

# COBOL INTERVIEW QA

## 1. What is a reentrant program? How will you make a program Reentrant?

A. A reentrant program, while being executed resides in the common virtual area so that one copy of it may be shared among all callers. Use RENT compiler option.

## 2. What is the difference between a 01 level and 77 level?

A. 01 level can have sublevels from 02 to 49. 77 cannot have sublevel.

## 3. What are the few advantages of VS COBOL II over OS/VS COBOL?

A. The working storage and linkage section limit has been increased. They are 128 megabytes as supposed to 1 megabytes in OS/VS COBOL. Introduction of ADDRESS special register.

31-bit addressing. In using COBOL on PC we have only flat files and the programs can access only limited storage, whereas in VS COBOL II on M/F the programs can access up to 16MB or 2GB depending on the addressing and can use VSAM files to make I/O operations faster.

## 4. What are the steps you go through while creating a COBOL program executable?

A. DB2 pre-compiler (if embedded SQL is used), CICS translator (if CICS program), Cobol compiler, Link editor. If DB2 program, create plan by binding the DBRMs.

## 5. What are the minimum requirements to compile a program without errors?

A. Identification Division.

Program-ID. Program-name.

A) Is compute w=u a valid statement?

Yes, it is. It is equivalent to move u to w.

B) In the above example, when will you prefer compute statement over the move statement?

When significant left-order digits would be lost in execution, the `COMPUTE` statement can detect the condition and allow you to handle it. The `MOVE` statement carries out the assignment with destructive truncation. Therefore, if the size error is needs to be detected, `COMPUTE` will be preferred over `MOVE`. The `ON SIZE ERROR` phrase of `COMPUTE` statement, compiler generates code to detect size-overflow.

## 6. What happens when the `ON SIZE ERROR` phrase is specified on a `COMPUTE` statement?

A. If the condition occurs, the code in the `ON SIZE ERROR` phrase is performed, and the content of the destination field remains unchanged. If the `ON SIZE ERROR` phrase is not specified, the assignment is carried out with truncation. There is no `ON SIZE ERROR` support for the `MOVE` statement.

## 7. How will you associate your files with external data sets where they physically reside?

A. Using `SELECT` clause, the files can be associated with external data sets. The `SELECT` clause is defined in the `FILE-CONTROL` paragraph of Input-Output Section that is coded Environment Division. The File structure is defined by `FD` entry under File-Section of Data Division for the OS.

## 8. How will you define your file to the operating system?

A. Associate the file with the external data set using `SELECT` clause of `INPUT-OUTPUT SECTION`. `INPUT-OUTPUT SECTION` appears inside the `ENVIRONMENT DIVISION`. Define your file structure in the `FILE SECTION` of `DATA DIVISION`.

#### 9. Explain the use of Declaratives in COBOL?

Declaratives provide special section that are executed when an exceptional condition occurs. They must be grouped together and coded at the beginning of procedure division and the entire procedure division must be divided into sections. The Declaratives start with a USE statement. The entire group of declaratives is preceded by DECLARIVES and followed by END DECLARITIVES in area A. The three types of declaratives are Exception (when error occurs during file handling), Debugging (to debug lines with 'D' coded in w-s section) and Label (for EOF or beginning...) declaratives.

#### 10. A statically bound subprogram is called twice. What happens to working-storage variables?

A. The working-storage section is allocated at the start of the run-unit and any data items with VALUE clauses are initialized to the appropriate value at the time. When the subprogram is called the second time, a working-storage items persist in their last used state. However, if the program is specified with INITIAL on the PROGRAM-ID, working-storage section is reinitialized each time the program is entered.

PROGRAM-ID. <Pgmname> is INITIAL PROGRAM. Other verbs used with PROGRAM-ID are RECURSIVE and COMMON.

#### 11. When is COMMON attribute used?

A. COMMON attribute is used with nested COBOL programs. If it is not specified, other nested programs will not be able to access the program. PROGRAM-ID. Pgmname is COMMON PROGRAM.

#### 12. In which division and section, the Special-names paragraph appears?

A. Environment division and Configuration Section.

#### 13. What is the LOCAL-STORAGE SECTION?

A. Local-Storage is allocated each time the program is called and is de-allocated when the program returns via an EXIT PROGRAM, GOBACK, or STOP RUN. Any data items with a VALUE clauses are initialized to the appropriate value each time the program is called. The value in the data items is lost when the program returns. It is defined in the DATA DIVISION after WORKING-STORAGE SECTION

#### 14. What does passing BY REFERENCE mean?

A. When the data is passed between programs, the subprogram refers to and processes the data items in the calling program's storage, rather than working on a copy of the data. When CALL . . . BY REFERENCE *identifier*. In this case, the caller and the called share the same memory.

#### 15. What does passing BY CONTENT mean?

A. The calling program passes only the contents of the *literal*, or *identifier*. With a CALL . . . BY CONTENT, the called program cannot change the value of the *literal* or *identifier* in the calling program, even if it modifies the variable in which it received the *literal* or *identifier*.

#### 16. What does passing BY VALUE mean?

A. The calling program or method is passing the value of the *literal*, or *identifier*, not a reference to the sending data item. The called program or invoked method can change the parameter in the called program or invoked method. However, because the subprogram or method has access only to a temporary copy of the sending data item, those changes do not affect the argument in the calling program. Use By value, If you want to pass data to C program. Parameters must be of certain data type.

17. What is the default, passing BY REFERENCE or passing BY CONTENT or passing BY VALUE?

A. Passing by reference (the caller and the called share the same memory).

18. Where do you define your data in a program if the data is passed to the program from a Caller program?

A. Linkage Section

19. Define the structure of a COBOL subroutine.

A. The PROCEDURE DIVISION header must have a using a phrase, if the data needs to be passed to the program. The operands of the USING phrase must be defined in the LINKAGE SECTION as 01-level or 77-level entries. No VALUE clause is allowed unless the data defined is a condition name.

If the program needs to be returned to the CALLER, use EXIT PROGRAM statement at the end of the program. GOBACK is an alternative, but is nonstandard.

20. What is difference between next sentence and continue

A. NEXT SENTENCE gives control to the verb following the next period. CONTINUE gives control to the next verb after the explicit scope terminator. (This is not one of COBOL II's finer implementations). It's safest to use CONTINUE rather than NEXT SENTENCE in COBOL II. CONTINUE is like a null statement (do nothing) , while NEXT SENTENCE transfers control to the next sentence (!! ) (A sentence is terminated by a period). Check out by writing the following code example, one if sentence followed by 3 display statements: If 1 > 0 then next sentence end if display 'line 1' display 'line 2'. display 'line 3'.  
\*\*\* Note- there is a dot (.) only at the end of the last 2 statements, see the effect by replacing Next Sentence with Continue \*\*\*

21. What is the difference between Structured Cobol Programming and Object Oriented COBOL programming?

A. Structured programming is a Logical way of programming using which you divide the functionality's into modules and code logically. OOP is a Natural way of programming in which you identify the objects, first then write functions and procedures around the objects. Sorry, this may not be an adequate answer, but they are two different programming paradigms, which is difficult to put in a sentence or two.

22. PIC S9(4)COMP IS USED INSPITE OF COMP-3 WHICH OCCUPIES LESS SPACE.WHY?

A. S9(4) COMP uses only 2 bytes. 9(4) COMP-3 uses 3 bytes. 3 bytes are more than 2 bytes. Hence COMP is preferred over COMP-3 in this case.

