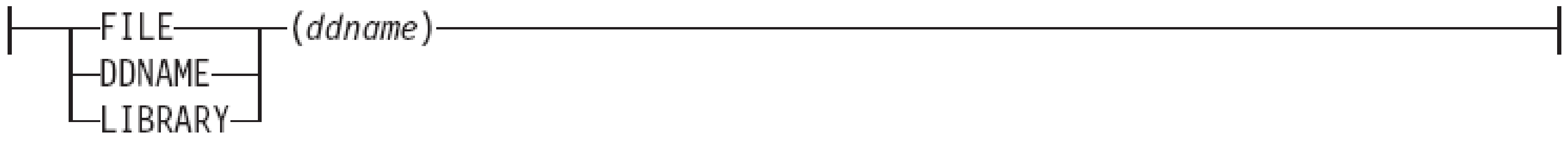
Application:



Dataset:

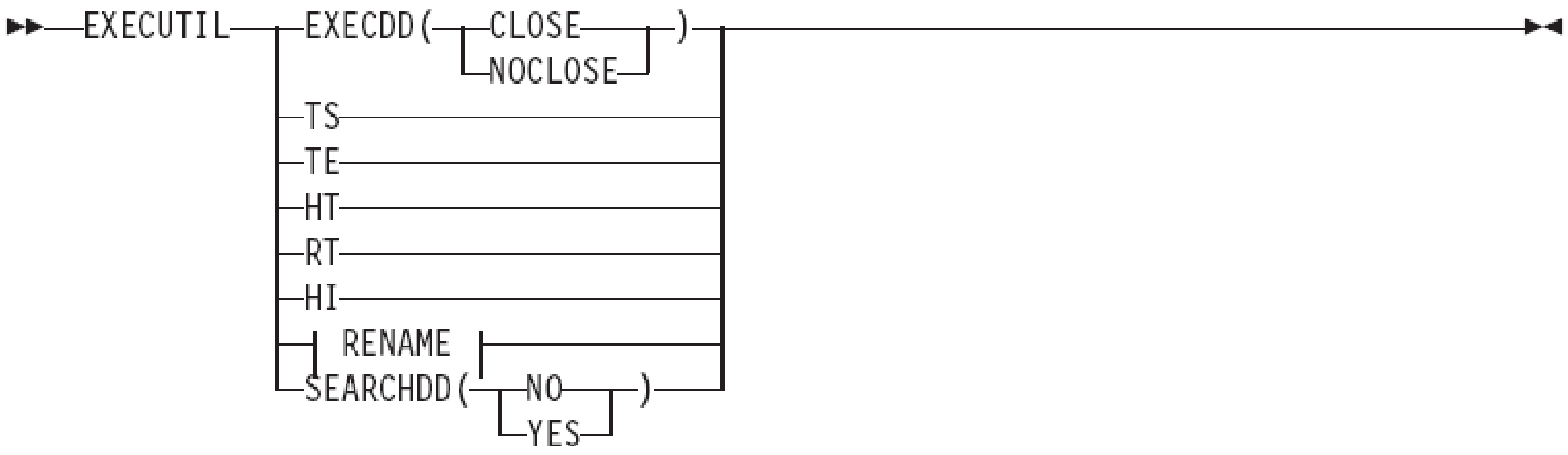


File:

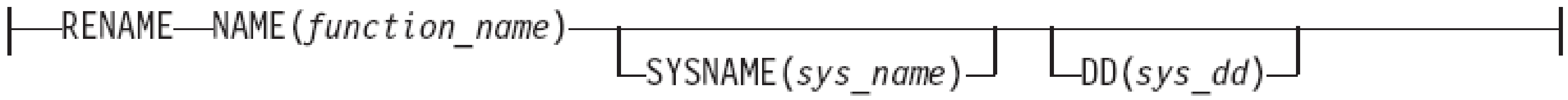


This course has been prepared by Milos Forman for MCoE needs only!

TSO Commands – Search order

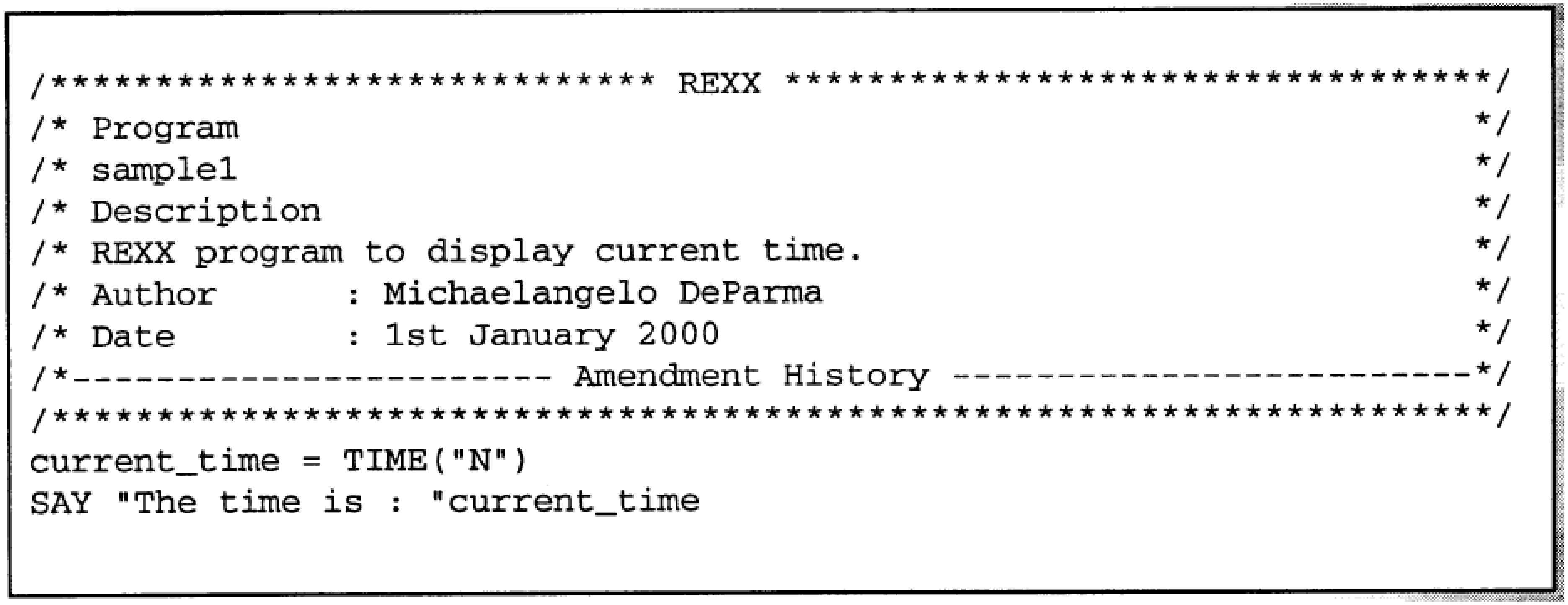


RENAME

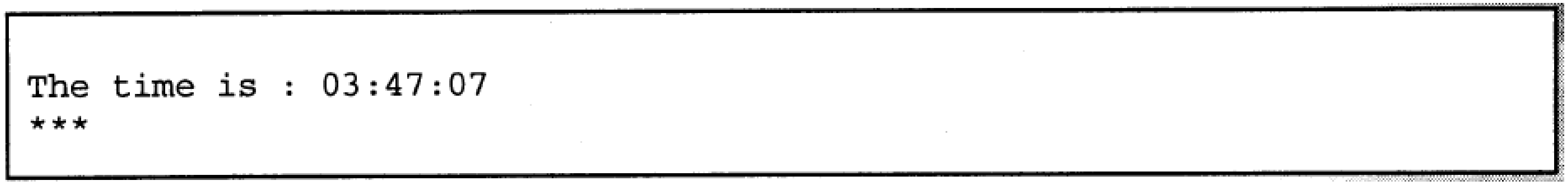


This course has been prepared by Milos Forman for MCoE needs only!

