

FORMATION

DEVELOPPEMENT MAINFRAME

COBOL/ PACBASE

Durée 399 heures – 57 jours

Auteur : SSE. 01/2020

Les objectifs de cette formation Mainframe

- **Savoir développer dans des environnements des Grands Systèmes IBM, COBOL, CICS, DB2**
- **Manipuler et gérer les fichiers batch et VSAM,**
- **Maîtriser la syntaxe globale du langage Cobol, et SQL DB2**
- **Comprendre et mettre en œuvre les liens entre programmes et sous-programmes**
- **Maîtriser les impressions**
- **Maitriser un projet depuis l'organisation jusqu'à la mise en production en passant par la conduite de changement.**
- **Analyser les problématiques, les méthodes, les techniques et les rôles associés au développement des systèmes d'information d'aujourd'hui**

Public : Jeunes diplômés scientifiques (Bac+5), développeurs débutants.

Modalités d'évaluation : Chaque module fera l'objet d'exercices pratique et mises en situation. Des entretiens réguliers avec le tuteur permettront d'évaluer l'acquisition des connaissances via les TP et QCM

Intervenant : Formateurs experts en langage de programmation et développement Mainframe Cobol

Lieu : 9-11 av de la Libération 94100 St Maur des fossés

Moyens : Ordinateurs / PC, vidéo projecteur, Paperboard, Supports pédagogiques

Coût : 18€ht/heure/stagiaire

Sanction de la formation : Attestation de formation, Certificat de réussite

Participants : 8 à 15 stagiaires

Modalités et délais d'accès : nous contacter au 01 43 53 84 43

Accessibilité : toutes nos salles sont accessibles aux personnes à mobilités réduites

Avant-propos - 0.5 jour

Présentation de la formation

- Tour de table**
- Objectif de la formation**
- Connexion au site de formation**

Origines

- Les computers**
- Un langage de programmation**
- Pourquoi COBOL ?**
 - **Spécifications**
 - **Normalisation**

COBOL dans le temps

- COBOL 74**
- COBOL 85**
- COBOL 89**
- COBOL 2002**

COBOL aujourd'hui

- Environnement mainframe**
- Autres environnement**
- Traitements batch**
- Autres traitements**

Concepts de bases

- Rappel**
- Les étapes du développement d'un programme**

Présentation (chapitre 1) - 0,5 jour

Structure d'un programme

Formalisme

- La division IDENTIFICATION**
- La division ENVIRONNEMENT**
- La division DATA**
- La division PROCEDURE**

Mise en oeuvre : compilation

Cas pratiques (exercice)

- Environnement de programmation COBOL sur le site de formation**
- Le programme complet**

DB2– Les requêtes –5 jours

Environnement

Environnement SPUFI sur le site de formation (requêtes DB2)

SQL

- La commande SELECT
- Les opérateurs
- Les clauses
- Les prédicats
- Les fonctions Built-in et Scalaires
- Les sous-requêtes et les corrélations
- Les jointures

Exercices SQL (requêtes)

Déclarer et manipuler les données (chapitre 2) - 7 jours

Introduction

Codes de caractères : EBCDIC

Déclaration des données

- Clause PICTURE
- Zones non utilisées : FILLER

Caractéristiques de la WORKING-STORAGE SECTION

- Constantes : clause VALUE
- Littéraux alphanumériques
- Littéraux numériques
- Constantes figuratives

Numéros de niveaux hiérarchiques

Présentation

- Niveau 01
- Niveaux 02 à 49

Niveaux hiérarchiques spéciaux

- Niveau 66
- Niveau 88
- Manipulation des noms de condition : instruction SET

Zones alphanumériques : déclaration et usage

MOVE avec zone réceptrice alphanumérique

- Clause JUSTIFIED RIGHT

- Notation hexadécimale
- Valeur nulle en fin de zone

Modification de référence

Analyse de zone

- Pour compter : TALLYING
- Pour remplacer : REPLACING
- Pour convertir : CONVERTING
- BEFORE / AFTER

Concaténation : STRING

Découpage : UNSTRING

- DELIMITER et COUNT

Les zones numériques

USAGE DISPLAY

Numérique compacté

Binaire

Binaire natif : COMP-5

MOVE avec zone réceptrice numérique

Calculs

- Addition : ADD
- Soustraction : SUBTRACT
- Multiplication : MULTIPLY
- Division : DIVIDE
- Expression arithmétique : COMPUTE