23. How many number of bytes and digits are involved in S9(10) COMP?

8 bytes (double word) and 10 digits. Up to 9(9) comp use full word, up to 9(18) comp needs double word.

24. Which picture clause will you use to define a hexadecimal item in a VALUE clause?

A. 01 ws-hexitem PIC X(2) value X'020C'.  
01 ws-hex redefines PIC S9(3) comp-3.

25. How many numbers of decimal digits are possible, when the amount of storage allocated for a USAGE COMP item is a) half word b) full word c) double word?

A. 2 bytes (halfword) for 1-4 Digits 4 bytes (fullword) for 5-9  
8 bytes (doubleword) for 10-18

26. What is a scope terminator? Give examples.

Scope terminator is used to mark the end of a verb e.g. EVALUATE, END-EVALUATE; IF, END-IF.

27. How many dimensions are allowed for a table?

A. 7

28. How many subscripts or indexes are allowed for an OCCURS clause?

A. 7

29. Why cannot Occurs be used in 01 level?

A. Because, Occurs clause is there to repeat fields with the same format, but not the records.

30. Can a REDEFINES clause be used along with an OCCURS clause?

A. Yes, if the REDEFINES clause is subordinate to OCCURS clause.

31. Can you specify PIC clause and a VALUE with an OCCURS clause? Will the following code compile without errors?

01 WS-TABLE.

03 WS-TABLE-EL OCCURS 5 TIMES PIC X(1) VALUE SPACES.

A. Yes, the code will compile without any errors.

32. What would be the output, when the following code is executed?

01 WS-TABLE.

03 WS-TABLE-EL OCCURS 5 TIMES PIC X(1) VALUE 'AAAAA'.

A. It cannot be executed because the code will compile with error ' "VALUE" literal "AAAAA" exceeded the length specified in the "PICTURE" definition'.

33. 01 WS-TABLE.

03 WS-TABLE-EL OCCURS 5 TIMES PIC X(1) VALUE 'A'.

03 WS-EX REDEFINES WS-TABLE-EL PIC X(5). What can you expect?

A. Compile error. Direct Redefinition of OCCURS clause is not allowed.

34. 01 WS-TABLE.

03 WS-TABLE-EL OCCURS 5 TIMES.

04 FILLER-X PIC X(1) VALUE 'A'. 04 WS-EX REDEFINES FILLER-X PIC X(1).

What would be the output of DISPLAY WS-TABLE?

A. 'AAAAA'. The code will compile and execute as Redefinition of an item subordinate to OCCURS clause.

35. Is this allowed?

01 WS-TABLE.

03 FILLER-X PIC X(5) VALUE 'AAAAA'.

03 WS-EX REDEFINES FILLER-X.

04 WS-TABLE-EL OCCURS 5 TIMES PIC X(1).

A. Yes

35. Is this allowed?

01 WS-TABLE.

03 FILLER-X PIC X(5) VALUE 'AAAAA'.

03 WS-EX REDEFINES FILLER-X OCCURS 5 TIMES PIC X(1).

A. Yes

36. Which SEARCH verb is equivalent to PERFORM...VARYING?

The serial SEARCH verb (SEARCH without ALL)

37. What would be the output, when the following code is executed?

01 WS-TABLE.

03 WS-TABLE-EL OCCURS 5 TIMES PIC X(1) VALUE 'A'.

...

DISPLAY WS-TABLE.

A. The output will display 'AAAAA'

38. Can a SEARCH be applied to a table which does not have an INDEX defined?

A. No, the table must be indexed.

39. What are the different rules applicable to perform a serial SEARCH?

A. The SEARCH can be applied to only a table which has the OCCURS clause and INDEXED BY phrase,

Before the use of the SEARCH the index must have some initial value. To search from beginning, set the index value to 1. Use the SEARCH verb without ALL phrase

40. A table has two indexes defined. Which one will be used by the SEARCH verb?

A. The index named first will be used, by Search.

41. What are the different rules applicable to perform a binary SEARCH?

A. The table must be sorted in ascending or descending order before the beginning of the SEARCH. Use OCCURS clause with ASC/DESC KEY IS dataname1 option

The table must be indexed. There is no need to set the index value. Use SEARCH ALL verb

42. How does the binary search work?

A. First the table is split into two halves and in which half, the item need to be searched is determined. The half to which the desired item may belong is again divided into two halves and the previous procedure is followed. This continues until the item is found. SEARCH ALL is efficient for tables larger than 70 items.

43. What is the difference between a binary search and a sequential search? What are the pertinent COBOL commands?

A. In a binary search the table element key values must be in ascending or descending sequence. The table is 'halved' to search for equal to, greater than or less than conditions until the element is found. In a sequential search the table is searched from top to bottom, so (ironically) the elements do not have to be in a specific sequence. The binary search is much faster for larger tables, While sequential Search works well with smaller ones. SEARCH ALL is used for binary searches; SEARCH for sequential.

44. Explain the difference between an internal and an external sort. The pros & cons & internal sort syntax ...

A. An external sort is not coded as a COBOL program; it is performed through JCL and PGM=SORT. One can use IBM utility SYNCSORT for external sort process. It is understandable without any code reference. An internal sort can use two different syntaxes: 1.) USING, GIVING sorts are comparable to external sorts with no extra file processing; 2) INPUT PROCEDURE, OUTPUT PROCEDURE sorts allow for data manipulation before and/or after the sort. Syntax:

- SORT file-1 ON ASCENDING/DESCENDING KEY key...USING file-2 GIVING file-3.
- USING can be substituted by INPUT PROCEDURE IS para-1 THRU para-2
- GIVING can be substituted by OUTPUT PROCEDURE IS para-1 THRU para-2.
- file-1 is the sort workfile and must be described using SD entry in FILE SECTION.
- file-2 is the input file for the SORT and must be described using an FD entry in FILE SECTION and SELECT clause in FILE CONTROL.
- file-3 is the outfile from the SORT and must be described using an FD entry in FILE SECTION and SELECT clause in FILE CONTROL.
- file-1, file-2 & file-3 should not be opened explicitly.

- INPUT PROCEDURE is executed before the sort and records must be RELEASEd to the sort work file from the input procedure.
- OUTPUT PROCEDURE is executed after all records have been sorted. Records from the sort work file must be RETURNed one at a time to the output procedure.
- .How do you define a sort file in JCL that runs the COBOL program?
- Use the SORTWK01, SORTWK02,..... dd names in the step. Number of sort datasets depends on the volume of data being sorted, but a minimum of 3 is required.

45. Which is the default, TEST BEFORE or TEST AFTER for a PERFORM statement?

TEST BEFORE. By default the condition is checked before executing the instructions under Perform.

46. What is the difference between PERFORM ... WITH TEST AFTER and PERFORM ... WITH TEST BEFORE?

- A. If TEST BEFORE is specified, the condition is tested at the beginning of each repeated execution of the specified PERFORM range. If TEST AFTER is specified, the condition is tested at the end of the each repeated execution of the PERFORM range. With TEST AFTER, the range is executed at least once.

47. How do you code an in-line PERFORM?

- A. PERFORM ... <UNTIL> ... <statements> END-PERFORM.

48. In an EVALUTE statement is the order of the WHEN clauses significant?

Yes. Evaluation of the WHEN clauses proceeds from top to bottom and their sequence can determine results.

49. What is the default value(s) for an INITIALIZE and what keyword allows for an override of the default.

- A. INITIALIZE sets spaces to alphabetic and alphanumeric fields. Initialize sets Zeroes to numeric fields. FILLER, OCCURS DEPENDING ON items are left untouched. The REPLACING option can be used to override these defaults.