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
***
```

This course has been prepared by Milos Forman for MCoE needs only!

REXX Instruction Format

- 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.

This course has been prepared by Milos Forman for MCoE needs only!

REXX Instruction Format

- 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
```

This course has been prepared by Milos Forman for MCoE needs only!

SAY Instruction

- Output
- Format

```
SAY [expression]
```

- Examples

```
SAY error_message
SAY "Enter the error message :"
```

PARSE PULL Instruction

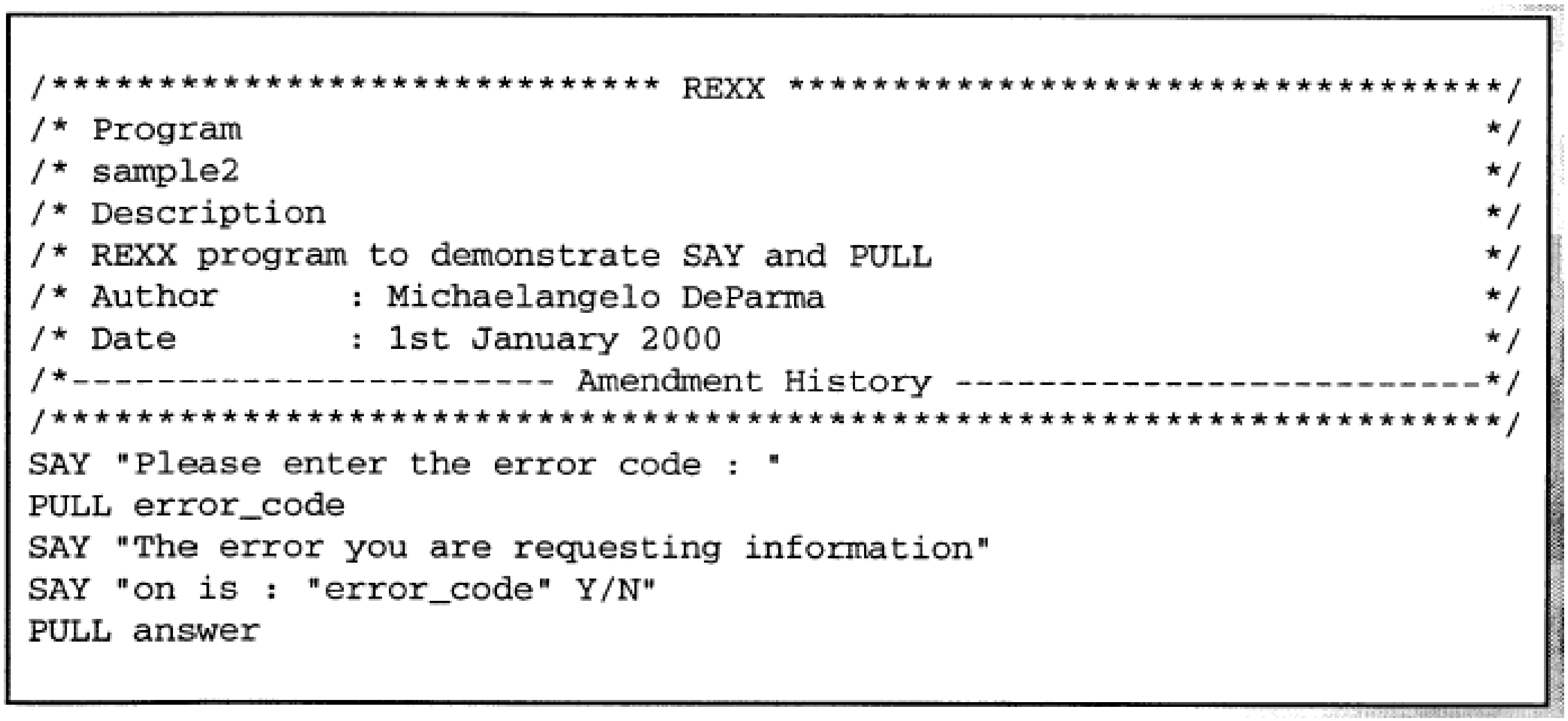
- Input
- Format

```
PARSE PULL [variable]
PULL [variable]
```

- Examples

```
PARSE PULL error_message
PULL answer
```

SAY and PULL Instruction



```
/****************************** 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
```

Work Section 1.1

- 1. Write a REXX program to display a list of three names.

```
Bob Flemming
Gary Stinker
John Thomas
***
```

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.
 ***
```

This course has been prepared by Milos Forman for MCoE needs only!

Additional Program

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

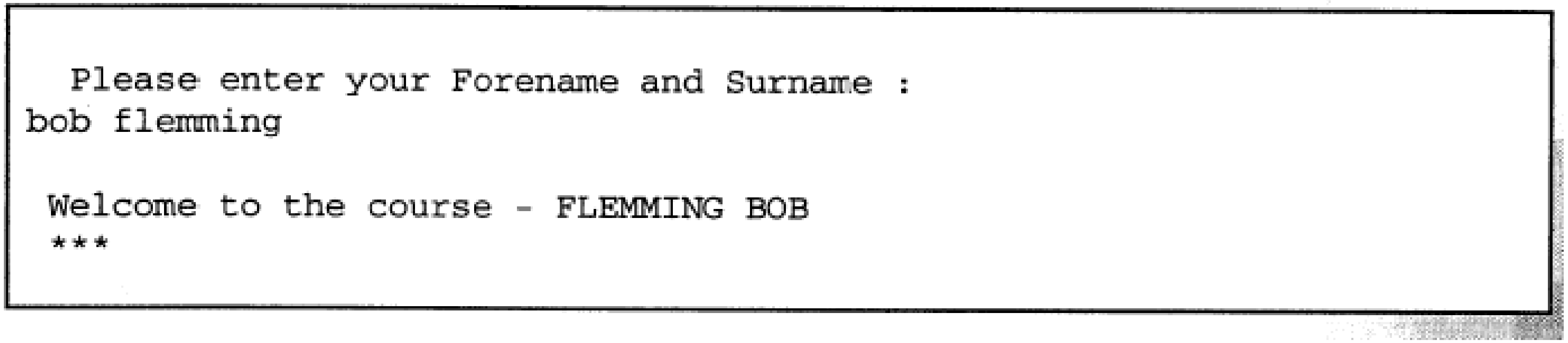
```
 Please enter your forename :
bob

