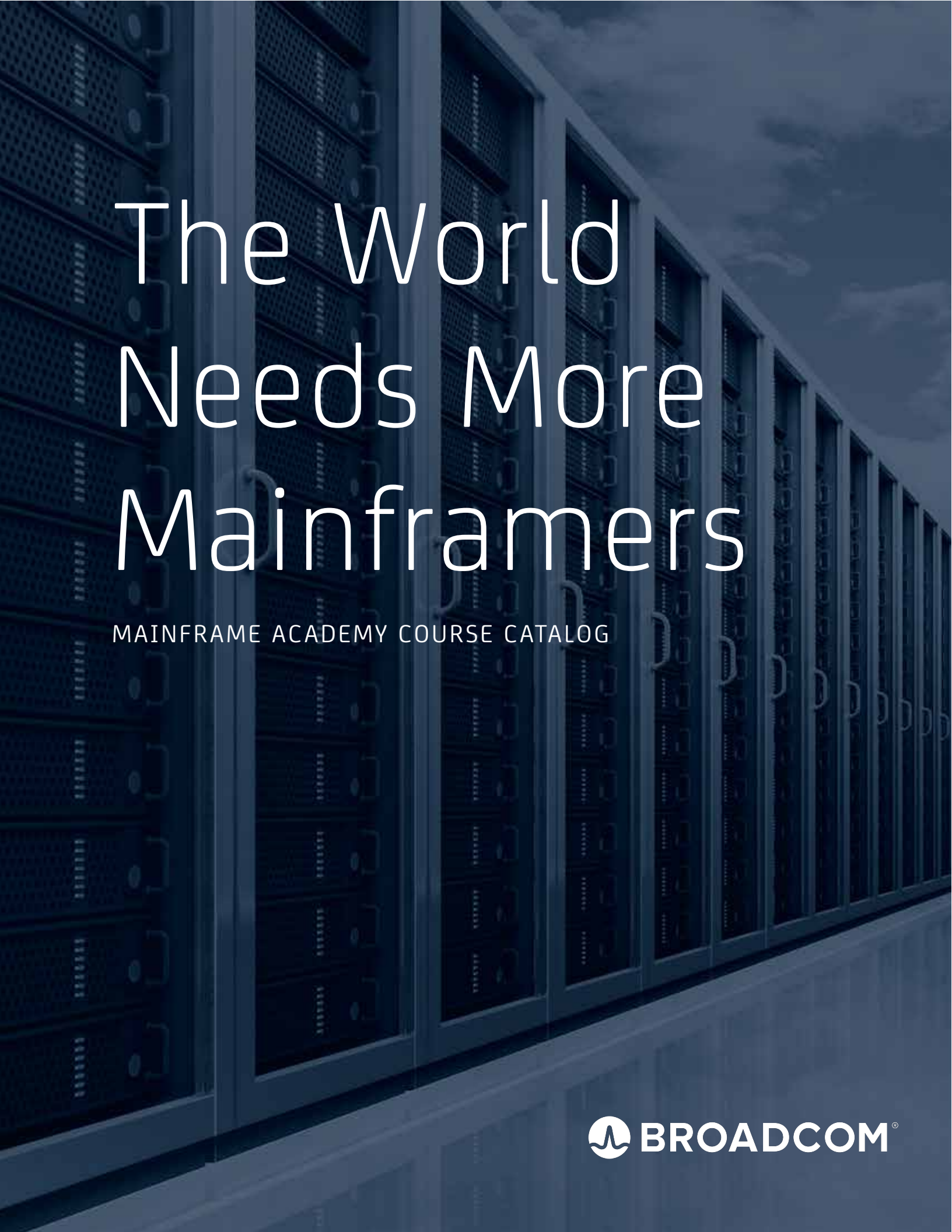


The World Needs More Mainframers

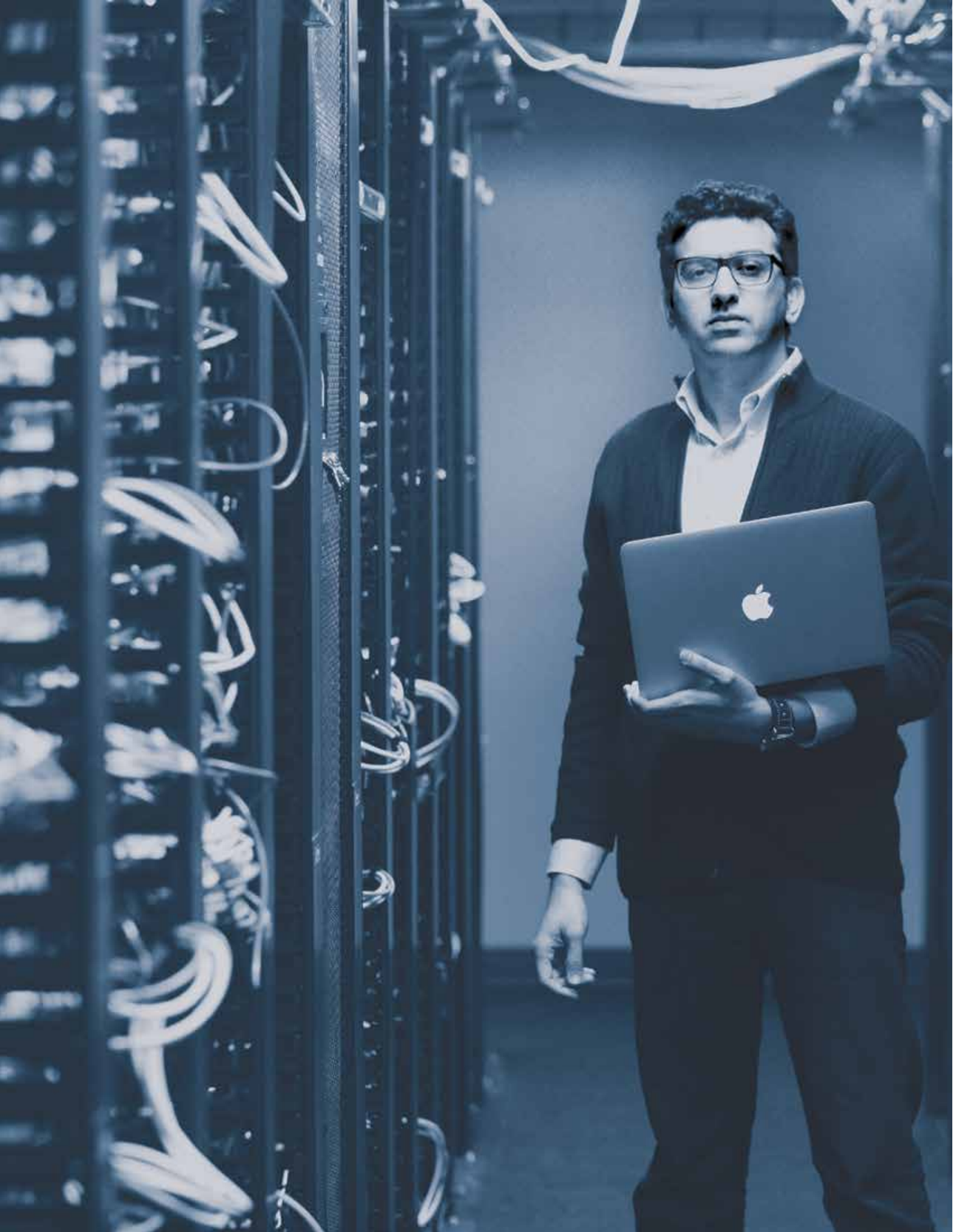
MAINFRAME ACADEMY COURSE CATALOG

BROADCOM®

Table of Contents

Mainframe Academy

Mainframe Academy with CA Technologies is a curriculum of core mainframe programming skills designed to accelerate learning and technical achievement via real world scenarios and specialized courses, all within a single class cohort. Mainframe Academy includes over 220 hours of training, 26 mastery tests and 3 certification exams focused on a targeted spectrum of mainframe processing, programming concepts and applications. The content is delivered with a unique, flexible blend of instructor-led, web-based and self-paced learning.



Mastering the mainframe starts right here.

I can discover new ways to utilize and leverage the mainframe. As a 26-year veteran of distributed platforms, I had avoided the mainframe. I am now seeing a shift in the way companies use their mainframe technology and an increase in the number of applications moving to z/OS. I recognized I needed to move with them and enhance my skill set, even in the middle of my career.

With Mainframe Academy, I am more self-sufficient and am able resolve more issues on my own. In the past, I took a lot of courses and could not recall the content a year later. Mainframe Academy with CA Technologies is different. Because of the multi-sensory approach, I was able to fully grasp the material. I was able to gather and retain the skills I need to remain competitive and effective.

Kevin Kubacki
Senior Support Engineer

"The best thing about Mainframe Academy is the teamwork. Sitting down with other students and collectively working through a project has made me learn even more."

– Kevin Kubacki, Sr. Support Engineer

"I feel like I truly now understand JCL."