50. What is SET <condition-name> TO TRUE all about, anyway?

- A. In COBOL II the 88 levels can be set rather than moving their associated values to the related data item. (Web note: This change is not one of COBOL II's better specifications.)

51. What is LENGTH in COBOL II?

- A. LENGTH acts like a special register to tell the length of a group or an elementary item.

52. What is the function of a delimiter in STRING?

- A. A delimiter in STRING causes a sending field to be ended and another to be started.

53. What is the function of a delimiter in UNSTRING?

- A. A delimiter when encountered in the sending field causes the current receiving field to be switched to the next one indicated.

54. How will you count the number of characters in a null-terminated string?

- A. MOVE 0 TO char-count  
INSPECT null-terminated-string TALLYING char-count FOR CHARACTERS BEFORE  
X"00"

55. Which statement will you use to move non-null characters from a null-terminated String?

- A. UNSTRING null-terminated-string DELIMITED BY X"00" INTO target-area  
COUNT IN char-count. (There are other methods, such as 1) using PERFORM 2) using SEARCH 3) using INSPECT and MOVE etc...)

56. 77 COUNTR PIC 9 VALUE ZERO.

01 DATA-2 PIC X(11) . . .

INSPECT DATA-2

TALLYING COUNTR FOR LEADING "0"

REPLACING FIRST "A" BY "2" AFTER INITIAL "C"

If DATA-2 is 0000ALABAMA, what will DATA-2 and COUNTER be after the execution of INSPECT verb?

A. Counter=4. Data-2 will not change as the Initial 'C' is not found.

57. 01 DATA-4 PIC X(11) .

:::

INSPECT DATA-4 CONVERTING

"abcdefghijklmnopqrstuvwxyz" TO "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

AFTER INITIAL "/" BEFORE INITIAL "?"

What will the contents of DATA-4 be after the conversion statement is performed, if before conversion

a) DATA-4 = a/five/?six b) DATA-4 = r/Rexx/RRRR c) DATA-4 = zfour?inspe

A. a) a/FIVE/?six b) r/REXX/RRRR c) zfour?inspe (no change at all)

58. What kind of error is trapped by ON SIZE ERROR option?

A. Fixed-point overflow. Zero raised to the zero power.

Division by 0. Zero raised to a negative number.

A negative number raised to a fractional power.

59. What is the point of the REPLACING option of a copy statement?

A. REPLACING allows for the same copy to be used more than once in the same code by changing the replace value. COPY xxx REPLACING <psuedotext1> BY <psuedotext2>.

60. When is a scope terminator mandatory?

A. Scope terminators are mandatory for in-line PERFORMS and EVALUATE statements. For readability, it's recommended coding practice to always make scope terminators explicit.

61. Can you use REDEFINES clause in the FILE SECTION?

A. No

62. How will you define your record descriptions in the FILE SECTION if you want to use three different record descriptions for the same file?

A. FD filename

DATA RECORDS ARE rd01, rd02, rd03.

01 rd01 PIC X(n).

01 rd02 PIC X(n).

01 rd03 PIC X(n).

63. When will you open a file in the EXTEND mode?

A. When an existing file should be appended by adding new records at its end. EXTEND mode opens the file for output, but the file is positioned following the last record on the existing file.

64. What does a CLOSE WITH LOCK statement do?

A. The statement closes an opened file and it prevents the file from further being opened by the same program.

65. Which mode of opening is required when REWRITE is used?

A. I-O mode



66. Why is it necessary that the file be opened in I-O mode for REWRITE?

A. Before the REWRITE is performed, the record must be read from the file. Hence REWRITE includes an input operation and an output operation. Therefore, the file must be opened in I-O mode.

67. Which clause can be used instead of checking for FILE STATUS = 10?

A. FILE STATUS 10 is the end of file condition. Hence AT END clause can be used.

68. What is the format of a simple SORT verb? What kinds of files can be sorted using SORT?

A. SORT workfile ON ASC/DESC KEY key1, ASC/DESC KEY key2 ...

USING inputfile GIVING outputfile

Only sequential files can be sorted in this way.

69. What are the different rules of SORT that needs to be considered?

The input and output files must remain closed because SORT opens them and closes during the operation, The work file must have a SELECT clause. The work file must have sort description SD entry in the FILE SECTION. Input and Output files must have FD entries

70. What are INPUT PROCEDURE and OUTPUT PROCEDURE?

A. Sometimes, it is necessary that the records must be edited before or after the sorting. In such cases,

SORT workfile ASC/DESC KEY key1, ...

INPUT PROCEDURE IS ipproc

OUTPUT PROCEDURE is outproc

Is used. In the INPUT PROCEDURE the input file is opened, records are read and edited and then are released to the sorting operation. Finally the file is closed. RELEASE sortrecname FROM inp-rec.

In the OUTPUT PROCEDURE, output file is opened, the sorted record is returned to the Output record area and then the record is written. Finally the file is closed. RETURN workfile RECORD into out-rec.

71. What is the format of a simple MERGE verb? Can INPUT PROCEDURE and OUTPUT PROCEDURE can be specified for MERGE verb?

A. MERGE work file ON ASC/DESC KEY key1...

USING inputfile1, inputfile2...

GIVING outputfile

INPUT PROCEDURE cannot be specified. Only OUTPUT PROCEDURE can be specified

72. How will you position an indexed file at a specific point so that the subsequent sequential operations on the file can start from this point?

A. Use START

START filename KEY IS EQ/GT/LT.. dataname

INVALID KEY ...

73. What are the access mode requirements of START statement?

A. Access mode must be SEQUENTIAL or DYNAMIC

74. What are the opening mode requirements of START statement?

Files must be opened in the INPUT or I-O mode.

75. What is the LINKAGE SECTION used for?

A. The linkage section is used to pass data from one program to another program or to pass data from a PROC to a program. It is part of a called program that 'links' or maps to data items in the calling program's working storage. It is the part of the called program where these share items are defined.



76. If you were passing a table via linkage, which is preferable - a subscript or an index?

A. Wake up - you haven't been paying attention! It's not possible to pass an index via linkage. The index is not part of the calling programs working storage. Indexing uses binary displacement. Subscripts use the value of the occurrence.

77. What is the difference between a subscript and an index in a table definition?

A subscript is a working storage data definition item, typically a PIC (999) where a value must be moved to the subscript and then increment or decrement it by ADD TO and SUBTRACT FROM statements. An index is a register item that exists outside the program's working storage. You SET an index to a value and SET it UP BY value and DOWN BY value.

Subscript refers to the array occurrence while index is the displacement (in no of bytes) from the beginning of the array. An index can only be modified using PERFORM, SEARCH & SET. Need to have index for a table in order to use SEARCH, SEARCH ALL Cobol statements.

78. What is an in line PERFORM? When would you use it? Anything else to say about it?

The PERFORM and END-PERFORM statements bracket all COBOL II statements between them. The COBOL equivalent is to PERFORM or PERFORM THRU a paragraph. In line PERFORMs work as long as there are no internal GO TOs, not even to an exit. The in line PERFORM for readability should not exceed a page length - often it will reference other PERFORM paragraphs.

When the body of the Perform will not be used in other paragraphs. If the body of the Perform is a generic type of code (used from various other places in the program), it would be better to put the code in a separate para and use PERFORM paraname rather than in-line perform.