 Please enter your surname :
flemming

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

Additional Program

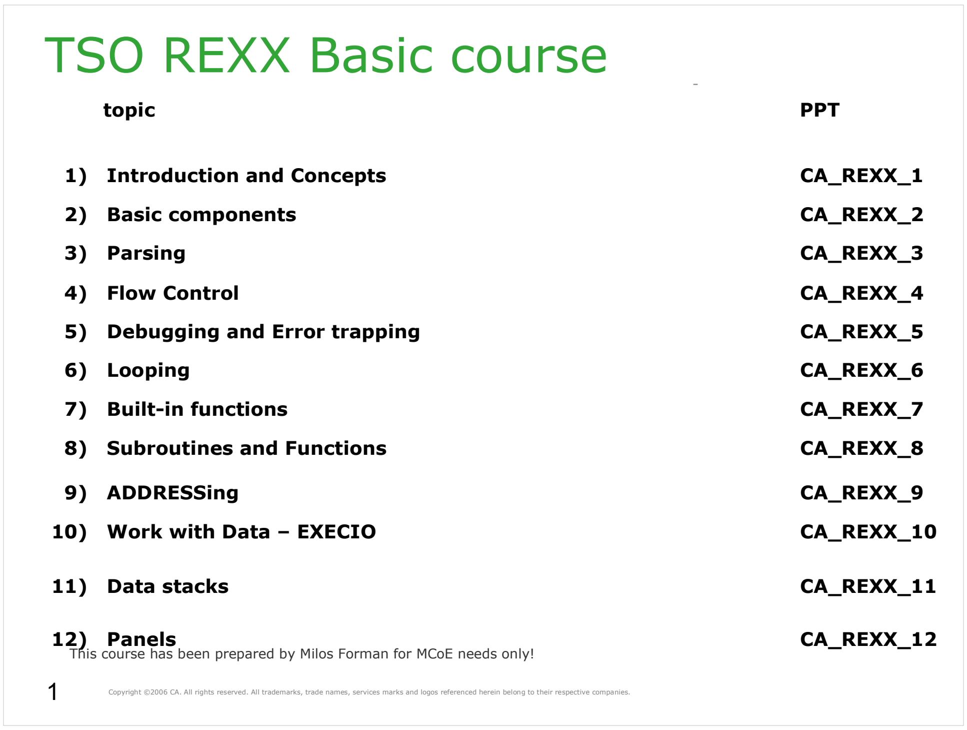
- Write a REXX program to collect a forename and surname on one line and display them both to the screen in reverse order..



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

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

This course has been prepared by Milos Forman for MCoE needs only!

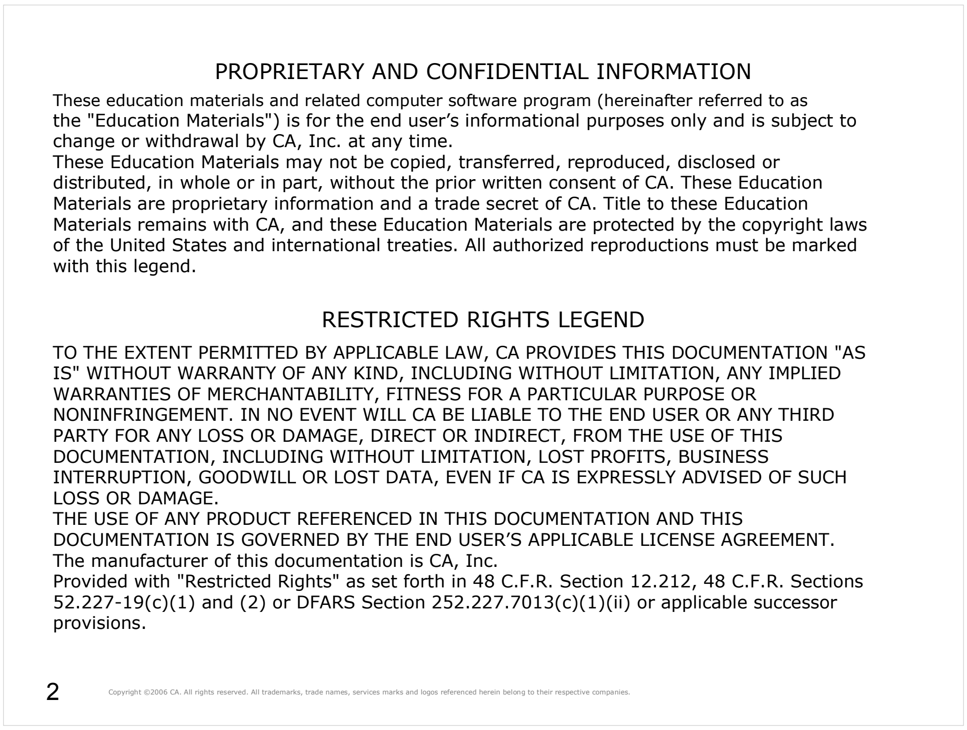


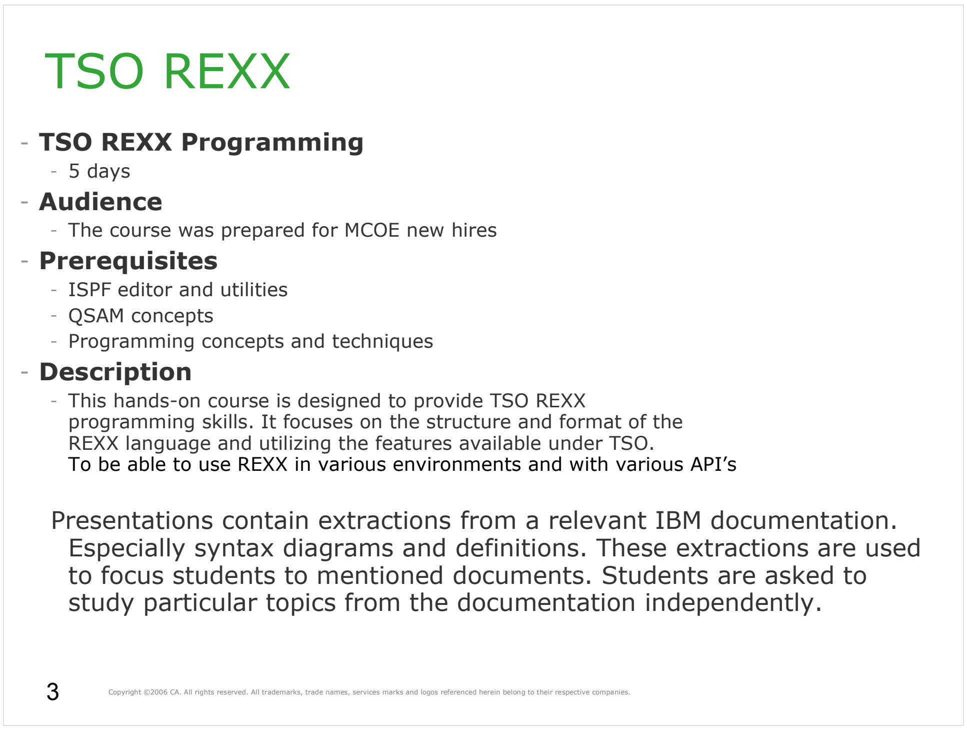
TSO REXX Basic course

	topic	PPT
1)	**Introduction and Concepts**	**CA_REXX_1**
2)	**Basic components**	**CA_REXX_2**
3)	**Parsing**	**CA_REXX_3**
4)	**Flow Control**	**CA_REXX_4**
5)	**Debugging and Error trapping**	**CA_REXX_5**
6)	**Looping**	**CA_REXX_6**
7)	**Built-in functions**	**CA_REXX_7**
8)	**Subroutines and Functions**	**CA_REXX_8**
9)	**ADDRESSing**	**CA_REXX_9**
10)	**Work with Data – EXECIO**	**CA_REXX_10**
11)	**Data stacks**	**CA_REXX_11**
12)	**Panels**	**CA_REXX_12**

This course has been prepared by Milos Forman for MCoE needs only!

1




TSO REXX

- **TSO REXX Programming**
  - 5 days
- **Audience**
  - The course was prepared for MCOE new hires
- **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 utilizing the features available under TSO. To be able to use REXX in various environments and with various API's

Presentations contain extractions from a relevant IBM documentation. Especially syntax diagrams and definitions. These extractions are used to focus students to mentioned documents. Students are asked to study particular topics from the documentation independently.

