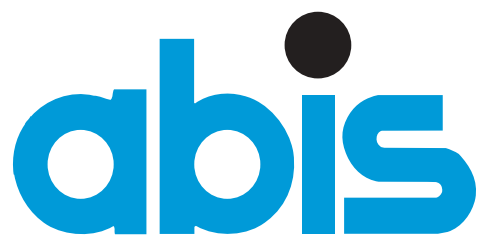


Auto-évaluation : La programmation en COBOL - 1e partie

Document: f1290test.fm

05/01/2016

ABIS Training & Consulting
P.O. Box 220
B-3000 Leuven
Belgium



TRAINING & CONSULTING

INTRODUCTION AUTO-ÉVALUATION : LA PROGRAMMATION EN COBOL - 1E PARTIE

Afin que vous puissiez évaluer vos connaissances en COBOL, nous vous proposons d'effectuer ce test. Ce test porte sur les sujets qui sont vus dans le cours [La programmation en Cobol - 1e partie](#).

Ce test comporte 20 questions à choix multiples. Pour certaines des questions, il est indiqué que plusieurs réponses correctes sont possibles. Dans ce cas là, une réponse est considérée comme correctes lorsque toutes les propositions correctes sont sélectionnées.

Il n'y a pas de "question piège", cependant, lisez attentivement les questions et les propositions. Comptez environ 15 minutes pour répondre à toutes les questions.

Vous trouverez les réponses correctes ainsi les instructions relatives à l'évaluation de ce test à la fin de ce document.

QUESTIONS AUTO-ÉVALUATION : LA PROGRAMMATION EN COBOL - 1E PARTIE

1. Lequel de ces termes n'est PAS un mot réservé en COBOL?
- (a) RECORD
 - (b) INDEX
 - (c) VARIABLE
 - (d) FILE
2. Lequel de ces termes NE PEUT PAS être utilisé pour exécuter des calculs dans un programme COBOL?
- (a) ASSIGN
 - (b) DIVIDE
 - (c) MULTIPLY
 - (d) ADD
3. Un programmeur COBOL désire qu'une instruction soit répétée 5 fois. Comment peut-il coder cela?
- (a)

```
REPEAT 5 TIMES
    instruction
END-REPEAT
```
 - (b)

```
PERFORM 5 TIMES
    instruction
END-PERFORM
```
 - (c)

```
MOVE 1 TO COUNTER
PERFORM UNTIL COUNTER = 5
    COMPUTE COUNTER = COUNTER + 1
    instruction
END-PERFORM
```
 - (d)

```
REPEAT VARYING COUNTER FROM 1 BY 1 UNTIL COUNTER = 5
    instruction
END-PERFORM
```

4. Laquelle de ces réponses donne l'ordre correct des divisions dans un programme COBOL?
- (a) Environment - Identification - Data - Procedure
 - (b) Identification - Environment - Data - Procedure
 - (c) Identification - Environment - Procedure - Data
 - (d) Environment - Identification - Procedure - Data
5. Quels noms de variable de données sont syntaxiquement corrects? (2 réponses)
- [a] CITY-CODE
 - [b] -NR-OF-ITEMS
 - [c] TOTAL-NUMBER-OF-LINES-PER-TEST
 - [d] PRODUCT NUMBER
 - [e] INDEX
6. Où est-on obligé, dans un programme COBOL, de coder un point (".")? (3 réponses)
- [a] après chaque déclaration de variable de niveau 01
 - [b] à la fin de chaque instruction
 - [c] à la fin de chaque paragraphe
 - [d] à la fin du programme
 - [e] pour terminer une instruction IF
7. Lequel de ces symboles NE PEUT PAS être codé à la position 7 dans un programme COBOL?
- (a) -
 - (b) *
 - (c) /
 - (d) #

8. Considérez le programme ci-dessous. Supposez que A,B, C, ... sont des instructions COBOL. Dans quel ordre, les instructions vont-elles être exécutées?

```
MAIN-PAR.  
  PERFORM INIT-PAR  
  Z  
  PERFORM TERM-PAR  
  Z  
  STOP RUN  
.  
INIT-PAR.  
  A  
  B  
  PERFORM READ-X  
.  
LOOP-PAR.  
  E  
  F  
.  
TERM-PAR.  
  G  
  H  
.  
READ-X.  
  X.
```

- (a) A B Z E F Z X
- (b) A B X Z E F G H Z X
- (c) A B X Z E F Z G H Z
- (d) A B X Z G H Z
9. Observez le calcul suivant. En tenant compte des règles de priorité, comment ce calcul va-t-il être exécuté?

$A ** 2 * B / 2 + c - 5$

- (a) $((A ** 2) * B) / 2 + c - 5$
- (b) $((A ** 2) * (B / 2)) + c - 5$
- (c) $(A ** 2) * (B / (2 + c)) - 5$
- (d) $((A ** (2 * B)) / 2) + c - 5$

10. Un programmeur COBOL désire afficher un nombre suivant l'exemple ci-dessous. Quelle définition de variable est-elle adaptée à cela?

+123.45

(Il est supposé que le signe décimal est le '.')

