COBOL

Objetivos Gerais

Este curso é dirigido a todos os profissionais que gostaria de aprender o básico da linguagem COBOL. Ele fornece suficiente para compreender a linguagem de programação COBOL.

Objetivos Específicos

No final do curso os formandos ficarão aptos:

- A Contruir um Programa em COBOL.

Destinatários

A todos os profissionais que necessitem conhecer e aplicar o básico da linguagem COBOL

Pré-requisitos

Os pré-requisitos necessários para frequentar este curso são:

- Ter acesso a um computador ou um tablet com ligação à Internet e um browser (programa para navegar na web), como o Chrome, Safari, Firefox ou Internet Explorer.
- Pode aceder ao curso a partir de qualquer computador (por exemplo, em casa e no escritório), tablet ou smartphone.

Carga Horária

80 horas

Conteúdo Programático

MODULE 0 - Presentation of Platform and Method of Use

MODULE I - Introduction to COBOL Bootcamp; Various Administrative Topics; Introduction to Mainframe, TSO and ISPF Overview, TSO/ISPF Options 0 and 1

- Learning Goals:
- Set the agenda for the Mainframe Course;
- Describe basics of Mainframe Operating System, Hardware, Software and Networks;
- Explain how data is stored both physically and logically, and what fields, records and files are;
- Describe how the mainframe computer gets work done via JCL and JES;
- Explain the purpose of and differences between Sequential & Partitioned Data Sets;

- Know how to run basic TSO commands;
- Describe TSO and ISPF purpose, functions and modes;
- Be able to log on, log off and browse file;
- Introduction to Mainframe;
- Overview of JCL and JES
- Multiprogramming, multitasking, address spaces, paging
- Online vs. batch
- TSO Overview;
- Introduction to Native TSO, including special keys;
- Accessing Data on TSO;
- Using TSO Commands;
- ISPF Overview;
- ISPF Options;
- ISPF Browse;
- LABS (approximately 55 minutes):
- Log on, browse the catalog and some files;
- Navigate TSO, Execute TSO commands (e.g. LISTALC, LISTCAT, LISTDS);

MODULE II - ISPF Editing and Utilities

- Learning Goals
- Perform basic Edit functions, including line commands;
- Perform advanced Edit functions, including primary commands;
- Know how to allocate, delete, rename, and view catalog using ISPF Utilities;
- Create .COBOL and .JCL partitioned data sets;
- Lectures:
- Edit - Primary Commands;
- Edit - Line Commands;
- ISPF Utilities (e.g. Create, Copy, Delete, Move, List, Compare, Libraries)
- LABS (approximately 60 minutes) - Browse files, search for files, create .COBOL and .JCL partitioned data sets, create 2 simple members, Edit line commands, Edit primary commands.

MODULE III - Introduction to JCL, JOB and EXEC Statement

- Learning Goals:
- Know purpose of JCL;
- Understand what a job stream is;
- Code JOB and EXEC statements correctly;
- Lectures:
- Introduction to JCL;
- JOB Statement:
- EXEC Statement;
- LABS (approximately 50 minutes):
- Coding JOB statements at end of chapter;
- Building valid JOB statement on TSO;
- Coding EXEC statements at end of chapter;

MODULE IV - DD Statements

- Learning Goals:
- Allocate new and old datasets;
- Understand how to point files to printers and dummy devices;
- Understand the Data Control Block and associated JCL keywords (e.g. LRECL);
- Building a single step JOB on TSO and using the output viewing facility (SDSF);
- Assess Basic Mainframe Skills (TSO/ISPF, JCL) and Identify learning gaps (if any);
- Lecture:
- DD Statement;
- Review topics from first 4 days;
- LABS (approximately 60 minutes):
- Coding DD statements at end of chapter;
- Build and run 1 step job and view output;

MODULE V - Skill Assessment Results, Introduction to COBOL Development (Logic, Structure, Pseudocode, Test Plans)

- Learning Goals:

- Describe Sequence, Selection, Iteration;
- Use basic flowcharting symbols;
- Be able to code a simple flowchart;
- Be able to create a task analysis (pseudocode);
- Be able to create a Test Plan;
- Lecture - Skill Assessment School Solution and Review;
- Program Development;
- Structured and Top-Down Programming;
- Flowcharts, Structure Charts, Pseudo Code, and Test Plans;
- LABS - Logic Basics (approximately 90 minutes);
- “Logic 1” assignment from COBOL;
- Test Plan Workshop;

MODULE VI - COBOL Overview, Four Divisions

- Learning Goals:
- Explain general COBOL Syntax;
- Describe the compile and link process;
- Code the Identification, Environment, Data Division;
- Utilize picture clause, usage clause, and data categories withing a COBOL program, including differences between display, COMP-3 and binary (COMP) numbers;
- Lecture:
- COBOL Essentials;
- COBOL Overview;
- COBOL Definition;
- LABS (approximately 90 minutes):
- Create COBOL program with necessary definitions and simple logic;
- Identify numeric data in display, COMP-3 and binary format;
- Create and run “Hello World” COBOL program;

MODULE VII - COBOL Procedure Coding

- Learning Goals:
- Identify statements, sentences, scope terminators, paragraphs, sections;

- Compile, Link, and Test a simple COBOL program;
- Lecture:
- Simple Procedures;
- Explain COB01 Lab.
- LABS (approximately 70 minutes)
- COB01: Create a simple COBOL program and fix compile errors;

MODULE VIII - Data Editing

- Learning Goals:
- Explain end-of-file and loop processing;
- Utilize coding standards and good structure;
- Use switches properly within a program;
- Explain how IF/THEN/ELSE and EVALUATE work;
- Show how PERFORM statement works and why it is important;
- Lecture:
- Review COB01 Lab using Lab Solution;
- Editing and Branching, Explain COB10 Lab;
- LABS (approximately 140 minutes):
- Finish lab (COB01) from previous day; it is due today;
- Create a COBOL edit program (COB10) and run it successfully;

MODULE IX - Review and Assessment

- Learning Goals:
- Deepen understanding of COBOL;
- Assess Basic COBOL Skills, Identify learning gaps (if any);
- Review Session (approximately 60 minutes);
- Review topics from previous 4 days;
- Lecture:
- Structured COBOL;
- LABS (approximately 60 minutes):
- Additional remedial and/or “deep dive” labs

MODULE X - COBOL Reports - Part 1

- Learning Goals:
- Understand standard report structure (header, body, footer);
- Know how to use numeric editing masks;
- Code carriage control in the report layouts;
- Create WRITE AFTER ADVANCING statements;
- Lecture:
- COBOL Reports;
- Review COB10 program solution and Explain COB20 Lab
- LABS (approximately 120 minutes):
- Finish lab (COB10x) from previous day; it is due today;
- Create a COBOL report program (COB20x) and run it successfully;

MODULE XI - COBOL Reports - Part 2; Problem Solving & Debugging

- Learning Goals:
- Understand purpose of control breaks;
- Successfully code control break according to specifications;
- Understand testing strategies;
- Know available debugging tools for solving abends;
- Understand how to fix common programming errors;
- Lecture
- Review COBOL Reports;
- Review COB20 Lab with addition of control break requirement;
- Interpreting the Compiler Listing;
- Testing and Debugging;
- Common System Abend Codes;
- LABS (approximately 110 minutes):
- The Debugging Derby -Teams execute 6 programs that don’t work, debug, fix, and re-test them;
- Finish lab (COB20x) without control break:
- Begin making changes to COBOL report program (COB20x) for control breaks;

MODULE XII (7/13) - Arithmetic, Intrinsic Functions

- Understand basic arithmetic statements;
- Utilize counters in a program;
- ROUND arithmetic results;
- Lecture:
- COBOL Arithmetic, Intrinsic Functions;
- Explain COB30 Lab;
- LABS (approximately 120 minutes)
- Finish lab with control breaks (COB20x); it is due today;
- Modify program (creating COB30x) to include arithmetic calculations;