3

You can install Open Object REXX from folder: \\CZPRDB02\install\OOREXX\oorexx320-1.exe

QSAM – Queued Sequential Acceess Method. An extended version of the basic sequential access method (BSAM).

1) Introduction and Concepts

1) Introduction + Concepts:
Course objectives.
Documentation and resources.
Concepts:
How, why and where REXX.
TSO command ALTLIB, EXEC…
REXX instruction format.
Instructions: SAY, PULL…

2) Basic components:
Clauses, Statements, Comments, Literal strings, Numbers, Symbols, Variables, Labels…
Arithmetic and Concatenation operators.
Instructions: Assignment, DROP, UPPER, NUMERIC…

3) Parsing:
Instructions: PARSE, ARG…
Paterns.

4) Flow control:
Logical operators, comparative operators.
Instructions: IF/THEN/ELSE, SELECT, NOP…

Introduction

5) Debugging and error trapping:
Instructions: SIGNAL, TRACE…
TSO Immediate commands HT, RT, HE, HI, TE, TS.

6) Looping:
Instructions: DO, LEAVE and ITERATE…

7) Subroutines and functions:
Internal, External.
Instructions: CALL, PROCEDURE. EXPOSE, RETURN, EXIT, INTERPRET…
Addressing: ADDRESS, OUTTRAP instruction.

8) Built-in functions:
DATATYPE, POS, LEFT/RIGHT, STRIP, ABBREV, FORMAT, COMPARE, arithmetic functions.
TSO Built-in functions.

This course has been prepared by Milos Forman for MCoE needs only!

Introduction

9) **Data stacks:**
QUEUE, PUSH instructions.
TSO commands and functions for work with stacks.

10) Work with data:
Instruction: EXECIO, Read/Write instruction.
TSO commands to work with datasets.

11) TSO REXX in TSO:
Panels: ISPF.
EDIT macros.
REXX in Background TSO or Non TSO address space.
REXX from JCL.
REXX in USS.

12) REXX Edit Macros (extended version):
Introduction.
Examples of ISPF Macros.
ISPF Commands.
Practice.

Introduction

13) REXX & ISPF (extended version):

Dialog.

Dialog Elements and Structure.

Starting Dialog.

Dialog Variables.

Table services.

File tailoring services.

Panel Definition.

Panel Definition Sections.

Panel Definition Procedural Sections.

Panel Definition Control variables.

14) Compiler and Library (extended version):

Compiler inputs/outputs.

Compiler invocation.

Compiler Option and Control directives.

Other considerations.

Introduction

15) REXX in Different Address Spaces (extended version):

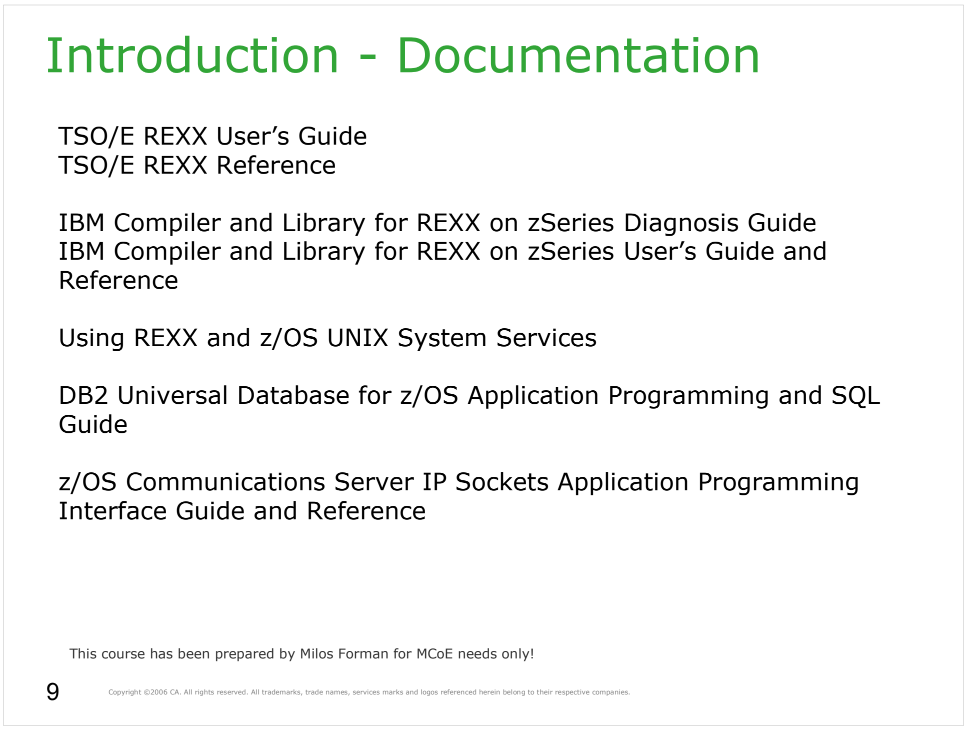
REXX in Batch.
Invocation of REXX from Programming languages.
Manipulating with current generation of REXX variables.

16) REXX in USS (extended version):

Available environments.
SYSCALL.
Built-in Functions.

17) REXX Sockets API (extended version):

Socket concept.
Client / Server.
REXX Socket API.



Introduction - Documentation

TSO/E REXX User’s Guide
TSO/E REXX Reference

IBM Compiler and Library for REXX on zSeries Diagnosis Guide
IBM Compiler and Library for REXX on zSeries User’s Guide and Reference

Using REXX and z/OS UNIX System Services

DB2 Universal Database for z/OS Application Programming and SQL Guide

z/OS Communications Server IP Sockets Application Programming Interface Guide and Reference

This course has been prepared by Milos Forman for MCoE needs only!

9

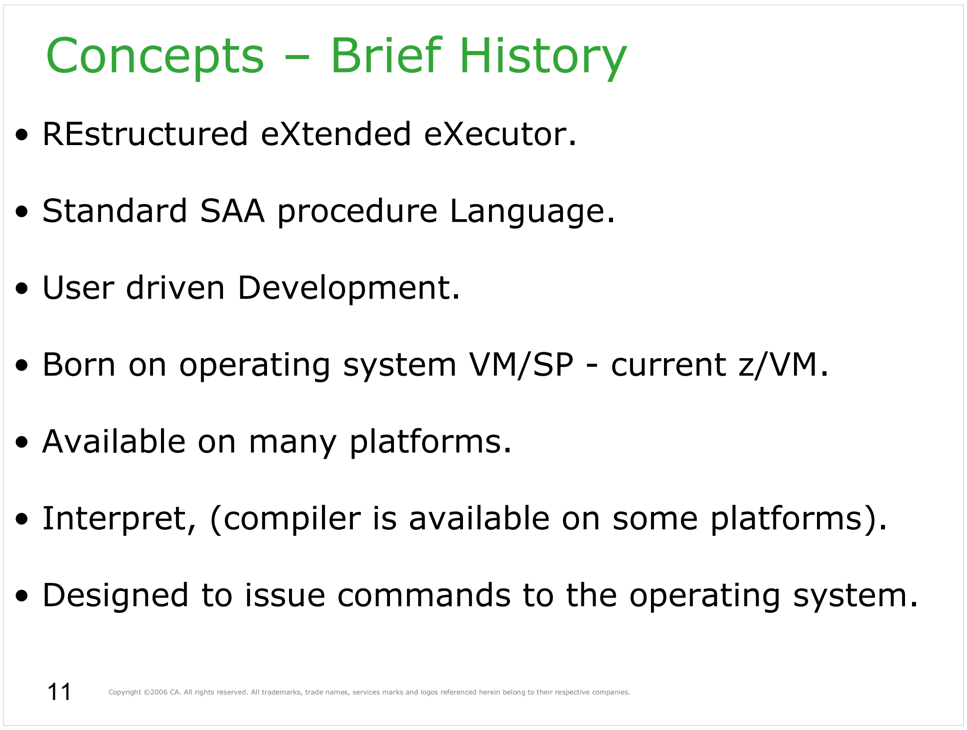
There is a list of documentation resources in each presentation.