79. What is the use of EVALUATE statement? How do you come out of an EVALUATE statement?

A. Evaluate is like a case statement and can be used to replace nested ifs. The difference between EVALUATE and case is that no 'break' is required for EVALUATE i.e. control comes out of the EVALUATE as soon as one match is made, There is no need of any extra code. EVALUATE can be used in place of the nested IF THEN ELSE statements.

80. What are the different forms of EVALUATE statement?

A. EVALUATE EVALUATE SQLCODE ALSO FILE-STATUS  
WHEN A=B AND C=D WHEN 100 ALSO '00'

Imperative stmt imperative stmt

WHEN (D+X)/Y = 4 WHEN -305 ALSO '32'

imperative stmt imperative stmt

WHEN OTHER WHEN OTHER

imperative stmt imperative stmt

END-EVALUATE END-EVALUATE

EVALUATE SQLCODE ALSO A=B EVALUATE SQLCODE ALSO TRUE

WHEN 100 ALSO TRUE WHEN 100 ALSO A=B

imperative stmt imperative stmt

WHEN -305 ALSO FALSE WHEN -305 ALSO (A/C=4)

imperative stmt imperative stmt

END-EVALUATE END-EVALUATE

81. Can you use the INSPECT (with TALLYING option) Cobol verb in a CICS COBOL program?

A. Yes, under COBOL II environment, but not OS/VS COBOL.

82. What is an explicit scope terminator?

A scope terminator brackets its preceding verb, eg. IF .. END-IF, so that all statements between the verb and its scope terminator are grouped together. Other common COBOL II verbs are READ, PERFORM, EVALUATE, SEARCH and STRING.

83. What is the significance of 'above the line' and 'below the line'?

A. Before IBM introduced MVS/XA architecture in the 1980's a program's virtual storage was limited to 16 megs. Programs compiled with a 24-bit mode can only address 16 MB of

space, as though they were kept under an imaginary storage line. With COBOL II a program compiled with a 31 bit mode can be 'above the 16 Mb line. (This 'below the line', 'above the line' imagery confuses most mainframe programmers, who tend to be a literal minded group.)

**84. What was removed from COBOL in the COBOL II implementation?**

A. Partial list: REMARKS, NOMINAL KEY, PAGE-COUNTER, CURRENT-DAY, TIME-OF-DAY, STATE, FLOW, COUNT, EXAMINE, EXHIBIT, READY TRACE and RESET TRACE.

**85. Explain call by context by comparing it to other calls.**

A. The parameters passed in a call by context are protected from modification by the called program. In a normal call they are able to be modified.

**86. What is the difference between comp and comp-3 usage? Explain other COBOL usages.**

Comp is a binary usage, while comp-3 indicates packed decimal. The other common usages are binary and display. Display is the default. Comp is defined as the fastest/preferred numeric data type for the machine it runs on. IBM Mainframes are typically binary and AS400's are packed.'

**87. I understand the possible causes for S0C1 & S0C4 abends, but what are they really?**

A. A S0C1 occurs if the CPU attempts to execute binary code that isn't a valid machine instruction; e.g. if you attempt to execute data. A S0C4 is a memory protection violation. This occurs if a program attempts to access storage beyond the areas assigned to it.

**88. What happens when we move a comp-3 field to an edited ( say z(9).zz-)**

A. The editing characters are to be used with data items with usage clause as display, which is the default. When you try displaying a data item with usage as computational it does not give the desired display format because the data item is stored as packed decimal. So if u want this particular data item to be edited u have to move it into a data item whose usage is display and then have that particular data item edited in the format desired.

**89. What are the causes for S0C1, S0C4, S0C5, S0C7, S0CB abends**

S0C1 - May be due to 1.Missing or misspelled DD name 2.Read/Write to unopened dataset 3.Read to dataset opened output 4.Write to dataset opened input 5.Called subprogram not found.

S0C4 may be due to 1.Missing Select statement(during compile) 2.Bad Subscript/index 3.Protection Exception 4.Missing parameters on called subprogram 5.Read/Write to unopened file 6.Move data from/to unopened file.

S0C5 May be due to 1.Bad Subscript/index 2.Close an unopen dataset 3.Bad exit from a perform 4.Access to I/O area(FD) before read.

S0C7 may be due to 1.Numeric operation on non-numeric data 2.Un-initialize working-storage 3.Coding past the maximum allowed sub script. S0CB may be due to 1.Division by Zero

**90. What will happen if you code GO BACK instead of STOP RUN in a stand-alone COBOL program i.e. a program which is not calling any other program.**

Both give the same results when a program is not calling any other program.

**91. What is the difference between an External and a Global Variable 's?**

A. Global variables are accessible only to the batch program whereas external variables can be referenced from any batch program residing in the same system library.

**92. WHAT IS REPORT-ITEM?**

A. A REPORT-item is a field to be printed that contains EDIT SYMBOLS

**93. You are writing report program with 4 levels of totals:city,state,region and country. The codes being used can be the same over the different levels, meaning a city code of 01 can be in any number of states, and the same applies to state and region code show. Do you do your checking for breaks and how do you do add to each level?**

Always compare on the highest-level first, because if you have a break at a highest level, each level beneath it must also break. Add to the lowest level for each rec but add to the higher level only on break.

**94. What is PSB & ACB?**

- A. PSB : Program specification block. Inform about how a specific program is to be access one or more IMS DB. It consists of PCB(Prg Communication Block). Information to which segment in DB can be accessed, what the program is allowed to do with those segment and how the DB is to be accessed. ACB : Access Control Blocks are generated by IMS as an expansion of information contained in the PSB in order to speed up the access to the applicable DBD's.

**95. What's a LDS(Linear Data Set) and what's it used for ?**

- A. LDS is a VSAM dataset in name only. It has unstructured 4k (4096 bytes) fixed size CIs which do not contain control fields and therefore from VSAM's standpoint they do not contain any logical records. There is no freespace, and no access from Cobol. Can be accessed by DB2 and IMS fast path datasets. LDS is essentially a table of data maintained on disk. The 'table entries' must be created via a user program and can only be logically accessed via a user program. When passed, the entire LDS must be mapped into storage, then data is accessed via base and displacement type processing.

**96. What is the Importance of GLOBAL clause According to new standards of COBOL**

- A. When any data name, file-name , Record-name, condition name or Index defined in an Including Program can be referenced by a directly or indirectly in an included program, Provided the said name has been declared to be a global name by GLOBAL Format of Global Clause is 01 data-1 PIC 9(5) IS GLOBAL.

**97. What is the Purpose of POINTER Phrase in STRING command**

- A. The Purpose of POINTER phrase is to specify the leftmost position within receiving field where the first transferred character will be stored

**98.How do we get currentdate from system with century?**

- A. By using Intrinsic function, FUNCTION CURRENT-DATE

**99.what is the difference between search and search all in the table handling?**

- A. Search is a linear search and search all is a binary search.

**100.What is the maximum length of a field you can define using COMP-3?**

- A. 10 Bytes (S9(18) COMP-3).

**101.How many Sections are there in Data Division?.**

- A. SIX SECTIONS 1.'FILE SECTION' 2.'WORKING-STORAGE SECTION' 3.'LOCAL-STORAGE SECTION' 4.'SCREEN SECTION' 5.'REPORT SECTION' 6.'LINKAGE SECTION' In COBOL II, there are only 4 sections. 1.'FILE SECTION' 2.'WORKING-STORAGE SECTION' 3.'LOCAL-STORAGE SECTION' 4.'LINKAGE SECTION'.