Performances vs sécurité

Condition ON SIZE ERROR

Conversion automatique

Les conditions de signe et de classe

Les conditions de signe

Les conditions de classe

Expression CORRESPONDING

Les registres spéciaux

WHEN-COMPILED

RETURN-CODE

LENGTH OF

ADRESS OF

Clause REDEFINES : masque de donnée

Initialisations : INITIALIZE

Tableaux : clause OCCURS

Déclarations

- Tableau à plusieurs dimensions

- Tableau de longueur variable
- Tableau trié
- Indice vs Index

Initialisation

- Indice
- Index
- Tableau de constantes
- Tableau de données
- Charger un tableau de données
- Position relative

Recherche dans un tableau

- SEARCH : recherche séquentielle

Première approche (chapitre 3) - 5 jours

Programmation structurée

Le déroulement d'un programme (organigramme)

Les commentaires

L'aide au débogage

Les mots en COBOL

Instructions de structure

- Sections et paragraphes
- Modules

PERFORM : les débranchements

- Appel de paragraphe
- Appel de section
- Appel de plusieurs paragraphes ou sections
- Débranchement par un GO TO

Contrôle de séquence : fin de module

- EXIT, pour la fin d'une section
- Particularité du module directeur
- Utilité des sections

Fin de programme

- STOP RUN : pour programme batch principal
- EXIT PROGRAM : pour la fin d'un sous-programme
- GOBACK : pour programmes et sous-programmes

Traitement itératif

PERFORM UNTIL : boucle jusqu'à

Variantes du PERFORM : PERFORM en ligne

PERFORM VARYING : boucle pour

- Boucles imbriquées : **PERFORM... VARYING... UNTIL... AFTER...**
- **PERFORM** : option **WITH TEST AFTER**

PERFORM n TIMES : boucle répéter

Instructions de tests

IF THEN ELSE : la condition simple

- Points vs **SCOPE TERMINATOR**
- Les opérateurs
- Les combinaisons
- Les imbrications

EVALUATE : la condition à choix multiple

- Evaluation d'une zone
- Evaluation booléenne
- Evaluations multiples
- **CONTINUE** : conditions et débranchement
- **NEXT SENTENCE** : autre branchement

MOVE : manipulation des données

Cas pratiques (exercices)

Les organigrammes

Exercice (suite aux organigrammes)

Identifier les grandes parties

- Repérage
- Algorithme de traitement
- Amélioration du code : simplifier
- Amélioration du code : pas de code mort

Techniques avancées (chapitre 4) - 5 jours

Code réutilisable

Modules COPY

- Pour les déclarations
- Pour les traitements
- Plusieurs fois dans le même programme

Sous-programmes

- Appel de sous-programme : **CALL**
- Le passage des paramètres
- Sous-programmes et **RETURN-CODE** : **RETURNING**
- **LINKAGE SECTION**

- **REFERENCE vs CONTENT**
- **Application : COBOL et l'allocation dynamique**
- **Compléments**
- **Soumettre un JCL**

Mise en forme des données

Fonctions intrinsèques

- **Quelques fonctions**

Cas particulier : les dates

- **Les instructions COBOL « historiques »**
- **Les fonctions intrinsèques pour les dates**

Les états

- **Déclarations : les PICTURE d'édition**
- **Mise en forme**
- **Traitement**

DB2 en COBOL (chapitre 5) - 10 jours

Bases de données DB2

Contenu de la SQLDA

Declare table

Les Hosts Variables

Zone NULL

Les instructions d'entrée-sortie

Exercices SQL en COBOL

Lecture séquentielle

Lecture directe

Fonctions Built-in

Jointure directe (non applicative)

Exercices complémentaires

CICS - 10 jours

Objectif :