MODULE XIII- Report Control Breaks and Page Breaks

- Learning Goals:
- Know how to use control breaks, page breaks, grand totals;
- Lecture - Control breaks, Page breaks, Grand Totals:
- Review COBOL Reports chapter and discuss grand totals, Review COB30x program solution, Explain COB50 Lab;
- LABS (approximately 120 minutes):
- Finish lab (COB30x); it is due today;
- Modify COB30x (creating COB50x) to put control break logic in place (when STATE changes), page break logic in place (every 40 lines), grand totals in place;

MODULE XIV - Called Modules

- Learning Goals:
- Utilize module modularization with CALL, USING and LINKAGE;
- Understand differences between static and dynamic modules;
- Use CALL BY REFERENCE, CALL BY CONTENT, CALL BY VALUE, ON EXCEPTION, ON OVERFLOW;
- Lecture:
- Calling other programs;
- Explain CALL Lab;

- Pass at least GROSS-AMOUNT/SALARY and FED-TAX as variables;
- Determine how you will test this (but do not fully implement yet);
- LABS (Approximately 130 minutes):
- Modify COB30x to call COB40x for date and time (comment out your data/time logic in case you want to reuse this later on);
- Study COB40 to see how the time and date processing is done;
- Make this change to COB50 as well;
- If you are ahead of pace, contact instructor about COB15 specifications

MODULE XV - Table Processing I

- Learning Goals:
- Describe basic table terminology
- Load a table from an external file
- Lecture - Loading a Table:
- Table Structure;
- Table Handling – Loading a table;
- LABS (approximately 130 minutes):
- Write the first part of COB60x – loading the table using DISPLAYs to verify that every record has been loaded properly;
- Finish lab (revised COB30x); it is due today;

MODULE XVI - Table Processing II

- Learning Goals:
- Assess COBOL Skills;
- Use subscripts and indexes within a table;
- Code the SEARCH and PERFORM varying statements to search tables;
- Lecture:
- Table Handling – Searching the Table (Serial and Binary);
- LABS (approximately 60 minutes)
- Finish coding COB60x to successfully search the table.
- Finish lab (COB50x); it is due today;

MODULE XVII - Additional COBOL/JCL Topics: SMS, GDGs, Utilities

- Learning Goals:
- Use Generation Data Groups and Generation Data Sets
- Use Referbacks, Concatenated Datasets, and SMS JCL Parameters
- Run necessary IDCAMS to create GDG file
- Exploit IBM Utilities (IEBCOPY, IEBGENER, IEFBR14, SORT, IDCAMS)
- Lecture:
- Advanced Data Usage;
- JCL Utilities;
- LABS (approximately 140 minutes):
- Code 4-step job (IDCAMS, IEFBR14, IEBGENER, SORT);
- Finish lab (COB60x); it is due today;

MODULE XVIII - JCL Procedures

- Learning Goals:
- Convert a multi-step JCL to a Catalogued Procedure
- Use Instream and Cataloged Procedures
- Provide symbolic variables and overrides for JCL procs
- Understand how JCL from the catalogued procedure is merged with the run JCL
- Lecture:
- JCL Procedures (PROCs)
- LABS (approximately 150 minutes):
- Convert a 4-step job (IDCAMS, IEFBR14, IEBGENER, SORT) to a catalogued procedure;
- Examine the details of JCL merged from the procedure with the run JCL;
- Create JCL procedure to run COB10x and COB60x programs;
- Run JCL procedure with symbolic variables and overrides;

MODULE XIX - VSAM File Handling and IDCAMS

- Learning Goals:
- Ability to use FileAid or other screen utility to create VSAM and GDG datasets

- Provide IDCAMS commands for deleting and defining a VSAM dataset
- Run necessary IDCAMS to load VSAM KSDS file
- Code COBOL to search a VSAM file
- Lecture:
- Basic VSAM, IDCAMS;
- VSAM for COBOL Programmers - part one;
- Explain COB80 program - VSAM Search;
- LABS (approximately 140 minutes):
- Load data into VSAM KSDS file using IDCAMS and/or FileAid;
- Code COB80x, VSAM search program;
- Create JCL to run program COB80x;