**102.How can I tell if a module is being called DYNAMICALLY or STATICALLY?**

- A. The ONLY way is to look at the output of the linkage editor (IEWL)or the load module itself. If the module is being called DYNAMICALLY then it will not exist in the main module, if it is being called STATICALLY then it will be seen in the load module. Calling a working storage variable, containing a program name, does not make a DYNAMIC call. This type of calling is known as IMPLICITE calling as the name of the module is implied by the contents of the working storage variable. Calling a program name literal (CALL).

**103.What is the difference between a DYNAMIC and STATIC call in COBOL.**

- A. To correct an earlier answer:All called modules cannot run standalone if they require program variables passed to them via the LINKAGE section. DYNAMICally called modules are those that are not bound with the calling program at link edit time (IEWL for IBM) and so are loaded from the program library (joblib or steplib) associated with the job.

For DYNAMIC calling of a module the DYNAM compiler option must be chosen, else the linkage editor will not generate an executable as it will expect null address resolution of all called modules. A STATICally called module is one that is bound with the calling module at link edit, and therefore becomes part of the executable load module.

104.What is the difference between PIC 9.99 and 9v99?

- A. PIC 9.99 is a FOUR-POSITION field that actually contains a decimal point where as PIC 9v99 is THREE-POSITION numeric field with implied or assumed decimal position.

105.How is PIC 9.99 is different from PIC 9v99?

- A. PIC 9.99 is a four position field that actually contains a decimal point where as 9v99 is a three position numeric field with an implied or assumed decimal point.

106.what is Pic 9v99 Indicates?

- A. PICTURE 9v99 is a three position Numeric field with an implied or assumed decimal point after the first position; the v means an implied decimal point.

107.what guidelines should be followed to write a structured COBOL program?

- 1) Use 'EVALUATE' stmt for constructing cases. 2) Use scope terminators for nesting. 3)Use in-line Perform stmt for writing 'do ' constructions. 4) Use Test Before and test after in the Perform stmt for writing Do-While constructions.

108.Read the following code.

01 ws-n PIC 9(2) value zero.

a-para.

move 5 to ws-n.

perform b-para ws-n times.

b-para.

move 10 to ws-n.

How many times will b-para be executed ?

- A. 5 Times only. it will not take the value 10 that is initialized in the loop.

109.What are some examples of command terminators?

- A. END-IF, END-EVALUATE

110.What care has to be taken to force program to execute above 16 Meg line?

- A. Make sure that link option is AMODE=31 and RMODE=ANY. Compile option should never have SIZE(MAX).BUFSIZE can be 2K, efficient enough.

111.Give some advantages of REDEFINES clause.

- A. You can REDEFINE a Variable from one PICTURE class to another PICTURE class by using the same memory location. By REDEFINES we can INITIALISE the variable in WORKING-STORAGE Section itself.3. We can REDEFINE a Single Variable into so many sub-variables.(This facility is very useful in solving Y2000 Problem.)

112.Why do we code s9(4)comp. Inspite of knowing comp-3 will occupy less space.

- A. Here s9(4)comp is small integer ,so two words equal to 8 bytes. Totally it will occupy 2 bytes(4 words).here in s9(4) comp-3 as one word is equal to 1/2 byte.4 words equal to 2 bytes and sign will occupy 1/2 bytes totally it will occupy 3 bytes.

113.The maximum number of dimensions that an array can have in COBOL-85 is \_\_\_\_\_.

Answer: SEVEN in COBOL - 85 and THREE in COBOL - 84

114.Name the divisions in a COBOL program.

- A. IDENTIFICATION DIVISION, ENVIRONMENT DIVISION, DATA DIVISION, PROCEDURE DIVISION.

115.What are the different data types available in COBOL?

A. Alpha-numeric (X), alphabetic (A) and numeric (9).

116.What is 77 level used for ?

A. Elementary level item. Cannot be subdivisions of other items (cannot be qualified), nor can they be subdivided themselves.

117.What is 88 level used for ?

A. For defining condition names.

118.What is level 66 used for ?

A. For RENAME clause.

119.What does the IS NUMERIC clause establish?

A. IS NUMERIC can be used on alphanumeric items, signed numeric & packed decimal items and unsigned numeric & packed decimal items. IS NUMERIC returns TRUE if the item only consists of 0-9. However, if the item being tested is a signed item, then it may contain 0-9, + and - .

120.My program has an array defined to have 10 items. Due to a bug, I find that even if the program access the 11th item in this array, the program does not abend. What is wrong with it?

A. Must use compiler option SSRANGE if you want array bounds checking. Default is NOSSRANGE.

121. What is the difference between performing a SECTION and a PARAGRAPH?

A. Performing a SECTION will cause all the paragraphs that are part of the section, to be performed.

Performing a PARAGRAPH will cause only that paragraph to be performed.

122.Can I redefine an X(200) field with a field of X(100) ?

Yes.

123.What does EXIT do?

A. Does nothing ! If used, must be the only sentence within a paragraph.

124.Can I redefine an X(100) field with a field of X(200)?

A. Yes. Redefines just causes both fields to start at the same location. For example:

01 WS-TOP PIC X(1)

01 WS-TOP-RED REDEFINES WS-TOP PIC X(2).

If you MOVE '12' to WS-TOP-RED,

DISPLAY WS-TOP will show 1 while

DISPLAY WS-TOP-RED will show 12.

125.Can I redefine an X(200) field with a field of X(100) ?

A. Yes.

126.What do you do to resolve SOC-7 error?

A. Basically you need to correcting the offending data. Many times the reason for SOC7 is an un-initialized numeric item. Examine that possibility first. Many installations provide you a dump for run time abends ( it can be generated also by calling some subroutines or OS services thru assembly language). These dumps provide the offset of the last instruction at which the abend occurred. Examine the compilation output XREF listing to get the verb and the line number of the source code at this offset. Then you can look at the source code to find the bug. To get capture the runtime dumps, you will have to define some datasets (SYSABOUT etc ) in the JCL. If none of these are helpful, use judgement and DISPLAY to localize the source of error. You may even use batch program debugging tools.

**127.How is sign stored in Packed Decimal fields and Zoned Decimal fields?**

A. Packed Decimal fields: Sign is stored as a hex value in the last nibble (4 bits ) of the storage. Zoned Decimal fields: As a default, sign is over punched with the numeric value stored in the last bite.

**128.How is sign stored in a comp-3 field?**

A. It is stored in the last nibble. For example if your number is +100, it stores hex 0C in the last byte, hex 1C if your number is 101, hex 1D if the number is -101, hex 2D if the number is -102 etc...

**129.How is sign stored in a COMP field ?**

A. In the most significant bit. Bit is on if -ve, off if +ve.

**130.What is the difference between COMP & COMP-3 ?**

A. COMP is a binary storage format while COMP-3 is packed decimal format.

**131.What is COMP-1? COMP-2?**

COMP-1 - Single precision floating point. Uses 4 bytes.

COMP-2 - Double precision floating point. Uses 8 bytes.

**132.How do you define a variable of COMP-1? COMP-2?**

A. No picture clause to be given. Example 01 WS-VAR USAGE COMP-1

**133.How many bytes does a S9(7) COMP-3 field occupy ?**

A. Will take 4 bytes. Sign is stored as hex value in the last nibble.

General formula is  $\text{INT}((n/2) + 1)$ , where  $n=7$  in this example.