REXX Samples

REXX samples: (You should first copy them under your pmfID)

```
MCOE.REXA.REXX     -> pmfid.REXA.REXX
MCOE.REXA.JCL      -> pmfid.REXA.JCL
MCOE.REXA.SKEL     -> pmfid.REXA.SKEL
MCOE.REXA.PANEL    -> pmfid.REXA.PANEL

/u/users/kotmi01/REXX       -> /u/users/pmfid
```



Concepts – Brief History

- REstructured eXtended eXecutor.
- Standard SAA procedure Language.
- User driven Development.
- Born on operating system VM/SP - current z/VM.
- Available on many platforms.
- Interpret, (compiler is available on some platforms).
- Designed to issue commands to the operating system.

11

SAA – System Application Architecture.

Concepts – Why Program in REXX

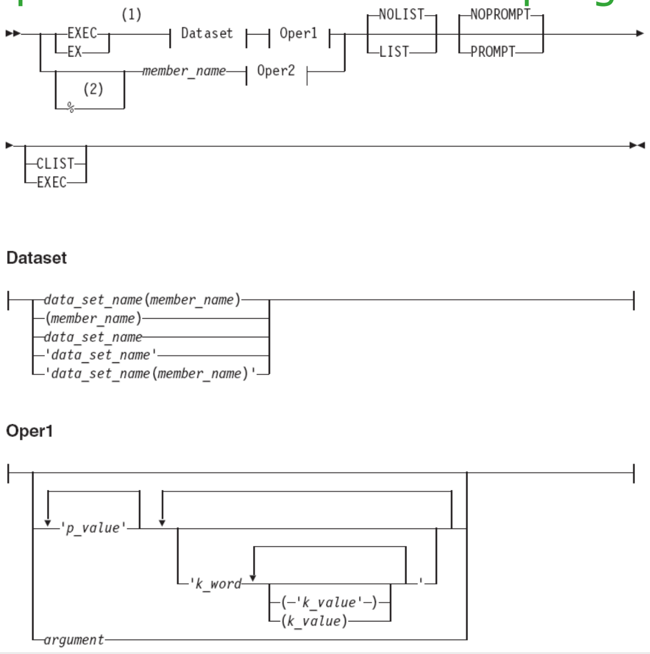
- Easy to read and write, using meaningful English words.
- Very litle punctuation.
- Free format – instruction are not column dependant and can span multiple line.
- Variable declaration is not required.
- Includes built-in functions.
- Trace capabilities.
- Extensive word and character manipulation.

Concepts – Where to use REXX

Use REXX to:

- Create easy and error free tasks.
- Automate repetitive tasks.
- Create program that behave like system commands.
- Tailor and enhance software systems.
- Create complex application.
- Ad hoc applications.

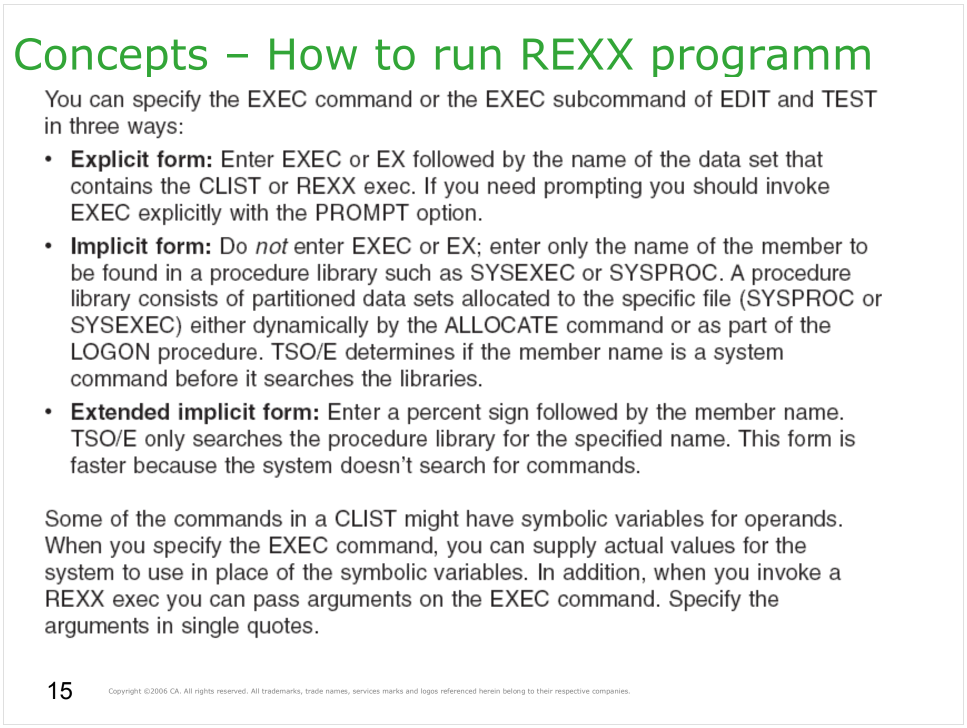
Concepts – How to run REXX program.



14

Use the EXEC command to execute a CLIST or REXX exec. Syntax diagram is commented on the next slide.

See TSO Command Reference.



Concepts – How to run REXX programm

You can specify the EXEC command or the EXEC subcommand of EDIT and TEST in three ways:

- **Explicit form:** Enter EXEC or EX followed by the name of the data set that contains the CLIST or REXX exec. If you need prompting you should invoke EXEC explicitly with the PROMPT option.
- **Implicit form:** Do *not* enter EXEC or EX; enter only the name of the member to be found in a procedure library such as SYSEXEC or SYSPROC. A procedure library consists of partitioned data sets allocated to the specific file (SYSPROC or SYSEXEC) either dynamically by the ALLOCATE command or as part of the LOGON procedure. TSO/E determines if the member name is a system command before it searches the libraries.
- **Extended implicit form:** Enter a percent sign followed by the member name. TSO/E only searches the procedure library for the specified name. This form is faster because the system doesn't search for commands.

Some of the commands in a CLIST might have symbolic variables for operands. When you specify the EXEC command, you can supply actual values for the system to use in place of the symbolic variables. In addition, when you invoke a REXX exec you can pass arguments on the EXEC command. Specify the arguments in single quotes.

15

Explicit form: TSO EXEC ‚MCOE.REXA.REXX(RX20115)‚

Implicit form: The REXX library must be allocated to SYSEXEC or SYSPROC DD name.

See ‘MCOE.REXA.REXX(SETEXEC and SETPROC)’

But be careful with these execs. If you execute them, they can deallocate other SYSEXEC and SYSPROC libraries.

You can also use CONCATDD program – see my #KOTMI01 member.

We will see later how to pass arguments to REXX exec.

Concepts – How to run REXX program

Command Exec can be invoked:

TSO command line:

```
tso exec ‘mcoe.rexa.rexx(rx20115)’
```

From ISPF:

p=PDF

6=Command

```
exec 'mcoe.rexa.rexx(rx20115)'
```

Type EXEC in front of member name.