MODULE XX - VSAM Updating

- Learning Goals:
- Understand how COBOL programs handle VSAM files;
- Describe differences between QSAM and VSAM;
- Code COBOL to update a VSAM file.
- Lecture:
- VSAM for COBOL Programmers – part two;
- Explain COB90 program – VSAM Update;
- LABS (approximately 120 minutes):
- Code COB90x, VSAM update program;
- Create JCL to run program COB90x;

MODULE XXI - Introduction to DB2, Basic SQL Data Manipulation

- Learning Goals:
- Describe the overall purpose of a database management system and its components;
- Describe the Relational Model of data using common relational terminology
- List the manipulative operations and give examples of each;
- List the data types supported by DB2 and describe their associated defaults;
- Define DB2’s support for missing information (nulls);
- Code restricts and projects in SQL using basic predicates and expressions;

- Become familiar with SPUFI for coding and running queries.
- Lecture:
- Relational Database Concepts;
- Basic Data Manipulation;
- SPUFI;
- LABS (approximately 60 minutes)
- Basic Data Manipulation;
- Knowledge Assessment - 60 minutes:
- Introduction to DB2 and SQL Review;

MODULE XXII - Intermediate Data Manipulation with SQL, Built-in Functions

- Learning Goals:
- Define the functionality of special registers in DB2;
- Code SQL statements using compound conditions with boolean operators;
- Code where-clause predicates utilizing special SQL keywords for restriction;
- State the difference between scalar functions and column functions;
- Define the basic families of string, numeric, and date/time functions;
- Code SQL operations that implement summarization via the GROUP BY clause and restrict groups as necessary with the HAVING clause;
- Code single-row and multiple-row INSERT statements.
- Lectures:
- Intermediate Data Manipulation;
- Built-in Functions;
- Aggregation;
- LABS (approximately 120 minutes)
- Intermediate Data Manipulation;
- Built-in Functions.

MODULE XXIII - Introduction to Embedded SQL

- Learning Goals:
- List the steps in program development;
- Relate DB2 datatypes to COBOL elementary items;

- Use SELECT...INTO for retrieving single rows into working storage variables;
- State the approach and methods associated with the DECLARE TABLE statement;
- Define and use the fields of the SQL communications area in programming situations;
- State the steps in program preparation and describe the inputs & outputs of each step;
- Describe the approaches available for binding packages and plans.
- Lecture:
- Introduction to Embedded SQL;
- Program Preparation;
- Using SELECT . . . INTO;
- LABS (approximately 130 minutes):
- Embedded SQL Workshop (Singleton Select).

MODULE XXIV - CICS Overview

- Learning Goals:
- Understand the general concepts and principles of CICS;
- Define CICS & its terminology (OLTP, Concurrency, LUW, Transaction, BMS, etc.);
- Discuss the flow of a CICS transaction through the various components of CICS;
- Be aware of the different CICS supplied transactions;
- Understand syntax of CICS commands;
- Define Mapsets, Maps, and Fields; Physical and Symbolic Maps; Steps involved in creating and compiling Maps;
- List the tables used to define CICS Resources and explain CICS Naming Conventions;
- Describe coding requirements and restrictions for COBOL CICS programs;
- Discuss program design considerations for CICS applications and the pseudo-conversational method of connecting CICS tasks;
- List options for passing application data between CICS tasks;
- Lectures:
- Chapter 1 - Introduction to CICS;
- Chapter 2 - CICS Architecture;
- Chapter 3 - Basic Mapping Support (SEND MAP, RECEIVE MAP);
- Chapter 4 - Programming Considerations;
- LABS (approximately 120 minutes)

- Learning to use CICS Supplied Transactions;
- Display a BMS map, navigate CECI, and make use of the CLEAR, RESET, and function keys;
- CEMT, CEDF, CEOT, CESF, CECI: ADDRESS, ASSIGN and SEND MAPS;
- Test a Map.