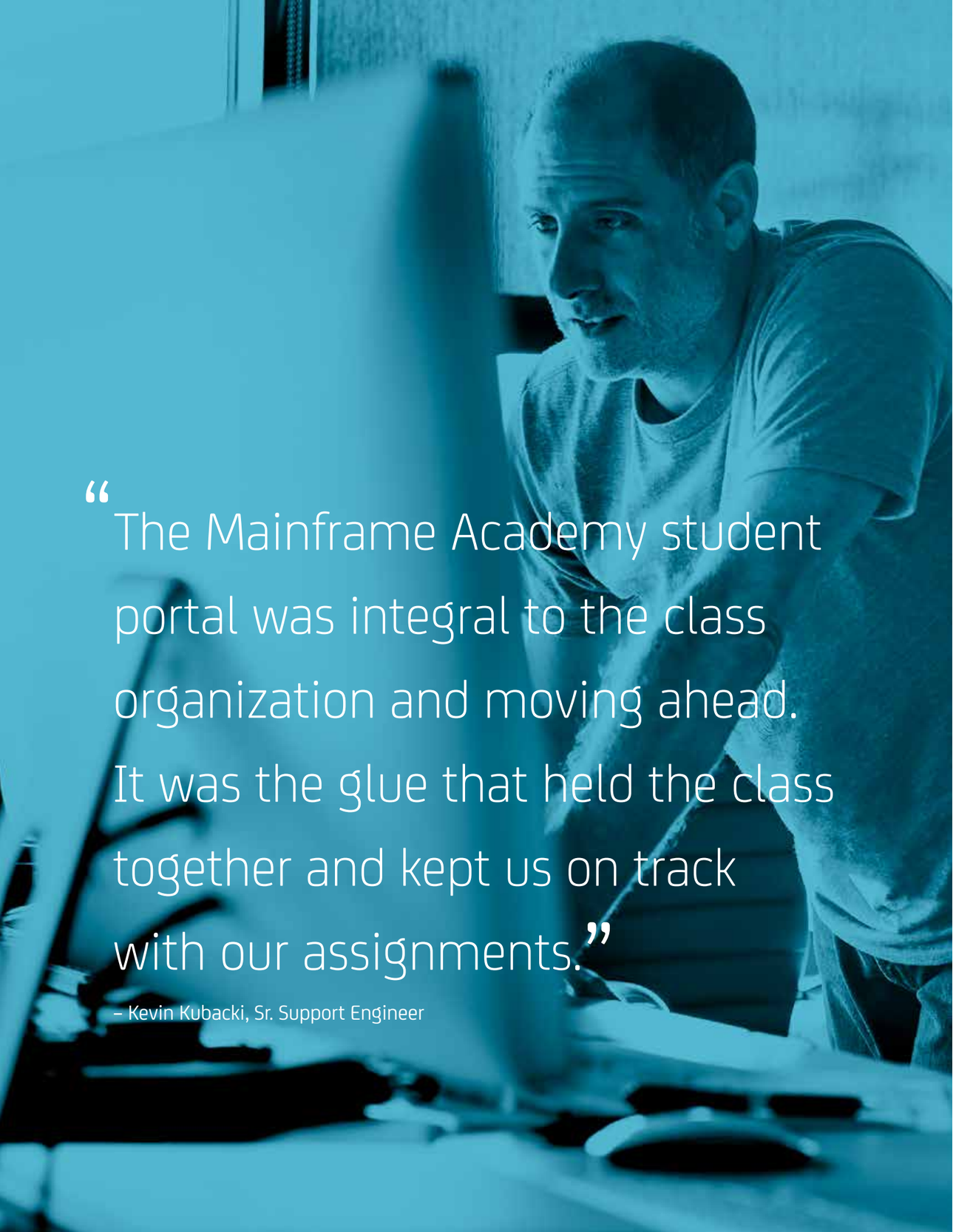
– Sam Thomas, Support Engineer

"I wish I had this kind of training when I was starting in mainframe."

– Mainframe Academy student

"Mainframe Academy can be valuable on your resume. The certification is important."

– Sam Thomas, Support Engineer



“The Mainframe Academy student portal was integral to the class organization and moving ahead. It was the glue that held the class together and kept us on track with our assignments.”

– Kevin Kubacki, Sr. Support Engineer

Student Portal

The Mainframe Academy student portal is a unique, collaborative experience. Students can easily navigate and access relevant information. They can also track their progress, view current assignments and access course materials instantly.

This rich user interface is an important element for how students experience Mainframe Academy.

Course Overview

Mainframe Academy with CA Technologies
Course Code:
06MFO3005B (core 1)
06MFO3006B (core 2)

Course types:
Web-based training
Instructor-led training
Self-paced, hands-on projects

Mainframe Academy with CA Technologies is organized into two consecutive Cores, building discrete mainframe skill sets. Mainframe Academy with CA Technologies is specifically targeted to computer science graduates and experienced information system programmers with a desire to accelerate or expand their skill repertoire towards becoming a Certified Mainframe Professional by CA Technologies.

Mainframe Academy with CA Technologies delivers the rigorous technical course content over approximately seven weeks, adjusting for breaks and self-paced learning. The program has firm deadlines for materials due and is a commitment, both on the part of the participant and the sponsoring business. The program is specifically designed to help address mainframe skill gaps in organizations by broadening their in-house skills and helping information system professionals become effective on the mainframe.

Program Agenda

Core Part 1

Intro to z/OS: Data Management Systems
Overview and Documentation for z/OS (1 day)

TSO/ISPF for z/OS Curriculum (6 days)

JCL (7 days)

Live Debrief and Exam (2 days)

Core Part 2

Data Set Utilities (3 days)

REXX Scripting Skills (5 days)

Introduction to z/OS Systems Programming (2 days)

Live Debrief and Exam (1 days)

Final Certification Exam (1 day)

