

**COBOL Assessment-Simple**

- 1) What are the three different access methods for accessing records from files ? a  
 (a) Sequential, Random and Dynamic  
 (b) Sequential, Relative and Dynamic  
 (c) Line Sequential, Random and Sequential  
 (d) (d) Dynamic, Random and Line Sequential
- 2) What are the possible File Organization methods available ? P  
 (a) Sequential , Indexed, Random and Sequential Line  
 (b) Sequential , Index, Relative and Line Sequential  
 (c) Sequential , Indexed, Relational and Line Sequential  
 (d) Sequential , Indexed, Relative and Line Sequential
- 3) What are the different kind of files available in COBOL ? Q  
 (a) Sequential Files , Indexed, Relative and Line Sequential  
 (b) Sequential Files , Index, Relative and Line Sequential  
 (c) Sequential Files , Indexed, Relational and Line Sequential  
 (d) Sequential Files , Indexed, Random and Sequential Line
- 4) ACCESS MODE IS RANDOM can be specified for what kind of files ? J  
 (a) Indexed Files  
 (b) Random Files *Relative*  
 (c) Indexed and Random files  
 (d) Sequential , Indexed , Relative and Line Sequential
- 5) ACCESS MODE IS DYNAMIC can be specified for what kind of files ? d  
 (a) Indexed Files  
 (b) Random Files *Relative*  
 (c) Indexed and Random files  
 (d) Sequential , Indexed , Relative and Line Sequential
- 6) ACCESS MODE IS SEQUENTIAL can be specified for what kind of files ? J  
 (a) Indexed Files  
 (b) Random Files  
 (c) Indexed and Random files  
 (d) Sequential , Indexed , Relative and Line Sequential
- 7) When the File organization is Indexed , what clause needs to be used ? b  
 (a) RELATIVE KEY  
 (b) RECORD KEY  
 (c) RELRECORD KEY  
 (d) RECRELATIVE KEY
- 8) When the File organization is Relative , what clause needs to be used ? o  
 (a) RELATIVE KEY  
 (b) RECORD KEY  
 (c) RELRECORD KEY  
 (d) RECRELATIVE KEY
- 9) What is the abbreviation for FD for files in the FILE SECTION ? a  
 (a) File Descriptor  
 (b) File Description  
 (c) File Definition  
 (d) File Declaration
- 10) Match the following:  
 A1) Open Input A2)Permits opening the file for both input and output operations b  
 B1) Open Output B2) Permits opening the file for input operations  
 C1) Open I-O C2)Permits opening the file for output operations  
 D1)Open Extend D2)Permits opening the file for output operations for sequential access files  
 (a) A1-B1,B1-C2,C1-A2,D1-D2  
 (b) A1-B2,B1-C2,C1-A2,D1-D2  
 (c) A1-B2,B1-D2,C1-A2,D1-D2  
 (d) A1-D2,B1-C2,C1-A2,D1-D2