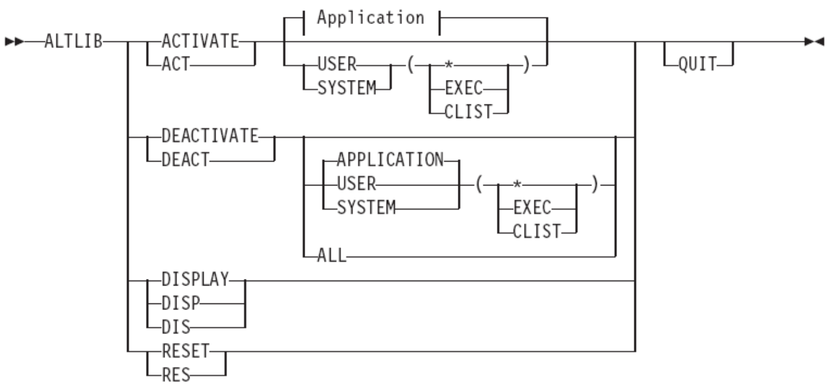
```
exec__  RX20115  9  2005/08/12 2006/05/22 03:37:41 KOTMI01
```

Other kinds of invocation are mentioned in:

7) Subroutines and Functions

11) TSO REXX in TSO

TSO Commands – ALTLIB



This course has been prepared by Milos Forman for MCoE needs only!

17

Use the ALTLIB command to:

- Define alternative application-level libraries of REXX execs or CLISTs.

- Indicate that user-, application-, and system-level libraries of REXX execs and CLISTs are being searched before SYSEXEC and SYSPROC system libraries.

- Exclude one or more library levels (user, application, system) from being searched.

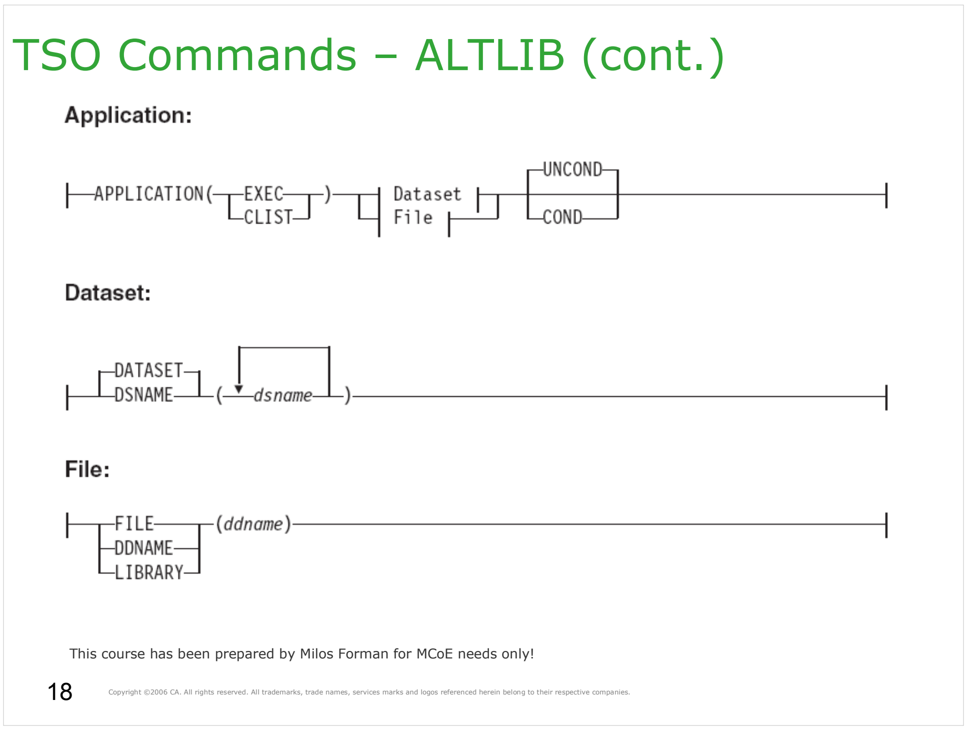
- Reset the search order to the system level.

- Obtain a display of the search order that is in effect.

TSO/E searches the user-, application-, and system-level libraries for REXX execs or CLISTs that are executed implicitly or when searching for REXX external functions or subroutines.

The Command is mentioned for completnes, so we will not go to details.

See TSO Command Reference.



TSO Commands – ALTLIB (cont.)

Application:

```
                                        ┌UNCOND┐
├─APPLICATION(─┬─EXEC──┬─)─┤ Dataset ├─┤      ├──────────────────────────┤
               └─CLIST─┘   │ File    │ └COND──┘
```

Dataset:

```
                     ┌───────┐
├─┬─DATASET─┬─(─▼─dsname─┴─)──────────────────────────────────────────┤
  └─DSNAME──┘
```

File:

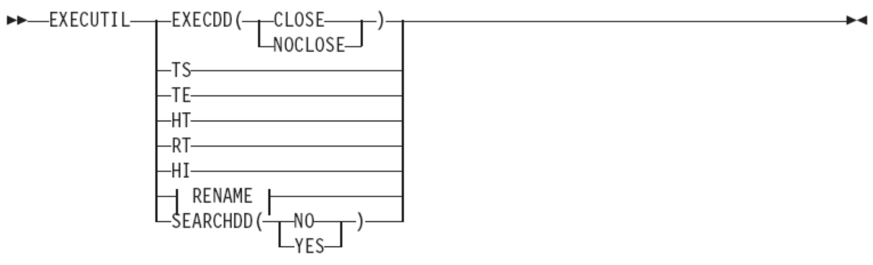
```
├─┬─FILE────┬─(ddname)────────────────────────────────────────────────┤
  ├─DDNAME──┤
  └─LIBRARY─┘
```

This course has been prepared by Milos Forman for MCoE needs only!

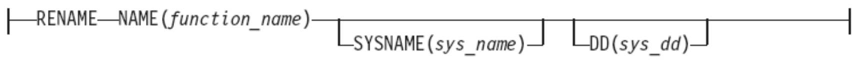
18

See ‘MCOE.REXA.REXX(ALTLIB and ALTEST)’

TSO Commands – Search order



RENAME



This course has been prepared by Milos Forman for MCoE needs only!

Use EXECUTIL to:

- Specify whether the system exec library, whose default name is SYSEXEC, is to be closed upon completion of the exec or is to remain open