**134.How many bytes does a S9(7) SIGN TRAILING SEPARATE field occupy ?**

A. Will occupy 8 bytes (one extra byte for sign).

**135.What is the maximum size of a 01 level item in COBOL I? in COBOL II?**

In COBOL II: 16777215

**136.What is COMP SYNC?**

A. Causes the item to be aligned on natural boundaries. Can be SYNCHRONIZED LEFT or RIGHT.

For binary data items, the address resolution is faster if they are located at word boundaries in the memory. For example, on main frame the memory word size is 4 bytes. This means that each word will start from an address divisible by 4. If my first variable is x(3) and next one is s9(4) comp, then if you do not specify the SYNC clause, S9(4) COMP will start from byte 3 ( assuming that it starts from 0 ). If you specify SYNC, then the binary data item will start from address 4. You might see some wastage of memory, but the access to this comp field is faster.

**137.How do you reference the following file formats from COBOL programs?**

Fixed Block File - Use ORGANISATION IS SEQUENTIAL. Use RECORDING MODE IS F, BLOCK CONTAINS 0.

Fixed Unblocked - Use ORGANISATION IS SEQUENTIAL. Use RECORDING MODE IS F, do not use BLOCK CONTAINS.

Variable Block File - Use ORGANISATION IS SEQUENTIAL. Use RECORDING MODE IS V, BLOCK CONTAINS 0. Do not code the 4 bytes for record length in FD. i.e. JCL record length will be max record length in program + 4

Variable Unblocked - Use ORGANISATION IS SEQUENTIAL. Use RECORDING MODE IS V, do not use BLOCK CONTAINS. Do not code 4 bytes for record length in FD ie JCL rec length will be max rec length in pgm + 4.

ESDS VSAM file - Use ORGANISATION IS SEQUENTIAL.

KSDS VSAM file - Use ORGANISATION IS INDEXED, RECORD KEY IS, Alternate Record Key Is

RRDS File - Use ORGANISATION IS RELATIVE, RELATIVE KEY IS

Printer File - Use ORGANISATION IS SEQUENTIAL. Use RECORDING MODE IS F, BLOCK CONTAINS 0. (Use RECFM=FBA in JCL DCB).



138.What are different file OPEN modes available in COBOL? In which modes are the files Opened to write.

A. Different Open modes for files are INPUT, OUTPUT, I-O and EXTEND. Of which Output and Extend modes are used to write new records into a file.

139.In the JCL, how do you define the files referred to in a subroutine?

A. Supply the DD cards just as you would for files referred to in the main program.

140.Can you REWRITE a record in an ESDS file? Can you DELETE a record from it?

A. Can rewrite(record length must be same), but not delete.

141.What is file status 92?

A. Logic error. e.g., a file is opened for input and an attempt is made to write to it.

142.What is file status 39?

A. Mismatch in LRECL or BLOCKSIZE or RECFM between your COBOL pgm & the JCL (or the dataset label). You will get file status 39 on an OPEN.

143.What is Static, Dynamic linking ?

A. In static linking, the called subroutine is link-edited into the calling program , while in dynamic linking, the subroutine & the main program will exist as separate load modules. You choose static/dynamic linking by choosing either the DYNAM or NODYNAM link edit option. (Even if you choose NODYNAM, a CALL identifier (as opposed to a CALL literal), will translate to a DYNAMIC call).

A statically called subroutine will not be in its initial state the next time it is called unless you explicitly use INITIAL or you do a CANCEL. A dynamically called routine will always be in its initial state.

144.Explain NEXT and CONTINUE verbs for file handling.

A The Continue verb is used for a situation where there is no EOF condition. i.e. The records are to be accessed again and again in a file. Whereas in the next verb the indexed file is accessed sequentially, whence when index clause is accessed sequentially read next record command is used.

145.What is AMODE(24), AMODE(31), RMODE(24) and RMODE(ANY)? (applicable to only MVS/ESA).

A. These are compile/link edit options. AMODE - Addressing mode. RMODE - Residency mode.

AMODE(24) - 24 bit addressing. AMODE(31) - 31 bit addressing. AMODE(ANY) - Either 24 bit or 31 bit addressing depending on RMODE. RMODE(24) - Resides in virtual storage below 16 Meg line. Use this for 31 bit programs that call 24 bit programs. (OS/VS Cobol pgms use 24 bit addresses only). RMODE(ANY) - Can reside above or below 16 Meg line.

146.What compiler option would you use for dynamic linking?

DYNAM.

147.What is SSRANGE, NOSSRANGE ?

A. These are compiler options w.r.t subscript out of range checking. NOSSRANGE is the default and if chosen, no run time error will be flagged if your index or subscript goes out of the permissible range.

148.How do you set a return code to the JCL from a COBOL program?

A. Move a value to RETURN-CODE register. RETURN-CODE should not be declared in your program.

149.How can you submit a job from COBOL programs?

A. Write JCL cards to a dataset with



//xxxxxxx SYSOUT=(A,INTRDR) where 'A' is output class, and dataset should be opened for output in the program. Define a 80 byte record layout for the file.

#### 150.What are the differences between OS VS COBOL and VS COBOL II?

- OS/VS Cobol pgms can only run in 24 bit addressing mode, VS Cobol II pgms can run either in 24 bit or 31 bit addressing modes allowing program to address above 16 Meg main storage line.
- Report writer is supported only in OS/VS Cobol.
- USAGE IS POINTER is supported only in VS COBOL II.
- Reference modification eg: WS-VAR(1:2) is supported only in VS COBOL II.
- COBOL II introduces new features (EVALUATE, SET ... TO TRUE, CALL ... BY CONTEXT, etc)
- Scope terminators are supported in COBOL II.
- OS/VS Cobol follows ANSI 74 stds while VS COBOL II follows ANSI 85 stds.
- Under CICS Calls between VS COBOL II programs are supported.
- COBOL II supports structured programming by using in-line PERFORM 's.
- COBOL II does not support old features (READY TRACE, REPORT-WRITER, ISAM, etc.).
- In non-CICS environment, it is possible. In CICS, this is not possible.

#### What is the difference between Structured Cobol Programming and Object Oriented COBOL programming ?

**Answer:** Structured programming is a Logical way of programming, you divide the functionalities into modules and code logically. OOP is a Natural way of programming, you identify the objects first, then write fuctions, procedures around the objects. Sorry, this may not be an adequate answer, but they are two different programming paradigms which is difficult to put in a sentence or two.

#### what is difference between next sentence and continue

**Answer:** They appear to be similar, that is, the control goes to the next sentence in the paragraph. But, Next Sentence would take the control to the sentence after it finds a fullstop (.). Check out by writing the following code example, one if sentence followed by 3 display statements (sorry they appear one line here because of formatting restrictions) If 1 > 0 then next sentence end if display 'line 1' display 'line 2'. display 'line 3'. \*\*\* Note- there is a dot (.) only at the end of the last 2 statements, see the effect by replacing Next Sentence with Continue \*\*\* Hope this helps!

#### what is difference between next sentence and continue

**Answer:** i don't know

#### I understand the possible causes for S0C1 & S0C4 abends, but what are they really?

**Answer:** A S0C1 occurs if the CPU attempts to execute binary code that isn't a valid machine instruction; e.g. if you attempt to execute data. A S0C4 is a memory protection violation. This occurs if a program attempts to access storage beyond the areas assigned to it.