Sample Calendar

In person

Virtual web-based training

Virtual instructor

Virtual labs

1	2	3	4	5
8-9: Kick Off, Welcome, Introduction and Vision Setting **9-11:30: Intro to z/OS:** Data Management Systems Overview **11:30-12:** Documentation for z/OS **1-5: TSO/ISPF Phase 1:** Using Online Systems Facilities	**9-11: TSO/ISPF Phase 1:** Customizing the TSO/ISPF Environment **11-12: TSO/ISPF Phase 1:** Customizing the TSO/ISPF Environment – Lab **1-2: TSO/ISPF Phase 1:** Customizing the TSO/ISPF Environment – Lab cont'd **2-3: TSOISPF Phase 1:** Customizing the TSO/ISPF Environment Debrief **3-5: TSO/ISPF Phase 2:** Maintaining Data in Files with ISPF Editor 1.13	**9-3: TSO/ISPF Phase 2:** Maintaining Data in Files with ISPF Editor 1.13 cont'd + Test **3-5: TSO/ISPF Phase 2:** Editing Data Sets with the TSO/ISPF Editor	**9-11: TSO/ISPF Phase 2:** Editing Data Sets with the TSO/ISPF Editor - Lab **11-12: TSO/ISPF Phase 2:** Editing Data Sets with the TSO/ISPF Editor Debrief **1-5: TSO/ISPF Phase 3:** Managing Data Files and Definitions with ISPF/PDF 1.13	**9-11: TSO/ISPF Phase 3:** Managing Data Files and Definitions with ISPF/PDF 1.13 cont'd +Test **11- 12: TSO/ISPF Phase 3:** Working with Data Sets **1- 2: TSO/ISPF Phase 3:** Working with Data Sets cont'd **2-4: TSO/ISPF Phase 3:** Working with Data Sets – Lab **4- 5: TSO/ISPF Phase 3:** Working with Data Sets Debrief
6	**7**	**8**	**9**	**10**
9-11: TSO/ISPF Phase 4: Invoking Other Programs **11-12: TSO/ISPF Phase 4:** Invoking Other Programs - Lab **1-2: TSO/ISPF Phase 4:** Invoking Other Programs Debrief **2-4: TSO/ISPF Phase 5:** Advanced Concepts **4-5: TSO/ISPF Phase 5:** Advanced Concepts - Lab	**9-11: TSO/ISPF Phase 5:** Advanced Concepts Debrief and Review **11-12: JCL Phase 1:** Intro to z/OS JCL 1.13 **1-5: JCL Phase 1:** Intro to z/OS JCL 1.13 cont'd + Test	**9-12: JCL Phase 1:** Create the Job for Generating the List of States - Lab **1-4: JCL Phase 1:** Create the Job for Generating the List of States - Lab **4-5: JCL Phase 1:** Create the Job for Generating the List of States Debrief	**9-12: JCL Phase 2:** JCL – Coding Procedures and JES2 Control Statements 1.13 **1-3: JCL Phase 2:** JCL – Coding Procedures and JES2 Control Statements 1.13 cont'd + Test **3-5: JCL Phase 2:** Create a Job that Can Run on Demand	**9-10: JCL Phase 2:** Create a Job that Can Run on Demand cont'd **10-12: JCL Phase 2:** Create a Job that Can Run on Demand - Lab **1-3: JCL Phase 2:** Create a Job that Can Run on Demand - Lab cont'd **3-4: JCL Phase 2:** Create a Job that Can Run on Demand Debrief **4-5: JCL Phase 3:** Using Special Data Sets in Batch Jobs 1.13
11	**12**	**13**	**14**	**15**
9-12: JCL Phase 3: Using Special Data Sets in Batch Jobs 1.13 cont'd **1-4: JCL Phase 3:** Using Special Data Sets in Batch Jobs 1.13 cont'd + Test **4-5: JCL Phase 3:** Create System Generated Daily Backups	**9-10: JCL Phase 3:** Create System Generated Daily Backups Cont'd **10-11: JCL Phase 2:** Create a Job that can Run on Demand and Debrief **11-12: JCL Phase 3:** Create System Generated Daily Backups – Lab **1-3: JCL Phase 4:** Identifying and Resolving Batch Problems in JCL 1.13 **3-5: JCL Phase 4:** Maximize Efficiency by Cancelling the Job When Errors are Present	**9-11: JCL Phase 4:** Maximize Efficiency by Cancelling the Job When Errors are Present – Lab **11-12: JCL Phase 4:** Maximize Efficiency by Cancelling the Job When Errors are Present Debrief **1-3: JCL Phase 5:** Identifying and Resolving Batch Problems in JCL 1.13 **3-5: JCL Phase 5:** Correct JCL Errors		

Sample Calendar

In person

Virtual web-based training

Virtual instructor

Virtual labs

16	17	18	19	20
	9-11: JCL Phase 5: Correct JCL Errors – Lab **1-5: Core Part 1:** Debrief	**9-12: Core Part 1:** Debrief cont'd **1-5: Core Part 1:** Procter Exam		
21	**22**	**23**	**24**	**25**
9-10: Utilities Phase 1: Data Utilities **10-11: Utilities Phase 1:** General Data Set Utilities **11-12: Utilities Phase 1:** Data Set Utilities **1-2: Utilities Phase 1:** Data Set Utilities cont'd **2-5: Utilities Phase 1:** Data Set Utilities - Lab	**9-10: Utilities Phase 1:** Data Set Utilities Debrief **10-12: Utilities Phase 2:** Virtual Storage Access Method (VSAM) Utilities **1-4: Utilities Phase 2:** Virtual Storage Access Method (VSAM) Utilities cont'd + Test **4-5: Utilities Phase 2:** Virtual Storage Access Method (VSAM) Utilities	**9-11: Utilities Phase 2:** Virtual Storage Access Method (VSAM) Utilities - Lab **11-12: Utilities Phase 2:** Virtual Storage Access Method (VSAM) Utilities Debrief **1-3: Utilities Phase 3:** Data Utilities **3-3:30: Utilities Phase 3:** The SORT/MERGE Utility **3:30-5: Utilities Phase 3:** The SORT/MERGE Utility - Lab	**9-10: Utilities Phase 3:** The SORT/MERGE Utility Debrief and Review **10-12: REXX Phase 1:** Intro to the REXX Programming Language **1-3: REXX Phase 1:** Intro to the REXX Programming Language cont'd + Test **3-5: REXX Phase 1:** Intro to the REXX Programming Language	**9-11: REXX Phase 1:** Intro to the REXX Programming Language - Lab **11-12: REXX Phase 1:** Intro to the REXX Programming Language cont'd **1-2: REXX Phase 1:** Intro to the REXX Programming Language cont'd **2-4: REXX Phase 1:** Intro to the REXX Programming Language - Lab cont'd **4-5: REXX Phase 1:** Intro to the REXX Programming Language Debrief
26	**27**	**28**	**29**	**30**
9:-12: REXX Phase 2: REXX Parse Command and Conditional Processing **1-3: REXX Phase 2:** REXX Parse Command and Conditional Processing – Lab **3-5: REXX Phase 2:** REXX Parse Command and Conditional Processing Debrief	**9-12: REXX Phase 3:** REXX Advanced Processing **1-3: REXX Phase 3:** REXX Advanced Processing - Lab **3-5: REXX Phase 3:** REXX Advanced Processing cont'd	**9-11: REXX Phase 3:** REXX Advanced Processing - Lab cont'd **11-12:** REXX Topic Review **1-2: Systems Programming Phase 1:** Intro to the IBM Enterprise Environment 1.13 **2-4: Systems Programming Phase 1:** z/OS System Programming Fundamentals 1.13 **4-6: Systems Programming Phase 1:** Intro to z/OS Systems Programming Fundamentals	**9-10: Systems Programming Phase 1:** Intro to z/OS Systems Programming Fundamentals cont'd **10-12: Systems Programming Phase 2:** Intro to z/OS Systems Programming - SMP/E Processing **1-3: Systems Programming Phase 3:** Intro to z/OS Systems Programming - IBM Health Checker for z/OS **3-5: Systems Programming Phase 4:** Intro to z/OS Systems Programming - JES2	**9-11: Systems Programming Phase 4:** Intro to z/OS Systems Programming - JES2 cont'd **11-12: Systems Programming Phase 5:** Intro to z/OS Systems Programming - SDSF **1-3: Systems Programming Phase 5:** Intro to z/OS Systems Programming – SDSF cont'd **3-3:30: Systems Programming Phase 6:** z/OS Configuration Options and Subsystems
31	**32**	**33**		
	9-12: Core Part 2: Debrief **1-5: Core Part 2:** Debrief cont'd	**9-10:30: Core Part 2:** Debrief cont'd **10:30-12: Final Core Part 2:** Exam/Test **1-3:** Capstone Practicum Exam		