- Décrire le rôle de CICS dans un environnement transactionnel métier et son support des interfaces modernes Identifier les avantages métiers majeurs de CICS TS V5.3
- Décrire le flux d'une transaction CICS typique
- Décrire les fonctions que les transactions CICS primaires réalisent
- Utiliser CICS master terminal transaction (CEMT) pour interroger et contrôler les ressources systèmes
- Décrire les diverses façons de définir des ressources dans CICS
- Utiliser des transactions que CICS fournit pour définir et gérer des ressources
- Expliquer comment CICS stocke et gère les définitions des ressources
- Décrire les fonctionnalités, les avantages et l'usage de CICS et z/OS Explorer Utiliser CICS Explorer pour se connecter, se déplacer et gérer les ressources
- Faire une liste des ressources récupérables de CICS et la façon dont elles fournissent l'intégrité des données
- Décrire une transaction CICS backout et forward
- Décrire les utilitaires que CICS fournit pour maintenir l'intégrité des données
- Décrire le support CICS pour les bases de données, pour les fichiers Virtual Storage Access Method (VSAM), pour les files d'attente et IBM MQ
- Utiliser les transactions CICS et CICS Explorer pour créer et explorer les files d'attente CICS, les fichiers et les ressources des bases de données
- Décrire comment CICS implémente la sécurité, les types de sécurité et ce qu'il utilise pour l'identification et l'authentification des utilisateurs
- Faire une liste des types de démarrage et d'arrêt de CICS
- Faire une liste des utilitaires que CICSplex System Manager fournit pour gérer les topologies CICS et son rôle dans la mise en place du Cloud
- Décrire les fonctions de présentation, de business et de logique d'accès aux données
- Définir le rôle et l'utilisation de COMMAREA et CHANNEL dans les applications CICS
- Montrer comment CICS Basic Mapping Support (BMS) supporte l'affichage 3270
- Décrire les différents types de conception d'application CICS et les architectures de programmation supportées pour les langages procéduraux et les langages orientés objets Décrire comment CICS implémente le support de web services pour les services SOAP et JSON
- Utiliser CICS Explorer pour visualiser les web services de CICS
- Utiliser Rational Developer for System z pour visualiser la cartographie des ressources
- Décrire la syntaxe d'une commande CICS et comment gérer le traitement des exceptions
- Utiliser les transactions que that CICS fournit pour les tests et les débogages
- Décrire les utilitaires qui sont disponibles dans CICS pour les intercommunications

Projet – Evaluations - 8 jours

Mise en mode projet

Cette période d'immersion vise à permettre au stagiaire notamment :

- de mettre en pratique en environnement concret l'apprentissage théorique,
- de confirmer ses nouvelles compétences,
- d'appréhender les contraintes réelles d'un projet.

Visualage Pacbase V3.5 Terminal Passif Iso Fonctionnalité V2.5 Modules

Batch 6 jours + 1 jour évaluations Pacbase

Objectif :

- **appréhender la démarche de construction d'un programme Pacbase**
- **comprendre le principe des macro-structures.**
- **comprendre le raisonnement à mener pour insérer des traitements 'spécifiques' dans un programme, et la syntaxe du langage de réalisation, propre à Pacbase, permettant d'écrire ces traitements**
- **Découvrir la démarche de maquettage d'un écran**
- **Mettre en place l'aide en ligne d'une application TP**
- **Mettre en oeuvre les étapes de la mise en place des traitements conversationnels associés (standard, Macro-Structures, traitements spécifiques), en respectant l'architecture standard**
- **générée**
- **Réaliser la constitution des libellés d'erreurs**

Concepts et Fondamentaux PACBASE

Le développement industriel aujourd'hui

Principes généraux de PACBASE (Référentiel, méta-modèle, générateurs)

La base de spécifications

Le générateur

La documentation

Les bibliothèques

Les entités PACBASE

Les références croisées

L'architecture des traitements

L'histoire de PACBASE

Dictionnaire

L'accès à PACBASE (environnement et utilisation)

Présentation du dictionnaire (entité, langage de commande)

Le méta-modèle

Dossier d'entité

L'entité Rubrique

L'entité Mot-clé (recherche par mot-clé)

TP -Exercices pour : une rubrique, une structure de donnée, un segment, toutes les entités Pacbase

L'entité structure de données

L'entité segment

TP -Exercices sur lessegments

Création de structure de données

Création de segments

Les principes de la documentation

L'entité texte (création, consultation, ventilation)

Le Manuel utilisateur (création, génération)

TP -Saisie et ventilation de textes associés et documentaires

Langage Structuré

L'architecture d'un programme Pacbase

L'entité Programme

Les éléments du dossier de Programme