A1-B2, B1-C2, C1-A2, D1-D2

- 11) A \_\_\_\_\_ statement can be executed only for a file in open mode. a  
 (a) CLOSE  
 (b) OPEN  
 (c) WRITE  
 (d) READ
- 12) When the \_\_\_\_\_ Statement is executed, the associated file must be open in INPUT or I-O mode. b  
 (a) WRITE  
 (b) READ  
 (c) OPEN  
 (d) CLOSE
- 13) How can we access the previous record in Indexed and Relative files with Dynamic Access mode. b  
 (a) NEXT RECORD  
 (b) PREVIOUS RECORD  
 (c) READ without PREVIOUS RECORD  
 (d) READ without NEXT RECORD
- 14) How can we establish dynamic access for VSAM Indexed or VSAM Relative organization ? a  
 (a) ACCESS is DYNAMIC in FILE-CONTROL SELECT Statement  
 (b) ACCESS is RANDOM in FILE-CONTROL SELECT Statement  
 (c) ACCESS is DYNAMIC in I-O CONTROL SELECT Statement  
 (d) ACCESS is RANDOM in I-O CONTROL SELECT Statement
- 15) \_\_\_\_\_ statement provides a means of positioning within an indexed or relative file for subsequent sequential record retrieval. a  
 (a) START  
 (b) STAR  
 (c) WRITE  
 (d) READ
- 16) When the \_\_\_\_\_ statement is executed , the associated sequential file must be open in INPUT, OUTPUT or EXTEND MODE and the associated indexed or relative file must be opened in OUTPUT,I-O or EXTEND mode. c  
 (a) OPEN  
 (b) READ  
 (c) WRITE  
 (d) CLOSE
- 17) \_\_\_\_\_ statement is used to update an existing record and the file must be open in OUTPUT mode. b  
 (a) WRITE  
 (b) REWRITE  
 (c) DELETE  
 (d) UPDATE
- 18) \_\_\_\_\_ statement is used to add a record in Indexed and Relative files. b  
 (a) REWRITE  
 (b) WRITE  
 (c) DELETE  
 (d) UPDATE
- 19) \_\_\_\_\_ statement removes the record from an indexed or relative file and the files must be opened in I-O mode. c  
 (a) REWRITE  
 (b) WRITE  
 (c) DELETE  
 (d) UPDATE
- 20) Occurs clause can be specified in the data description entry that has a level number ?  
 (a) 01  
 (b) 66  
 (c) 77  
 (d) None of the above

- 21) Fixed length tables are specified using the \_\_\_\_\_ clause ?  
 (a) OCCURS  
 (b) OCCURS DEPENDING ON  
 (c) DEPENDING  
 (d) DEPENDING ON
- 22) How can we change the contents of an index defined for a table ?  
 (a) OCCURS statement  
 (b) SET statement *(PERFORM K SEARCH)*  
 (c) OCCURS DEPENDING ON statement  
 (d) SET INDEX statement
- 23) The value of an \_\_\_\_\_ is the displacement from the beginning of the table based on the length of the table and Subscript is a positive number whose value indicates the occurrence of a table element.  
 (a) Index ✓  
 (b) Subscript  
 (c) None  
 (d) Index and Subscript
- 24) Match the following :  
 A1) SET A TO B A2) the contents of the receiving field are increased by 1  
 B1) SET A UP BY 1 B2) the value of receiving field is replaced by the value of sending field  
 C1) SET A DOWN BY 1 C2) the contents of the receiving field are decreased by 1  
 D1) SET A TO 1. D2) the index variable A is set to 1  
 (a) A1-B2,B1-A2,C1-C2,D1-D2  
 (b) A2-B1,B1-A2,C1-C2,D1-D2  
 (c) A1-D2,B1-A2,C1-C2,D1-D2  
 (d) A1-C2,B1-A2,C1-C2,D1-D2
- 25) In Serial SEARCH of a table with indexing example , what does HIGH-VALUES mark ?  
 WORKING-STORAGE SECTION.  
 01 WHOLE-TABLE..  
 05 TABLE-ELMT OCCURS NNN TIMES INDEXED BY IDX.  
 10 EMPNO PIC X(05).  
 10 EMPNAME PIC X(75).....  
 PROCEDURE DIVISION...  
 SET IDX TO 1.  
 SEARCH TABLE-ELMT VARYING IDX WHEN EMPNO(IDX) = SEARCH-ARG SET FOUND  
 TO TRUE WHEN EMPNO(IDX) = HIGH-VALUES SET NOT-FOUND TO TRUE  
 END-SEARCH.  
 IF FOUND....END-IF  
 IF NOT-FOUND  
 DISPLAY SEARCH-ARG NOT IN THE TABLE  
 END-IF.
- (a) Beginning of the table  
 (b) End of the table  
 (c) Junk values  
 (d) null values

26) What type of search is being referred ?  
WORKING-STORAGE SECTION.  
01 WHOLE-TABLE.  
05 TABLE-ELMT OCCURS 1 to 100 TIMES DEPENDING ON PARM-VAR ASCENDING KEY IS  
EMPNOINDEXED BY IDX.  
10 EMPNO PIC X(05).  
10 EMPNAME PIC X(75).  
PROCEDURE DIVISION...  
SEARCH ALL TABLE-ELMT VARYING IDX AT END SET NOT-FOUND TO TRUE  
WHEN EMPNO(IDX) = SEARCH-ARG SET FOUND TO TRUE  
END-SEARCH.  
IF FOUND....END-IF  
(a) Serial search  
(b) Bubble sort  
(c) ~~Binary~~ search  
(d) Linear search

27) When we use \_\_\_\_\_, the search index need not to be initialized by SET statement ?