- (a) PIC 9(5)
 - (b) PIC +9(3).99
 - (c) PIC +9(3)V9(2)
 - (d) PIC 9(3).V9(2)
11. Observez les déclarations de variables suivantes. Quel sera, à la suite de toutes les instructions MOVE, le contenu de la variable RESULT?

```
01 VAR-1      PIC X(5) VALUE 'PL/I'.  
01 VAR-2      PIC X(10).  
01 VAR-3      PIC X(2).  
01 RESULT     PIC X(5).
```

```
MOVE 'COBOL' TO RESULT  
MOVE VAR-1 TO VAR-2  
MOVE VAR-2 TO VAR-3  
MOVE VAR-3 TO RESULT
```

- (a) PL____ (où ____ représente 3 blancs)
 - (b) COBOL
 - (c) COBPL
 - (d) PLBOL
12. Lesquelles parmi ces déclarations sont correctes?

```
(1) 01 VAR-1      PIC SX(4).  
(2) 01 VAR-2      PIC S9(4).  
(3) 01 VAR-1      PIC V99.  
(4) 01 VAR-1      PIC P99.  
(5) 01 VAR-1      PIC 9(4)S.  
(6) 01 VAR 1      PIC X(4).
```

- (a) 2, 3 et 6
- (b) 3, 4 et 6
- (c) 1, 2 et 3
- (d) 2, 3 et 4

13. Quelles définitions PIC peuvent-elles être utilisées pour la description d'un fichier séquentiel qui va être utilisé en INPUT (3 réponses)

- [a] PIC X(9)
- [b] PIC S9(4) COMP
- [c] PIC Z(4)
- [d] PIC 9(4)V99
- [e] PIC 9(4).99

14. Observez la variable suivante. Quels tests peut-on faire avec celle-ci? (2 réponses).

```
01 END-OF-FILE          PIC X.  
   88 EOF                VALUE 'N'.
```

- [a] IF EOF THEN ...
- [b] IF EOF IS TRUE THEN ...
- [c] IF END-OF-FILE = 'Y' THEN ...
- [d] IF END-OF-FILE THEN ...

15. Supposez qu'on désire imprimer un nombre suivant le format ci-dessous. Comment doit-on déclarer la variable exécution dans le programme COBOL?

```
contenu:      00232^45 négatif (le ^ est considéré comme signe décimal)  
exécution:   $___232.45-      ( où ___ représente 3 blancs)
```

- (a) PIC \$Z(6).99-
- (b) PIC \$(7).99-
- (c) PIC \$Z(6)V99-
- (d) PIC -\$Z(6).99

16. Laquelle parmi ces instructions COBOL EVALUATE n'est-elle syntaxiquement PAS correcte?

(a)

```
EVALUATE A
  WHEN 0          DISPLAY 'A = 0'
  WHEN 5          DISPLAY 'A = 5'
  WHEN OTHER     DISPLAY 'A IS NOT 0 NOR 5'
END-EVALUATE
```

(b)

```
EVALUATE TRUE
  WHEN A=0        DISPLAY 'A = 0'
  WHEN A=5        DISPLAY 'A = 5'
  WHEN OTHER     DISPLAY 'A IS NOT 0 OR 5'
END-EVALUATE
```

(c)

```
EVALUATE A
  WHEN A=0        DISPLAY 'A = 0'
  WHEN A=5        DISPLAY 'A = 5'
  WHEN OTHER     DISPLAY 'A IS NOT 0 NOR 5'
END-EVALUATE
```

17. Observez le code COBOL suivant. Quand l'instruction instruction-3 va-t-elle être exécutée?

```
IF A=B THEN
  instruction-1
ELSE
  instruction-2.
  instruction-3
```

(a) jamais

(b) si A et B sont égaux

(c) si A et B sont différents

(d) toujours

18. Pour chaque fichier que l'on veut lire dans un programme COBOL, il faut coder un certain nombre de déclarations et d'instructions. Dans quel ordre, les lignes de code suivantes doivent-elles être placées?

- (1) SELECT infile ASSIGN TO xx
- (2) READ infile
- (3) OPEN INPUT infile
- (4) CLOSE infile

- (a) 3, 1, 2, 4
- (b) 1, 3, 2, 4
- (c) 1, 2, 3, 4
- (d) 3, 2, 1, 4

19. Lesquelles parmi ces affirmations, en relation avec les fichiers, sont-elles vraies? (2 réponses)

- [a] l'entrée FD dans la DATA DIVISION donne une description de la structure du fichier.
- [b] lors de l'ouverture d'un fichier, le premier enregistrement est immédiatement lu.
- [c] lorsqu'un fichier est ouvert en mode I-O mode, on peut faire aussi bien des écritures que des lectures.
- [d] la lecture de fichiers indexés est impossible à partir d'un fichier COBOL.
- [e] dans la clause SELECT, il convient de décrire l'endroit exact où se trouve le fichier.

20. Quelle est la valeur de RESULT après l'exécution de l'instruction COBOL suivante si la valeur de départ de RESULT est 0?

```
PERFORM VARYING C FROM 1 BY 1 UNTIL C = 6
      COMPUTE RESULT = RESULT + 1
END-PERFORM
```

- (a) 0
- (b) 5
- (c) 6
- (d) 7

EVALUATION.

Ici sont les réponses correctes:

1. c
2. a
3. b
4. b
5. a c
6. a c d
7. d
8. d
9. a
10. b
11. a
12. d
13. a b d
14. a c
15. a
16. c
17. d
18. b
19. a c
20. b

Comptez 1 point par bonne réponse. Pour les questions avec plusieurs bonnes réponses, comptez 1 point seulement si vous avez coché toutes les bonnes alternatives et aucune mauvaise.

Si votre score est inférieur à 50%, il vous est vivement conseillé de suivre le cours [La programmation en Cobol - 1e partie](#).

Si votre score est entre 50% et 80%, vous avez déjà certaines connaissances sur le sujet. Malgré tout, le cours [La programmation en Cobol - 1e partie](#) vous permettra d'apprendre encore bien des choses utiles.

Si votre score atteint 80% ou plus, il ne vous est pas nécessaire de suivre ce cours. Vous êtes donc prêt pour le cours [La programmation en Cobol - 2e partie](#).