- Start and stop tracing of an exec

-Stop the execution of an exec

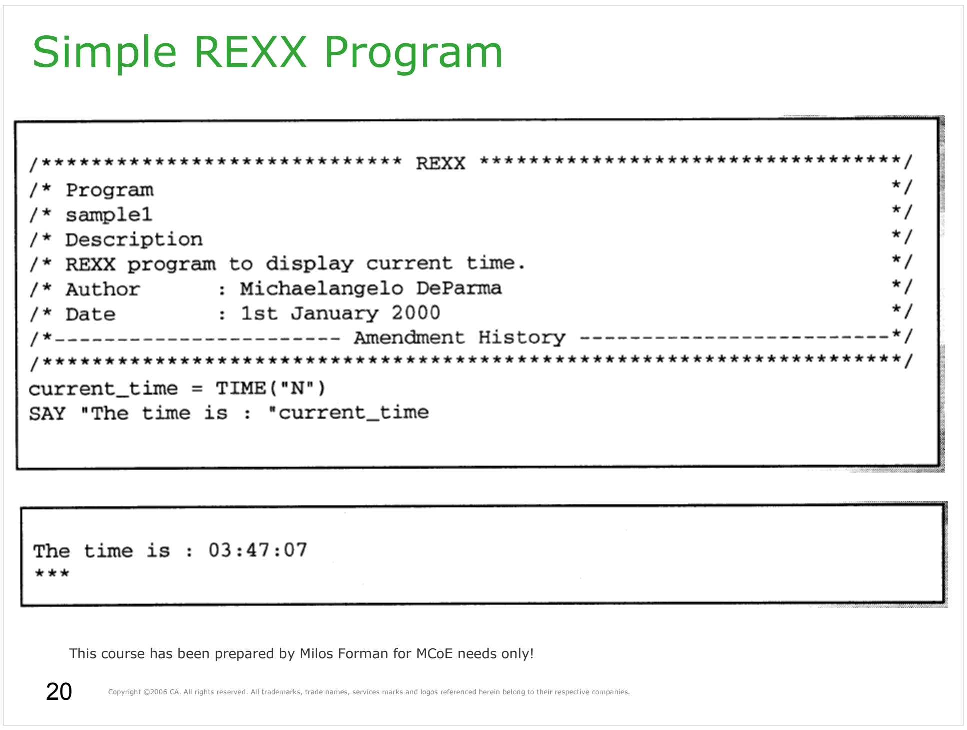
-**Suppress and resume terminal output from an exec**

- Change entries in a function package directory

- Specify whether the system exec library (the default is SYSEXEC) is to be searched in addition to SYSPROC.

The Command is mentioned for completnes, so we will not go to details.

See TSO Command Reference.



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
***
```

This course has been prepared by Milos Forman for MCoE needs only!

20

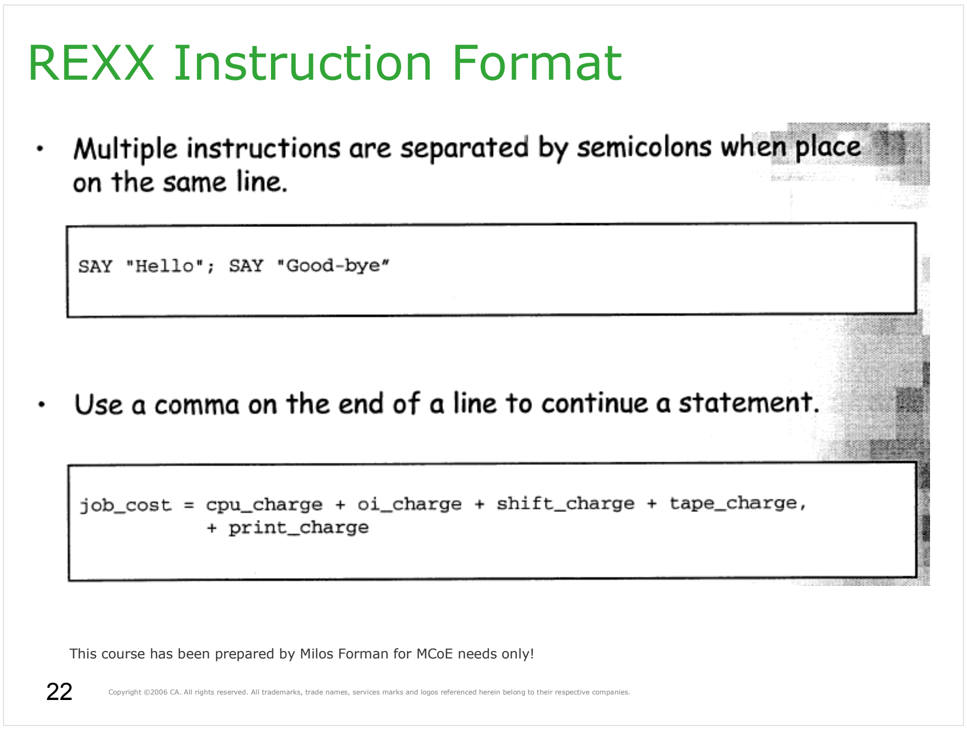
Try to write it and execute it.

See ‘MCOE.REXA.REXX(RX20115)’

REXX Instruction Format

- 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.

Just read the slide.



REXX Instruction Format

- Multiple instructions are separated by semicolons when place on the same line.

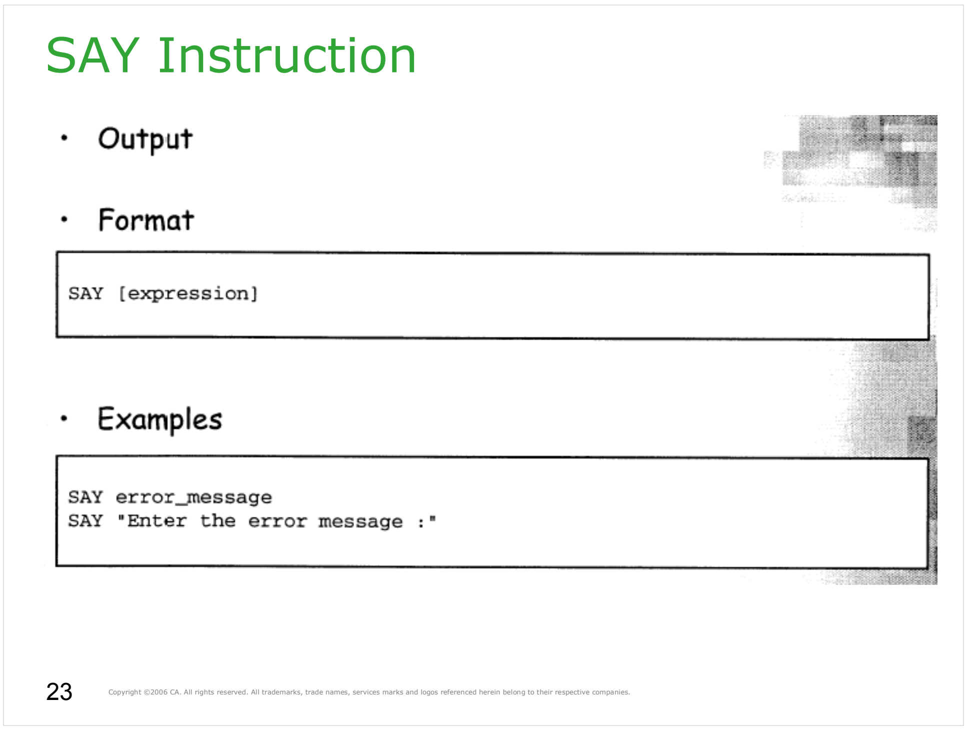
```
SAY "Hello"; SAY "Good-bye"
```

- 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
```

This course has been prepared by Milos Forman for MCoE needs only!

Just read the slide.