- (a) SEARCH ALL  
(b) SEARCH  
(c) BINARY  
(d) SERIAL

28) What is the maximum levels of nesting possible for Fixed Length of tables ?

- (a) 70  
(b) 7  
(c) 77  
(d) 17

29) In the below code what happens when none of the search conditions are satisfied ?  
WORKING-STORAGE SECTION.

01 WHOLE-TABLE.  
05 TABLE-ELMT OCCURS 1 to 100 TIMES DEPENDING ON PARM-VAR ASCENDING  
KEY IS EMPNO  
INDEXED BY IDX.

10 EMPNO PIC X(05).  
10 EMPNAME PIC X(75).....

PROCEDURE DIVISION...

~~SEARCH ALL~~ TABLE-ELMT VARYING IDX AT END SET NOT-FOUND TO TRUE  
WHEN EMPNO(IDX) = SEARCH-ARG SET FOUND TO TRUE  
END-SEARCH.

IF FOUND....END-IF

- (a) Nothing happens  
(b) Compile time error  
(c) ~~SET NOT-FOUND TO TRUE~~ is executed  
(d) SET FOUND TO TRUE is executed

30) The table has to be sorted either in Ascending Key / Descending Key for a particular key when \_\_\_\_\_ is used .

- (a) SET UP BY  
(b) SEARCH  
(c) ~~SEARCH ALL~~  
(d) SET DOWN BY

31) \_\_\_\_\_ statement is a library statement which places pre-written text in a COBOL program ?  
~~(a) COPY~~  
(b) REPLACE  
(c) ~~COPY WITH REPLACE~~  
~~(d) COPY REPLACING~~