Introduction to the z/OS Environment and Documentation

Course Type and Duration:

Instructor-led training
Duration: 1 day

Course overview

This course will teach you the basic concepts behind the z/OS environment and set the stage for more detailed training on individual topics and skills. The course is web-based, with a follow-up, facilitator-led question and answer session.

You will learn how to:

- Recognize the differences in z/OS file systems
- Understand the concept of records/blocks
- Understand the concept of device type, volume serial number
- Use data set naming conventions
- Locating and using IBM® manuals
- Locate and use IBM Redbooks®
- Identify how an IBM enterprise system allocates and uses disk storage
- Recognize the different types of data sets
- Identify how source and code is stored and organized
- Define the main data storage systems and databases on an IBM enterprise system
- Define IBM IMS™ TM, CICS®, IBM WebSphere® MQ and IBM WebSphere Application Server
- Identify the function of Job Entry Subsystem (JES)
- Identify how Job Control Language (JCL) controls the running of a program
- Code JCL statements to produce executable jobs

- Understand the role, function, and use of z/OS utility programs
- Effectively use the TSO/ISPF interface
- Develop proficiency in using the ISPF editor
- Use the ISPF utility functions
- Code REXX programs and the facilities of the language
- Define the basics for creating VSAM files
- Define and utilize system display and search facility (SDSF)
- Describe the characteristics of virtual storage
- Explain how address spaces are utilized
- Identify the concept of the Health Checker
- Define how the SMP/E process is used to install new load modules
- Recognize the components of the LOAD parameter
- Recognize the basic processes regarding the IPL procedure
- Identify the system parameter lists that will be used during system initialization
- Identify the types of information that must exist before the system initialization process begins
- Define the characteristics of JES2 spool data sets

Introduction to z/OS

Instructor-led training

- Introduction to z/OS: Data Management Systems Overview
- Documentation for z/OS

TSO/ISPF for z/OS

Course Type and Duration:

Web-based training
Instructor-led training
Self-paced, hands-on projects
Duration: 6 days

Course overview

Learning to use TSO/ISPF is a fundamental skill required for anyone working in the mainframe environment. TSO/ISPF provides an online environment for access to editing facilities and utilities. This course will teach you how to use the z/OS online tools to access services, edit data and manage data sets/files. The course is structured entirely around challenging project work, in which you will use TSO/ISPF to solve real-world problems.

You will learn how to:

- Access, add and edit system data sets (files)
- Use utilities facilities to manage data sets and exploit the tools for manipulating data
- Interact with other ISPF applications like SDSF to review job output
- Execute TSO scripts (CLIST and REXX) that allow you to execute predefined programs

Phase 1: Customize the TSO/ISPF environment

Web-based training

Introduction to TSO/ISPF
- Introduce concepts of the IBM Interactive
- System
- Productivity Facility (ISPF)
- Show methods used to navigate and utilize the common ISPFs

Using ISPF
- Locate and define the basic setting used to tailor the ISPF environment
- Use and control the ISPF split facility
- Define and use program function keys

Instructor-led training
- Introduction to ISPF and its capabilities
- Introduction to TSO
- ISPF menu navigation
- Use PF keys
- ISPF help system

Self-paced project work
- Introduction to ISPF and its capabilities
- Introduction to TSO
- ISPF menu navigation
- Use PF keys
- ISPF help system

Phase 2: Edit data sets with the TSO/ISPF editor

Web-based training

ISPF view and edit facilities
- View data sets using the ISPF View Facility
- Enter the Edit Facility in ISPF and navigate through a data set
- Use Edit Program function keys to assist in navigating an edited data set

ISPF edit line commands
- Insert and delete lines within a data set
- Copy, move and repeat lines in a data set
- Shift data within a line