#### which type of sort do you know?

**Answer:** internal sort and external sort.

#### PIC S9(4)COMP IS USED INSPITE OF COMP-3 WHICH OCCUPIES LESS SPACE.WHY?

**Answer:** The Q&A submitted by sivakumar on March 24,1998 is absolutely wrong. Dear Sivakumar, for your kind information: 9(4) COMP uses only 2 bytes and 2 bytes is not = 4 words. 9(4) COMP-3 uses 3 bytes. 3 bytes is more than 2 bytes. Hence COMP is preferred over COMP-3 in this case.

What divisions, sections and paragraphs are mandatory for a COBOL program?

**Answer:** IDENTIFICATION DIVISION and PROGRAM-ID paragraph are mandatory for a compilation error free COBOL program.

Can JUSTIFIED be used for all the data types?

**Answer:** No, it can be used only with alphabetic and alphanumeric data types.

Ans we to bala s bandlas question what happens when we move a comp-3 field to an edited ( say z(9).zz-)

**Answer:** the editing characters r to be used with data items with usage clause as display which is the default. when u try displaying a data item with usage as computational it does not give the desired display format becoz the data item is stored as packed decimal. So if u want this particular data item to be edited u have to move it into a data item whose usage is display and then have that particular data item edited in the format desired.

What are the causes for S0C1, S0C4, S0C5, S0C7, S0CB abends

**Answer:** S0C1 - May be due to

1. Missing or misspelled DD name
2. Read/Write to unopened dataset
3. Read to dataset opened output
4. Write to dataset opened input
5. Called subprogram not found

S0C4 may be due to

1. Missing Select statement (during compile)
2. Bad Subscript/index
3. Protection Exception
4. Missing parameters on called subprogram
5. Read/Write to unopened file
6. Move data from/to unopened file

S0C5 May be due to

1. Bad Subscript/index
2. Close an unopen dataset
3. Bad exit from a perform
4. Access to I/O area (FD) before read

S0C7 may be due to

1. Numeric operation on non-numeric data
2. Un-initialize working-storage
3. Coding past the maximum allowed sub script

S0CB may be due to

1. Division by Zero

What will happen if you code GO BACK instead of STOP RUN in a stand alone COBOL program i.e. a program which is not calling any other program.

**Answer:** Both give the same results when a program is not calling any other program. The answer given by Mr.Krishnan that when go back is coded the program goes into infinite loop is not correct. Goback will give the control to the system even though it is a single program.

#### WHAT IS THE DIFFERENCE BETWEEN EXTERNAL AND GLOBAL VARIABLES?

**Answer:** Global variables are accessible only to the batch program whereas external variables can be referenced from any batch program residing in the same system library.

#### What is difference between COBOL and VS COBOL II?

**Answer:** In using COBOL on PC we have only flat files and the programs can access only limited storage, whereas in VS COBOL II on M/F the programs can access upto 16MB or 2GB depending on the addressing and can use VSAM files to make I/O operations faster.

#### Why occurs can not be used in 01 level ?

**Answer:** because, Occurs clause is there to repeat fields with same format, not the records.

#### What is PSB & ACB?

**Answer:** PSB : Program specification block. Infor about how a specific program is to be access one or more IMS DB. It consist of PCB(Prg Communication Block). Information to which which segment in DB can be accessed, what the program is allowed to do with those segment and how the DB is to be accessed. ACB : Access Control Blocks are generated by IMS as an expansion of information contained in the PSB in order to speed up the access to the applicable DBD's.

#### What's a LDS(Linear Data Set) and what's it used for ?

**Answer:** LDS is a VSAM dataset in name only. It has unstructured 4k (4096 bytes) fixed size CIs which do not contain control fields and therefore from VSAM's standpoint they do not contain any logical records. There is no freespace, and no access from Cobol. Can be accessed by DB2 and IMS fast path datasets. LDS is essentially a table of data maintained on disk. The 'table entries' must be created via a user program and can only be logically accessed via a user program. When passed, the entire LDS must be mapped into storage, then data is accessed via base and displacement type processing.

#### What is the Importance of GLOBAL clause According to new standards of COBOL

**Answer:** When any data name, file-name , Record-name, condition name or Index defined in an Including Program can be referenced by a directly or indirectly in an included program, Provided the said name has been declared to be a global name by GLOBALFormat of Global Clause is 01 data-1 pic 9(5) IS GLOBAL.

#### What is the Purpose of POINTER Phrase in STRING command

**Answer:** The Purpose of POINTER phrase is to specify the leftmost position whithin Receiving field where the first transfered character will be stored

#### How do we get currentdate from system with century?

**Answer:** By using Intrinsic function, FUNCTION CURRENT-DATE

#### What is the maximum length of a field you can define using COMP-3?

**Answer:** 10 Bytes (S9(18) COMP-3).

#### HOW MANY SECTIONS ARE THERE IN DATA DIVISION?

**Answer:** SIX SECTIONS1.FILE SECTION2.WORKING-STORAGE SECTION3.LOCAL-STORAGE SECTION4.SCREEN SECTION5.REPORT SECTION6.LINKAGE SECTION

How can I tell if a module is being called DYNAMICALLY or STATICALLY?

**Answer:** The ONLY way is to look at the output of the linkage editor (IEWL) or the load module itself. If the module is being called DYNAMICALLY then it will not exist in the main module, if it is being called STATICALLY then it will be seen in the load module. Calling a working storage variable, containing a program name, does not make a DYNAMIC call. This type of calling is known as IMPLICIT calling as the name of the module is implied by the contents of the working storage variable. Calling a program name literal (CALL

What is the difference between a DYNAMIC and STATIC call in COBOL.

**Answer:** All called modules cannot run standalone if they require program variables passed to them via the LINKAGE section. DYNAMICALLY called modules are those that are not bound with the calling program at link edit time (IEWL for IBM) and so are loaded from the program library (joblib or steplib) associated with the job. For DYNAMIC calling of a module the DYNAM compiler option must be chosen, else the linkage editor will not generate an executable as it will expect full address resolution of all called modules. A STATICALLY called module is one that is bound with the calling module at link edit, and therefore becomes part of the executable load module.

What is the difference between PIC 9.99 and 9v99?

**Answer:** PIC 9.99 is a FOUR-POSITION field that actually contains a decimal point whereas PIC 9v99 is THREE-POSITION numeric field with implied or assumed decimal position.

How is PIC 9.99 different from PIC 9v99?

**Answer:** PIC 9.99 is a four position field that actually contains a decimal point whereas 9v99 is a three position numeric field with an implied or assumed decimal point.

What are some examples of command terminators?

**Answer:** END-IF, END-EVALUATE

What care has to be taken to force program to execute above 16 Meg line?

**Answer:** Make sure that link option is AMODE=31 and RMODE=ANY. Compile option should never have SIZE(MAX). BUFSIZE can be 2K, efficient enough.

Give some advantages of REDEFINES clause.

**Answer:** 1. You can REDEFINE a Variable from one PICTURE class to another PICTURE class by using the same memory location. 2. By REDEFINES we can INITIALISE the variable in WORKING-STORAGE Section itself. 3. We can REDEFINE a Single Variable into so many subvariables. (This facility is very useful in solving Y2000 Problem.)

Why do we code s9(4)comp. In spite of knowing comp-3 will occupy less space.

**Answer:** Here s9(4)comp is small integer, so two words equal to 1 byte so totally it will occupy 2 bytes (4 words). Here in s9(4) comp-3 as one word is equal to 1/2 byte. 4 words equal to 2 bytes and sign will occupy 1/2 byte so totally it will occupy 3 bytes.

The maximum number of dimensions that an array can have in COBOL-85 is \_\_\_\_\_.