- 32) What is the output compiler code produced?  
 COPY RECORD1.RECORD1 contains:  
 01 RECORD1.  
 05 NAMEPIC X(10).  
 COPY RECORD2.  
 05 DATAPIC X(70).  
 01 RECORD2 contains:-  
 { 05 WS-A PIC X(10).  
 05 WS-B PIC X(10).  
 05 WS-C PIC X(10).  
 05 WS-D PIC X(10). Compiler produces:  
 (a) 01 RECORD1.05 NAME PIC X(10).05 WS-A PIC X(10).05 WS-B PIC X(10).05 WS-C PIC X(10).05 WS-D PIC X(10).05 DATAPIC X(70).  
 (b) 01 RECORD2.05 NAME PIC X(10).05 WS-A PIC X(10).05 WS-B PIC X(10).05 WS-C PIC X(10).05 WS-D PIC X(10).05 DATAPIC X(70).  
 (c) 01 RECORD1.05 NAME PIC X(10).05 WS-A PIC X(10).05 WS-B PIC X(10).05 WS-C PIC X(10).05 WS-D PIC X(10).05 DATAPIC X(70).  
 (d) 01 RECORD1.05 NAME PIC X(10).05 WS-A PIC X(10).05 WS-B PIC X(10).05 WS-C PIC X(10).05 WS-D PIC X(10).05 DATAPIC X(70).01 RECORD2.05 NAME PIC X(10).05 WS-A PIC X(10).05 WS-B PIC X(10).05 WS-C PIC X(10).05 WS-D PIC X(10).05 DATAPIC X(70).
- 33) What will be compiled output generated code in the below example?  
 COPY PAYLIB REPLACING== :A== BY ==WS-A==01 :A:05 :A:-RATEPIC S9(02)V9(02) COMP-3.05 :A:-HRSPIC S9(02)V9(02) COMP-3.
- 34) (a) WS-A.  
 05 WS-A-RATEPIC S9(02)V9(02) COMP.  
 05 WS-A-HRSPIC S9(02)V9(02) COMP.
- 35) (b) WS-A.  
 05 WS-A-RATEPIC S9(02)V9(02) COMP-3.  
 05 WS-A-HRSPIC S9(02)V9(02) COMP-3.
- 36) (c) WS-A.  
 05 WS-A-RATEPIC S9(02)V9(02) COMP.  
 05 WS-A-HRSPIC S9(02)V9(02) COMP.
- 37) (d) WS-A.  
 05 WS-A-RATEPIC S9(02)V9(02) COMP-3.  
 05 WS-A-HRSPIC S9(02)V9(02) COMP-3.
- 38) Match the following LE services ?  
 A1) Storage Management *Language Environment*  
 A2) retrieve / set country, language ,currency and similar attributes  
 B1) Condition Handling B2) obtain and free memory dynamically  
 C1) Messaging services C2)detect errors and other conditions and handle conditions  
 D1) Date and Time services D2) Trigonometric functions, logarithmic and exponential functions  
 E1) Debugging services E2) get and store date and time in various formats  
 F1) Mathematical services F2) define message files , issue messages and routing messages  
 G1) International services G2)retrieve / set error codes, generate dumps , invoke a debug tool  
 (a) G1-A2,A1-B2,B1-C2,F1-D2,D1-E2,C1-F2,E1-D2  
 (b) G1-A2,A1-B2,B1-C2,F1-D2,D1-E2,C1-F2,E1-D2  
 (c) G1-A2,A1-B2,B1-C2,F1-D2,D1-E2,C1-F2,E1-d2  
 (d) G1-A2,A1-B2,B1-C2,F1-D2,D1-E2,C1-F2,E1-G2
- 39) \_\_\_\_\_ statement can cause recursion.  
 (a) REPLACING  
 (b) COPY  
 (c) REPLACE  
 (d) COPY WITH REPLACING
- 40) \_\_\_\_\_ statements are processed by the compiler after any COPY statements are processed  
 (a) REPLACE  
 (b) REPLACING

- (c) COPY WITH REPLACE
- (d) COPY WITH REPLACING

41) The files given / used in MERGE statement must be defined in the \_\_\_\_\_ entry.

- (a) SD
- (b) FD
- (c) SS
- (d) FF

42) \_\_\_\_\_ statement transfers control from one object program to another within the run unit.

- (a) LINK
- (b) XCTL
- (c) CALL
- (d) ATTACH

→ static call

43) Identify the type of CALL statement being executed. CALL PROGAUSING....

- (a) CALL with literal
- (b) CALL with identifier
- (c) CALL with content
- (d) CALL with reference

(default)

44) Identify the type of CALL statement being executed. 01 WS-PROGRAM PIC X(03) VALUE BM. CALL WS-  
PROGRAM USING ....

- (a) CALL with literal → (Static call)
- (b) CALL with content
- (c) CALL with identifier → Dynamic call).
- (d) CALL with reference

45) CALL by \_\_\_\_\_ technique allows the sub-program to access and process the data items in the callers storage.

- (a) ADDRESS
- (b) REFERENCE
- (c) VALUE
- (d) CONTENT

46) CALL by \_\_\_\_\_ technique allows the sub-program to access and process a copy of the data items in the callers storage and the sub-program can change the original data values in the callers storage.

- (a) ADDRESS
- (b) REFERENCE
- (c) VALUE
- (d) CONTENT

47) CALL by \_\_\_\_\_ technique allows the value of the argument to be passed and not a reference to the sending data item. The called program can modify the formal parameter corresponding to the BY VALUE argument but any such changes do not affect the argument since the called program has access to a temporary copy of the sending data item.

- (a) ADDRESS
- (b) REFERENCE
- (c) VALUE
- (d) CONTENT

48) In which section is the data received when we pass data from another external source to a COBOL program ?

- (a) File Section
- (b) Linkage Section
- (c) Working-storage section
- (d) All of the above

49) How can we pass a user parameter to the COBOL program ?