ISPF edit primary commands for locating and changing data
- Use simple find and change commands and string patterns

ISPF edit primary commands for managing internal and external data
- Learn to use the UNDO facility
- Use CREATE/REPLACE to create or replace external members of a data set
- Use COPY/MOVE to copy or move data between external members

ISPF edit tabbing facility
- Learn to use hardware, software and logical tab controls

ISPF edit boundaries, masks and profiles
- Learn to set boundaries for EDIT operations and use EDIT profiles

Instructor-led training

Manipulating contents of data sets
- View and edit options
- Scroll commands
- Locate and change data
- Basic editing commands

Self-paced project work
- Edit the provided data set and make changes requested by the customer

Phase 3: Working with data sets

Web-based training

Manage data sets using ISPF data set utility

- Identify partitioned and sequential data sets
- Access the data set utility and view data set information
- Allocate, rename and delete data sets

Manage partitioned data sets using the ISPF library utility

- Print, copy, rename, and delete partition

The ISPF Copy, Search and Statistics Utilities

- Copy or move data sets or members of data sets
- Reset and delete ISPF statistics
- Search a data set or members of a data set for text entries

Manage data sets using the DSLIST utility

- Access the DSLIST utility and used data set patterns to display lists of data sets
- Identify and use the common DSLIST and TSO commands in a data set list
- Display the VTOC of specified volumes

Instructor-led training

- Introduction to data sets
- Data set utility 3.2
- Create a data set
- Allocation panel
- Library utility 3.1
- Move copy utility 3.3
- Data set list utility 3.4

Self-paced project work

- Allocate the new data set
- Copy contents of the data set
- Rename the data set
- Add the edited file as a member in the library

Phase 4: Invoking other programs

Web-based training

- None for this phase

Self-paced project work

- Use the necessary command to execute a predefined REXX routine
- Invoke ISRDDN to check whether the REXX routine is in the TSO/ISPF allocation

Instructor-led training

- TSO commands
- TSO Ready prompt
- Invoke a TSO Command and CLIST
- TSO command from ISPF

Phase 5: Advanced concepts: split screen, SUBMIT and other products

Web-based training

- None for this phase

Self-paced project work

- Edit a predefined job
- Submit job for execution
- Record message for job name and job number
- Invoke SDSF
- Display HELD output queue–Select output job

Instructor-led training

- Split screens
- Split screen mode
- SUBMIT command
- Invoke SDSF to review output

Job Control Language

Course Type and Duration

Web-based training
Instructor-led training
Self-paced, hands-on projects
Duration: 7 days

Course overview

JCL is used to run every unit of work in the system whether it be a batch job, started task or TSO user. This course will build your skills by teaching you how to use JCL to set up your job streams, create JCL procedures for common tasks and correct JCL errors.

You will learn how to:

- Use JCL to set up job streams to run your programs–Examine and correct JCL errors
- Create JCL procedures for common tasks
- Use JCL to create and reference data sets (files)
- Override JCL statements in procedures to make temporary changes
- Create JCL to run utilities or user written programs
- Identify and correct JCL errors in a job stream
- Set up multi-step JCL procedures and job streams
- Use generation data sets to make backup copies

Phase 1: Create the job to generate the list of states.	
Web-based training Introduction to z/OS JCL ▪ Work with the JOB statement ▪ Work with the EXEC statement ▪ Work with the DD statement	**Instructor-led training** ▪ Units of work in z/OS ▪ JCL syntax review ▪ JCL coding requirements ▪ Key JCL statements ▪ JOB statement parameters ▪ EXEC statement parameters ▪ Special input DD statements ▪ DISP parameter ▪ UNIT and VOL=SER parameters ▪ SPACE parameter ▪ Data set organization ▪ Logical record length ▪ Block size ▪ DCB parameter
	Self-paced project work ▪ Code the JOB statement ▪ Code the EXEC statement ▪ Code the DD statements ▪ Specify data set ▪ Specify SYSOUT ▪ Specify SYSIN ▪ Create several single-step jobs to accomplish various tasks

Phase 2: Create a job that can run on demand

Web-based training

Coding procedures and JES2 control statements

- Work with procedures
- Work with JCL symbols in procedures
- Use advanced control of output to JES
- Use JES2 control statements

Instructor-led training

- Execute a PROC
- Data set name considerations
- Review of JCL examples
- Override JCL statements
- Use symbolic variables

Self-paced project work

- Combine jobs
- Use temporary data sets
- Convert into JCL PROC

Phase 3: Create system-generated daily backups

Web-based training

Use special data sets in batch jobs

- Work with generation data groups
- Concatenate data sets and using dummy data sets
- Work with tapes
- Use temporary data sets
- Pass parameters and use backward references
- Use additional DD statement parameters

Instructor-led training

- Data set naming syntax
- Systems catalog
- Catalog entry
- Create generation data groups
- Relative generation number
- Retrieve generation data sets
- Concatenate data sets

Self-paced project work

- Define generation data groups
- Use JCL coding for relative generation numbers
- Use concatenation and other special data set usage

Phase 4: Maximize efficiency by cancelling the job when errors are present

Web-based training

Identify and resolve batch problems in JCL

- Use conditional JCL
- Use advanced conditional JCL Logic

Instructor-led training

- Condition codes vs ABEND codes
- Condition code testing
- Condition code examples
- Advanced conditional logic

Self-paced project work

- Handle condition code errors
- Handle conditional execution due to ABENDs
- Code the COND parameter
- Test condition code
- Use COND=EVEN; COND=ONLY
- Apply advanced conditional logic

Phase 5: Correct JCL errors

Web-based training

Identify and resolve batch problems in JCL

- Handle JCL errors and ABENDs

Instructor-led training

- Allocation processing
- JCL jobstream
- Interpret JCL listings
- Types of errors
- Identify ABEND conditions
- Restart a job step

Self-paced project work

- Interpret the output listing
- Handle errors and ABENDs
- Use step restarts

Utilities Curriculum

Course Type and Duration:

Web-based training
Instructor-led training
Self-paced, hands-on projects
Duration: 3 days

Course overview

Utility programs are tools that are available to help make routine tasks easier to do. Using these programs avoids the need to write specific programs to do routine tasks, and provides access to system facilities that might be too complex for routine programming. This course will build your skills by teaching you how to use utilities as a standardized means of performing general functions and the ability to secure selected program's access.