MODULE XXV - CICS Programming - Part 1

- Learning Goals:
- Define the main concepts of VSAM data set organization, and the use of free space in adding records to and deleting records from a KSDS;
- Invoke access method services (IDCAMS) to (1) Print VSAM and non-VSAM data sets, (2) Print Catalog Entries, (3) Delete and Define VSAM data sets, (4) Load VSAM data sets, (5) Backup/Recover VSAM data sets, (6) Interpret control information from VSAM catalogs;
- Explain the relationship of data sets to control intervals and control areas
- Write COBOL CICS programs which process VSAM KSDS files and invoke CICS services;
- Understand the contents of the Execute Interface Block (EIB);
- Understand how to deal with anticipated and unexpected exceptional conditions (results of command requests);
- Know how to interface directly with the Execute Interface Program through the Command Level Interpreter (CECI);
- Transfer control between CICS programs, both at the same level and subordinate level, using LINK, XCTL, RETURN, and LOAD;
- Know how and why to return control to CICS, and abend a CICS program
- Understand how to use a Communication Area;
- Use the Execution Diagnostic Facility to interactively test a CICS application (CEDF);
- Know how to send and receive simple messages from the program to the terminal;
- Utilize both random and sequential file processing options with VSAM files;
- Know how to be aware of and avoid the possibility of a deadlock.
- Lectures:
- Chapter 5 - VSAM Essentials for CICS;

- Chapter 6 - Command Level Interface;
- Chapter 7 - Program Control;
- Chapter 8 - Terminal Control;
- LABS (approximately 110 minutes)
- Use CEMT command to retrieve file and program information;
- Create and run a “Hello World” CICS program, including navigating the compile, link, and CEMT process;
- Create and load data into a VSAM KSDS data set;
- Compile a program (COB92) to replace the VSAM update program (COB90). Instead of using an input file, the program will get information from a terminal map, and be able to browse records in the STATETAX file;
- Change program (COB92) to include functionality to make changes to state tax information, insert new records, and delete state tax records.

MODULE XXVI - CICS Programming - Part 2 - and Resources

- Learning Goals:
- Understand the concepts and use of Temporary Storage (TS)
- Understand the concepts and usage of Transient Data (TD)
- Explain the critical elements of using relational databases
- Use the appropriate method to access DB2 databases from CICS programs
- Show the similarity between the SQLCA and the EIB
- Tell which CICS table contains DB2 plan information
- Define Logical Unit of Work (LUW) and know how to protect CICS resources
- Understand how to create, retrieve, and delete Temporary Storage queues
- Create, retrieve, and delete Transient Data services
- Lectures:
- Chapter 9 - File Control;
- Chapter 11 - Additional Services (Temporary Storage, Transient Data);
- Chapter 12 - DB2 Interface;
- LABS (approximately 120 minutes)
- Use CECI to browse files with the STARTBR command

- Continue changing program (COB92) to include functionality to make changes to state tax information, insert new records, and delete state tax records

MODULE XXVII - CICS Review and Assessment; Case Study Presentations

- Learning Goals:
- Assess CICS Skills; Identify learning gaps (if any)
- Lecture:
- Review topics from previous 3 days;
- LABS (approximately 120 minutes:
- Continue changing program (COB92) to include functionality to make changes to state tax information, insert new records, and delete state tax records.

MODULE XXVIII - Union, Data Modification, Additional Concepts in Data Manipulation with SQL, Case Study Presentations

- Learning Goals:
- Define basic UNION operations and differentiate between UNION and UNION ALL
- State the requirement for self-joins and the procedure to accomplish them in SQL
- Describe the operation of the outer join and define their relationship to inner joins
- Use the EXISTS quantifier with correlated subqueries and compare its operation to non-correlated subqueries using IN
- Code searched update and deletes that affect one row, many rows, or all rows of a table
- Lectures:
- Set Operations (UNION, INTERSECT & EXCEPT);
- Data Modification.
- LABS (approximately 60 minutes):
- Set Operations (UNION, INTERSECT & EXCEPT);
- Data Modification;

Metodologia

Este curso tem sempre presente o formador, que irá mesmo dar a formação presencial através da plataforma.

O Formando pode intervir juntamente com o formador ou com os restantes formandos tal como faz na sala de aula.

As apresentações e exercícios serão sempre disponibilizados pelo formador no final de cada sessão de formação.

No final do curso receberá um Certificado de Formação Profissional caso frequente pelo menos 90% das aulas, realize os trabalhos e os testes propostos, participe nas discussões online e tenha avaliação final positiva.

Esta formação é certificada e reconhecida.