- (a) PARMLIB parameter of a EXEC statement.
- (b) PARM parameter of a JOB statement.
- (c) PARM parameter of a EXEC statement.
- (d) PARM parameter of a DD statement.

50) What is the special register which is used to determine the length of a variable.

- (a) LENGTH OF
- (b) ADDRESS OF
- (c) LENGTH
- (d) ADDRESS

51) When STOP RUN is coded in \_\_\_\_\_ programs , the STOP RUN statement does end the run unit just as in a main program.

- (a) CALLED
- (b) CALLING
- (c) LINKED
- (d) ATTACHED

52) Fill in the blank in the below code.

```
CALLING PROGRAM  
WORKING-STORAGE SECTION.  
01 PARM-LIST.  
05 WS-A PIC X(1).  
05 WS-B PIC X(1).  
PROCEDURE DIVISION.....CALL CALLED-PGM USING _____.STOP RUN.  
CALLED PROGRAM  
LINKAGE SECTION.  
01 USING-LIST.  
10 WS-CPIC X(1).  
10 WS-DPIC X(1).  
PROCEDURE DIVISION USING USING-LIST.....STOP RUN.  
(a) PARM-LIST  
(b) PARM  
(c) WS-A  
(d) WS-B
```

53) INTRINSIC Functions in COBOL allow you to access certain values that are derived

- (a) Compile Time .
- (b) Run Time.
- (c) Start of the Program.
- (d) All of the above.

54) Some INTRINSIC Functions in COBOL are

- (a) Current-Date
- (b) Length
- (c) Lower-case
- (d) Upper-case
- (e) All of the above.

55) INTRINSIC Functions statements in COBOL are coded in

- (a) Data Division
- (b) Procedure Division
- (c) Identification Division
- (d) None of the above
- (e) All of the above

56) Type of date formats that can be used in COBOL date Functions are

- (a) Gregorian Date
- (b) Integer Date
- (c) Julian Date
- (d) None of the above
- (e) All of the above

57) In COBOL date function JULIAN Date Format is

- (a) MMDDYY
- (b) MMDDYYYY

- (c) YYYYMMDD  
(d) YYYYDDMM  
~~(e)~~ YYYYDDD.
- 58) COBOL date functions has these many basic date Formats  
~~(a)~~ 3  
(b) 4  
(c) 1  
(d) Many
- 59) In COBOL date function DDD has a value of  
(a) 365  
~~(b)~~ 366  
(c) 12  
(d) 31
- 60) COBOL UPPER-CASE Intrinsic Function  
~~(a)~~ Change all content to upper case  
(b) Change all content to lower case  
(c) Upper case to lower case  
(d) All of the above
- 61) COBOL LOWER-CASE Intrinsic Function  
(a) Change all to upper case  
~~(b)~~ Change all to lower case  
(c) Lower case to upper case  
(d) All of the above
- 62) WHEN-COMPILED Field populated at the time of  
~~(a)~~ Compilation  
(b) Link Edit step  
(c) Execution of program  
(d) None of the above
- 63) Each Cobol program has to have \_\_\_\_\_ to identify the program.  
(a) Identification Division  
~~(b)~~ Procedure Division  
(c) Environment Division  
(d) None of the above
- 64) The \_\_\_\_\_ describes the computer equipment we are using.  
(a) Identification Division  
(b) Procedure Division  
~~(c)~~ Environment Division  
(d) None of the above
- 65) The \_\_\_\_\_ section and all of its paragraphs are optional.  
~~(a)~~ Configuration Section  
(b) File Section  
~~(c)~~ Both of them  
(d) None of the above
- 66) The \_\_\_\_\_ describes all of the data to be processed by the program.  
(a) Data Division  
(b) Identification Division  
(c) Procedure Division  
(d) None of the above
- 67) As per the Cobol reference format \_\_\_\_\_ position represents the continuation line.  
~~(a)~~ column 7  
(b) Anywhere between column 1 to column 6  
(c) Not required to represent continuation  
(d) First column