You will learn how to:

- Use the z/OS Utilities to manage data sets
- Copy, sort, and print files
- Manage data sets using z/OS facilitie

Phase 1: Data Set Utilities

Web-based training

- Data Utilities
- General Data Set Utilities

Self-paced project work

- Develop a REXX routine to read the data into memory and provide a mechanism so that the routine can be invoked for particular data
- Set up the I/O processes in a REXX program
- Build the tables to hold data
- Display data at terminal and taking queries from user
- Look up information and displaying results

Instructor-led training

- What Are Utilities
  - IEFBR14
  - IEBGENER
  - IEBCOPY
  - IEBPTPCH
  - IEBUPDTE
  - IEHLIST
  - LISTVTOC Example
  - IEHPROGM
  - IEBCOMPR
- Additional Utilities

Phase 2: Virtual Storage Access Method (VSAM) Utilities

Web-based training

Introduction to VSAM

- VSAM Basics
- Creating VSAM Data Sets
- Alternate Indexes, ALTER, and DELETE
- LISTCAT and PRINT
- Copying VSAM Data Sets

Self-paced project work

- Using IDCAMS to print the COUNTRY dataset
- Using IDCAMS to list all the data sets catalogs under your userid.

Instructor-led training

- Access Methods
- VSAM Files
- VSAM Terminology
- Special VSAM Usage
- IDCAMS
- System Catalog
- Non-VSAM Files
- VSAM File Definitions (KSDS)
- REPRO Sequential to VSAM Example
- VSAM Print Commands

Phase 3: The SORT/MERGE Utility

Web-based training

Data Utilities

- Introduction to DF SORT

Self-paced project work

- Sorting data in a data set
- Merging two data sets

Instructor-led training

- Sort Utility
- Sort JCL Requirements
- SORT General Control Statements
- SORT Exits / Internal Sorts

REXX Scripting Skills

Course Type and Duration:

Web-based training
Instructor-led training
Self-paced, hands-on projects
Duration: 4 days

Course overview

REXX is a commonly-used language and is portable to many platforms. This course will teach you to use basic REXX scripting skills to create programs, read and write data files and display data.

The course utilizes a blended approach, including web-based training, instructor-led sessions and hands-on lab exercises.

You will learn how to:

- Read and write records to and from a data set
- Manage a table of data in REXX
- Interact with the user in a query mode
- Code a conditional display of variables
- Code a PARSE PULL command

Phase 1: Creating a REXX program

Web-based training

Introduction to the REXX Programming Language

- Basic features of the REXX language
- Execute REXX programs in TSO/E
- REXX terms, variables and operators lower case

Instructor-led training

- Standard REXX
- REXX clauses
- Literal strings
- Simple variables
- Compound variables
- Operators
- External data queue
- Parse data
- Parse techniques
- Different types of loops
- Subroutes and functions
- Pass and return information
- Built-in functions
- Host environments

Self-paced project work

- Develop a REXX routine to read the data into memory and provide a mechanism so that the routine can be invoked for particular data
- Set up the I/O processes in a REXX program
- Build the tables to hold data
- Display data at terminal and taking queries from user
- Look up information and displaying results

Phase 2: The Rexx Parse command and conditional processing

Web-based training

None

Instructor-led training

- REXX PARSE command
- REXX conditional processing

Self-paced project work

- Code a conditional display of variables on the screen. Code a PARSE PULL logic.
- Execute the REXX routine and examine the results.

Phase 3: REXX advanced processing

Web-based training

None

Instructor-led training

- REXX subroutines and functions
- Basic REXX built-in functions
- Perform REXX I/O processing in the TSO environment

Self-paced project work

- Code Additional PARSE PULL Logics.

Systems Programming Curriculum

Course Type and Duration:

Web-based training
Instructor-led training
Duration: 3 days

You will learn how to:

- Concepts related to the z/OS operating system
- Examine the JES2 subsystem
- Introduce concepts associated with change management in z/OS [SMP/E]
- Review the role of SDSF in interacting with JES2
- Introduce IBM Health Checker for z/OS

Phase 1: z/OS system programming fundamentals

Web-based training

Introduction to the IBM enterprise environment

- Today's mainframe

z/OS systems programming fundamentals

- Work with system parameter lists
- Initialize the z/OS System

Instructor-led training

- Description of z/OS operating systems characteristics

Phase 2: Introduction to z/OS systems programming: SMP/E processing

Instructor-led training

- SMP/E overview
- Modification control statements
- SMP operation
- SMP/E database
- SMP/E JCL requirements
- SMP/E processing

Phase 3: Intro to z/OS systems programming - Health Checker

Instructor-led training

- Basic Health Checker display
- Health Checker detail
- Write your own health checks