SAY Instruction

- Output
- Format

```
SAY [expression]
```

- Examples

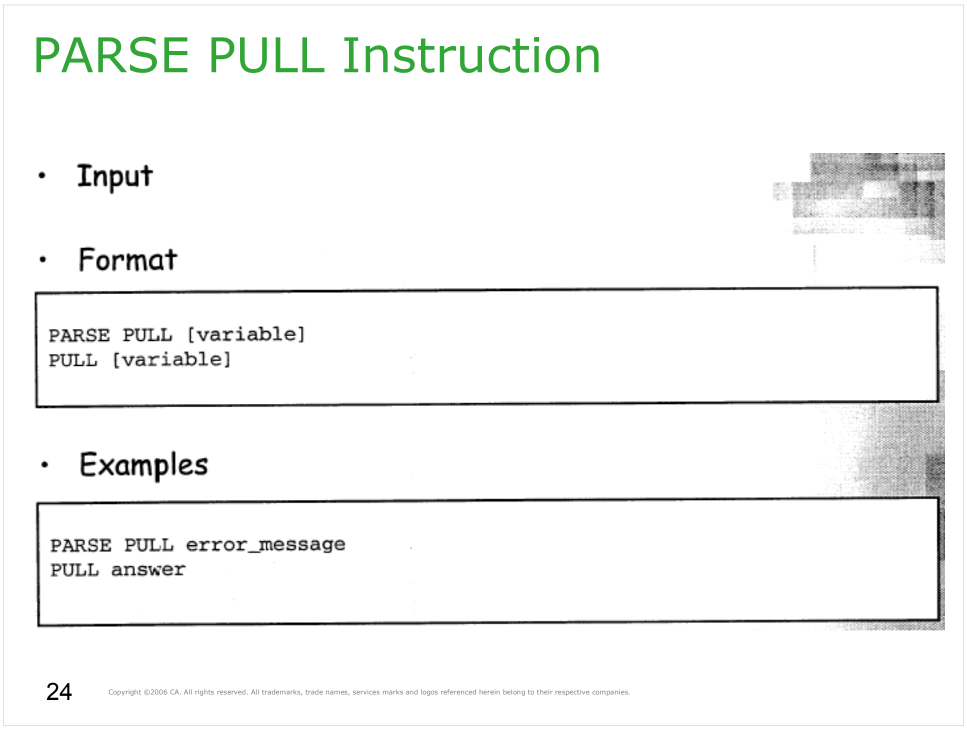
```
SAY error_message
SAY "Enter the error message :"
```

23

Example of output instruction.

SAY writes a line to the output stream. This typically displays it to the user.

See TSO/E REXX Reference.



PARSE PULL Instruction

- Input
- Format

```
PARSE PULL [variable]
PULL [variable]
```

- Examples

```
PARSE PULL error_message
PULL answer
```

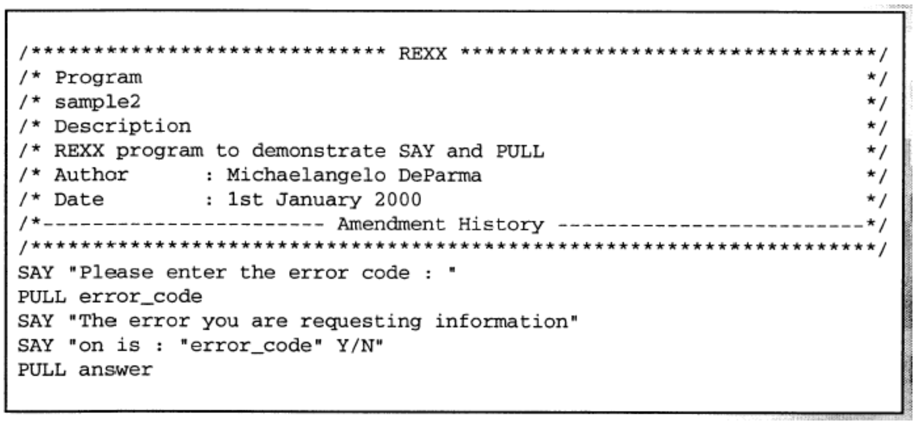
24

Example of input instructions.

PARSE assigns data (from various sources) to one or more variables.

See TSO/E REXX Reference.

SAY and PULL Instruction



```
/****************************** 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
```

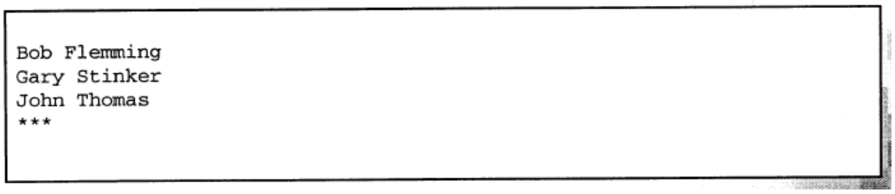
This course has been prepared by Milos Forman for MCoE needs only!

Try to write it and execute it.

See ‘MCOE.REXA.REXX(RX201110)’

Work Section 1.1

- 1. Write a REXX program to display a list of three names.

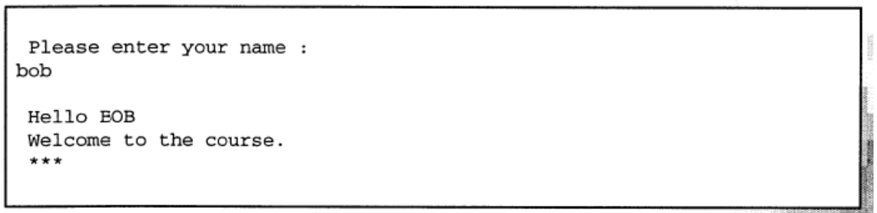


```
Bob Flemming
Gary Stinker
John Thomas
***
```

This course has been prepared by Milos Forman for MCoE needs only!

Work Section 1.2

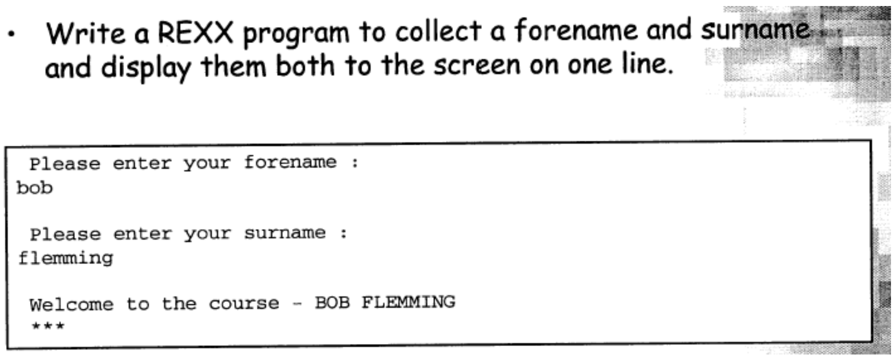
- 2. Write a REXX program to collect and name from the screen and display a welcome to the screen.



```
 Please enter your name :
bob

 Hello BOB
 Welcome to the course.
 ***
```

Additional Program



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

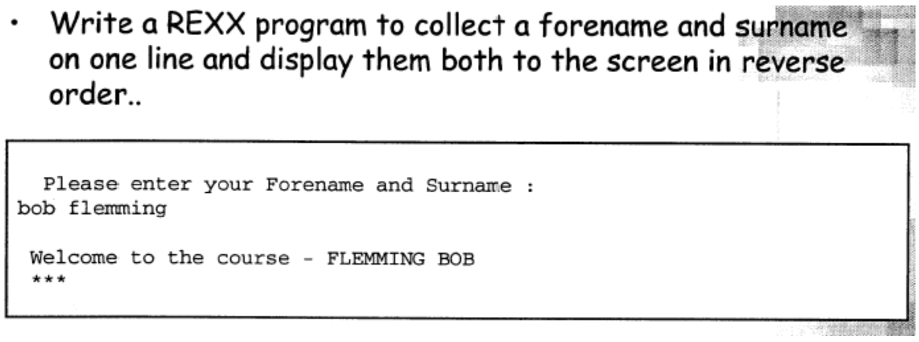
```
 Please enter your forename :
bob

 Please enter your surname :
flemming

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

This course has been prepared by Milos Forman for MCoE needs only!

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
 ***
```

This course has been prepared by Milos Forman for MCoE needs only!