- 68) As per the Cobol reference format \_\_\_\_\_ position represents the indicator area.  
(a) First column  
 (b) Anywhere between column 1 to column 6 → Identification area.  
(c) Anywhere between column 1 to column 72  
 (d) column 7
- 69) As per the Cobol reference format , area A is defined from \_\_\_\_\_ position to \_\_\_\_\_ position.  
(a) First column  
 (b) column 8 and column 11  
(c) column 1 and column 7  
(d) column 12 and column 72
- 70) As per the Cobol reference format , area B is defined from \_\_\_\_\_ position to \_\_\_\_\_ position.  
(a) column 1 and column 72  
(b) column 8 and column 11  
(c) column 1 and column 7  
 (d) column 12 and column 72
- 71) As per the Cobol reference format , \_\_\_\_\_ columns can hold the sequence numbers.  
(a) column 1 to column 72  
(b) column 8 to column 11  
 (c) column 1 to column 6  
(d) column 12 to column 72
- 72) As per the Cobol reference format , \_\_\_\_\_ columns can hold the program designation.  
(a) column 1 to column 8  
 (b) column 73 to column 80  
(c) column 8 to column 11  
(d) column 12 to column 72
- 73) Which of the below can be denoted in the indicator Area.  
(a) continuation  
(b) debugging and documentation  
 (c) All of the above  
(d) none of the above
- 74) In which area a continuation lines must begin?  
(a) area A  
 (b) area B  
(c) Any of the above  
(d) none of the above
- 75) In working-Storage section, \_\_\_\_\_ specifies hierarchy of data within the record.  
 (a) Level number  
(b) Column number  
(c) Line number  
(d) None of the above
- 76) A level-number has a value taken from one of the set of integers. Select the appropriate one.  
 (a) 1 to 49, 66,77,88  
(b) 1 to 66,77,88  
(c) 1 to 88  
(d) 1 to 99
- 77) A \_\_\_\_\_ item can not be further sub-divided.  
(a) Group item  
 (b) Elementary item  
(c) All the items  
(d) None of the above
- 78) An 01 level may be \_\_\_\_\_ item.  
(a) Subordinate  
(b) group  
(c) elementary

- 79) A set of elementary items can be called as \_\_\_\_\_.  
(a) Subordinate  
 (b) group  
(c) group of elementary items  
(d) group or elementary
- 80) All data descriptions must end with \_\_\_\_\_.  
(a) space  
(b) comma  
 (c) separator period  
(d) colon or semicolon
- 81) Which of the below can be used to separate clauses.  
(a) Space only  
(b) semicolon or colon  
 (c) space and comma only  
(d) space, comma or semicolon
- 82) A data-name must be specified for \_\_\_\_\_ level numbers.  
(a) 66 and 77  
(b) 66 and 88  
(c) 77 and 88  
 (d) 66, 77 and 88
- 83) A data item \_\_\_\_\_ changed during program execution.  
 (a) can not  
(b) can be  
(c) should not  
(d) none of the above
- 84) Based on the below example, choose the correct.
- If EMP-NAME from IN-REC to be moved to EMP-NAME of OUT-REC  
05 IN-REC.  
    15 EMP-NAME PIC X(20)...  
05 OUT-REC.  
    15 EMP-NAME PIC X(20)...  
(a) MOVE EMP-NAME TO EMP-NAME  
(b) MOVE IN-REC(EMP-NAME) TO OUT-REC(EMP-NAME)  
 (c) MOVE EMP-NAME IN IN-REC TO EMP-NAME IN OUT-REC  
(d) MOVE IN-REC TO OUT-REC
- 85) \_\_\_\_\_ defines the program.  
(a) Sections  
(b) Paragraphs  
 (c) Both of them  
(d) None of them
- 86) A paragraph name must be followed by \_\_\_\_\_.  
 (a) A Separator period.  
(b) space  
(c) hyphen -  
(d) comma ,
- 87) A data name cannot be the same as \_\_\_\_\_ name.  
(a) Section name  
(b) Paragraph name  
 (c) Section or paragraph name  
(d) Can be any name
- 88) If data name is omitted, it is treated as \_\_\_\_\_.  
 (a) Filler