Phase 4: Intro to z/OS systems programming - JES2

Instructor-led training

- JES2 startup and shut down
- JES2 processing phases
- SPOOL volumes
- JES2 Check Point and remote processing

Phase 5: Intro to z/OS systems programming - SDSF

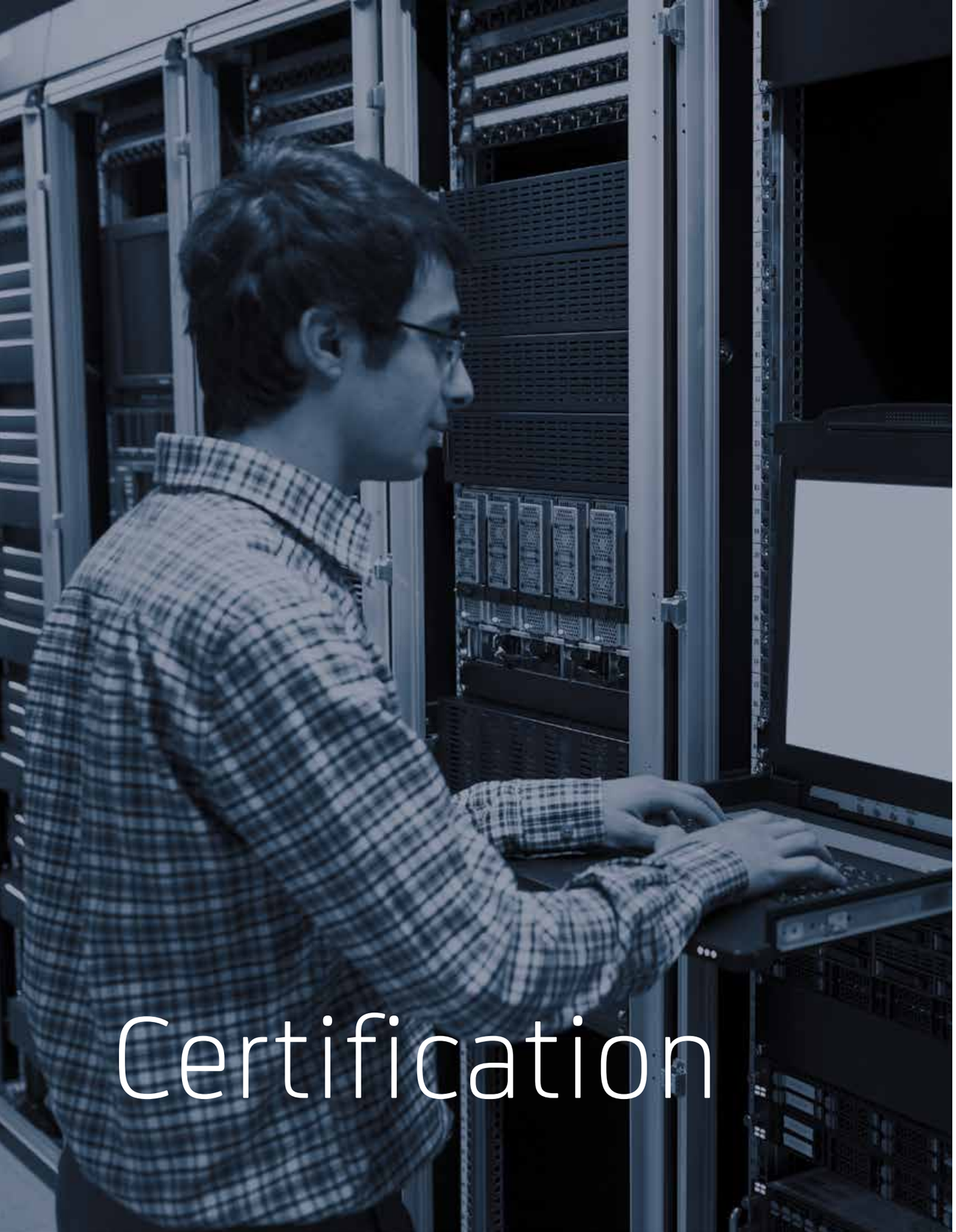
Instructor-led training

- Introduction to SDSF
- Input queue commands
- Actions and output
- SDSF edit
- SDSF JES2 resource commands

Phase 6: z/OS configuration options and subsystems

Instructor-led training

- Logical permissions
- Parallel Sysplex
- z/OS subsystems



CA Certified Mainframe Professional

Mainframe Academy is a certification program. It is proctored by a third party and establishes authoritative credentials at defensible costs for the investment. CA Technologies wanted Mainframe Academy to be certifiable, undisputable and ensure the highest standards are met. This program is vendor agnostic and as such, we needed rigorous testing. That is why when you earn the distinction of Certified Mainframe Professional by CA Technologies, it is recognizable. Being a certified mainframe professional is meaningful for CA Technologies, for you, for businesses and for the technical community.

At the conclusion of this program, some examples of what participants should be able to do:

- Perform basic programming and application development in z/OS.
- Issue job-level commands for z/OS job output.
- Partition z/OS in a real and virtual environment.
- Perform basic troubleshooting.
- Issue JES commands for batch jobs, monitor those jobs and analyze problems.
- Create, copy, delete and change data sets.
- Use TSO/ISPF to allocate and delete data sets.
- Navigate the ISPF menu structure.
- Use JCL statement syntax and format.
- Create and use JCL procedures.
- Issue SDSF system-level commands.
- Use basic REXX keyword commands.
- Identify REXX functions and their usage.