**Answer:** SEVEN in COBOL - 85 and THREE in COBOL - 84

What is the LINKAGE SECTION used for?

**Answer:** The linkage section is used to pass data from one program to another program or to pass data from a PROC to a program.

### What are the differences between COBOL and COBOL II?

There are at least five differences: COBOL II supports structured programming by using **in line PERFORMs** and explicit **scope terminators**, it introduces new features (EVALUATE, SET .. TO TRUE, CALL .. BY CONTEXT, etc), it permits programs to be loaded and addressed **above the 16 megabyte line**, it does not support many old features (READY TRACE, REPORT-WRITER, ISAM, etc.), and it offers enhanced CICS support.

### What is an explicit scope terminator?

A **scope terminator** brackets its preceding verb, eg. IF .. END-IF, so that all statements between the verb and its scope terminator are grouped together. Other common COBOL II verbs are READ, PERFORM, EVALUATE, SEARCH and STRING.

### What is an in line PERFORM? When would you use it? Anything else to say about it?

The PERFORM and END-PERFORM statements bracket all COBOL II statements between them. The COBOL equivalent is to PERFORM or PERFORM THRU a paragraph. In line PERFORMs work as long as there are no internal GO TOs, not even to an exit. The in line PERFORM for readability should not exceed a page length - often it will reference other PERFORM paragraphs.

### What is the difference between NEXT SENTENCE and CONTINUE?

NEXT SENTENCE gives control to the verb following the next period. CONTINUE gives control to the next verb after the explicit scope terminator. (This is not one of COBOL II's finer implementations). It's safest to use CONTINUE rather than NEXT SENTENCE in COBOL II.

### What COBOL construct is the COBOL II EVALUATE meant to replace?

EVALUATE can be used in place of the nested IF THEN ELSE statements.

### What is the significance of 'above the line' and 'below the line'?

Before IBM introduced MVS/XA architecture in the 1980's a program's virtual storage was limited to 16 megs. Programs compiled with a 24 bit mode can only address 16 Mb of space, as though they were kept under an imaginary storage line. With COBOL II a program compiled with a 31 bit mode can be 'above the 16 Mb line'. (This '**below the line**', '**above the line**' imagery confuses most mainframe programmers, who tend to be a literal minded group.)

### What was removed from COBOL in the COBOL II implementation?

Partial list: REMARKS, NOMINAL KEY, PAGE-COUNTER, CURRENT-DAY, TIME-OF-DAY, STATE, FLOW, COUNT, EXAMINE, EXHIBIT, READY TRACE and RESET TRACE.

### Explain call by context by comparing it to other calls.

The parameters passed in a **call by context** are protected from modification by the called program. In a normal call they are able to be modified.

### What is the linkage section?

The **linkage section** is part of a called program that 'links' or maps to data items in the calling program's working storage. It is the part of the called program where these share items are defined.

What is the difference between a **subscript** and an **index** in a table definition?

A subscript is a working storage data definition item, typically a PIC (999) where a value must be moved to the subscript and then incremented or decremented by ADD TO and SUBTRACT FROM statements. An index is a register item that exists outside the program's working storage. You SET an index to a value and SET it UP BY value and DOWN BY value.

If you were passing a table via linkage, which is preferable - a **subscript** or an **index**?

**Wake up** - you haven't been paying attention! It's not possible to pass an index via linkage. The index is not part of the calling programs working storage. Those of us who've made this mistake, appreciate the lesson more than others.

Explain the difference between an **internal** and an **external sort**, the pros and cons, internal sort syntax etc.

An external sort is not COBOL; it is performed through JCL and PGM=SORT. It is understandable without any code reference. An internal sort can use two different syntaxes: 1.) USING, GIVING sorts are comparable to external sorts with no extra file processing; 2) INPUT PROCEDURE, OUTPUT PROCEDURE sorts allow for data manipulation before and/or after the sort.

What is the difference between comp and comp-3 usage? Explain other COBOL usages.

Comp is a binary usage, while comp-3 indicates packed decimal. The other common usages are binary and display. Display is the default. **3/28/00 Dave Herrmann:** 'I was reading your FAQ on Cobol, as an fyi Comp is defined as the fastest/preferred numeric data type for the machine it runs on. IBM Mainframes are typically binary and AS400's are packed.'

When is a **scope terminator** mandatory?

**Scope terminators** are mandatory for in-line PERFORMS and EVALUATE statements. For readability, it's recommended coding practice to always make **scope terminators** explicit.

In a COBOL II PERFORM statement, when is the conditional tested, before or after the perform execution?

In COBOL II the optional clause WITH TEST BEFORE or WITH TEST AFTER can be added to all perform statements. By default the test is performed before the perform.

In an EVALUTE statement is the order of the WHEN clauses significant?

Absolutely. Evaluation of the WHEN clauses proceeds from top to bottom and their sequence can determine results.

What is the default value(s) for an INITIALIZE and what keyword allows for an override of the default.

INITIALIZE moves spaces to alphabetic fields and zeros to alphanumeric fields. The REPLACING option can be used to override these defaults.

What is SET TO TRUE all about, anyway?

In COBOL II the 88 levels can be set rather than moving their associated values to the related data item. (**Web note:** This change is not one of COBOL II's better specifications.)

What is LENGTH in COBOL II?

LENGTH acts like a **special register** to tell the length of a group or elementary item.

What is the difference between a **binary search** and a **sequential search**? What are the pertinent COBOL commands?

In a **binary search** the table element key values must be in ascending or descending sequence. The table is 'halved' to search for equal to, greater than or less than conditions until the element is found. In a **sequential search** the table is searched from top to bottom, so (ironically) the elements do not have to be in a specific sequence. The binary search is much faster for larger tables, while sequential works well with smaller ones. **SEARCH ALL** is used for binary searches; **SEARCH** for sequential.

What is the point of the REPLACING option of a copy statement?

REPLACING allows for the same copy to be used more than once in the same code by changing the replace value.

What is AMODE and RMODE? What does 24 or 31 mean to it?

Addressing Mode/Access Mode

AMODE(24) indicates 24-bit (three-byte) addressing - memory below the line.

AMODE(31) indicates 31-bit addressing - memory above and below the line.

AMODE=ANY indicates the program may use either of the addressing technique.

Run Mode/Residency Mode

RMODE(24) indicates that the program must be loaded into memory below the line

RMODE(31) indicates that the program can be loaded either below or above the line.

RMODE=ANY indicates that the program can be run in either 24 bit (below) or 31 bit memory (above).

What is the difference between a SEARCH and a SEARCH ALL statement?

The first is a sequential search, the second is a binary search

Explain difference between TEST and OPTIMIZE Compile directives

Use TEST to produce object code that can be executed with VSCOBOL II batch or interactive debug. When you specify TEST, the following options go into effect: RES, NOFDUMP, NOOPTIMIZE, and OBJECT. Use OPTIMIZE to reduce the run time of your object program; optimization may also reduce the amount of main storage your object program uses. Since OPTIMIZE increases compile time, it should not be used when debugging. The OPTIMIZE option is turned off in the case of a severe-level error or higher. OPTIMIZE and TEST are mutually exclusive. If you use both, OPTIMIZE is ignored.

What is the difference between a subscript and an index?



A subscript is an occurrence value. An index is a displacement value. For example if you have a list of month names each occupying 9 bytes, then the value of a subscript for the month of May would be 5, but the index would be 36. A formula to calculate index value when the occurrence and item length are known is:  $\text{Index-Value} = (\text{Occurrence} - 1) * \text{Item-Length}$