- (b) Any data item with group name as qualifier  
(c) Syntactically wrong  
(d) None
- 89) The data-item or FILLER must immediately follow \_\_\_\_\_ in working-storage section.  
(a) group name  
(b) a separator period  
 (c) level number  
(d) None of the above
- \* 90) With \_\_\_\_\_ you can have non-unique data-names.  
(a) Filler  
 (b) qualifiers  
(c) identifiers  
(d) None of the above
- 91) Qualification is specified by placing \_\_\_\_\_ followed by a qualifier.  
(a) parenthesis  
(b) IN  
(c) OF  
 (d) IN or OF
- 92) It is \_\_\_\_\_ necessary to specify all the levels of hierarchy.  
 (a) not always  
(b) always  
(c) not  
(d) None
- 93) In any hierarchy the highest level \_\_\_\_\_ qualified.  
 (a) can not be  
(b) should be  
(c) can be  
(d) None
- 94) In Reference Modification, the data name must have usage \_\_\_\_\_.  
(a) Pointer  
 (b) Display  
(c) Comp-3  
(d) None of the above
- 95) PICTURE clause can be specified only at \_\_\_\_\_.  
(a) Elementary level  
(b) Group level  
 (c) All the levels  
(d) Conditional level (88)
- 96) PICTURE IS \$99999.99 is same as \_\_\_\_\_.  
 (a) PIC IS \$9(5).9(2)  
 (b) PIC IS \$9(5)v9(2)  
(c) PIC IS \$S9(7)  
(d) none
- 97) PICTURE clause is not allowed for \_\_\_\_\_.  
(a) Index data items  
(b) Subscripts  
 (c) Both of them  
 (d) None
- 98) USAGE POINTER data items do not have \_\_\_\_\_ clause.  
 (a) PICTURE  
(b) POINTER  
(c) none
- 99) Which of the below do not have PICTURE clause.  
 (a) USAGE IS POINTER

- (b) USAGE IS OBJECT  
(c) Both of them  
(d) None of them
- 100) The PICTURE clause is not allowed for which of the below data items.  
 (a) Internal floating-point data items  
(b) all numeric edited data items  
(c) all alphanumeric data items  
 (d) None of them
- 101) Reference data items \_\_\_\_\_ PICTURE clause.  
 (a) can have  
(b) can not have  
(c) must have  
(d) may have
- 102) V in Picture clause indicates \_\_\_\_\_  
(a) Alphabet  
 (b) Location of assumed decimal point  
(c) Version of COBOL  
(d) Space
- 103) B in Picture clause indicates \_\_\_\_\_  
(a) Alphabet  
(b) Binary  
 (c) Space  
(d) DBCS
- 104) X indicates \_\_\_\_\_  
(a) Alphabetic  
(b) Numeric  
(c) Space  
 (d) Alpha-Numeric
- 105) CR or DB indicates in PICTURE clause \_\_\_\_\_  
 (a) Editing sign symbols  
(b) Currency  
(c) Simply the word CR or DB no meaning to that  
(d) Alpha-Numeric
- 106) If an USAGE Clause is not specified it is assumed that the usage is \_\_\_\_\_.  
(a) BINARY  
 (b) DISPLAY  
(c) COMP  
(d) NO USAGE CLAUSE
- 107) The USAGE Clause can be specified for a data description entry with a level-number other than \_\_\_\_\_  
(a) 66  
(b) 88  
 (c) 66 or 88 or 17  
(d) 77 or 88
- 108) A BINARY Half word occupies \_\_\_\_\_ bytes of memory.  
 (a) 4.  
(b) 4.  
(c) 1.  
(d) 8.
- 109) A BINARY Full word occupies \_\_\_\_\_ bytes of memory.  
(a) 2.  
 (b) 4.  
(c) 1.  
(d) 8.

- 110) A BINARY Double word occupies \_\_\_\_\_ bytes of memory.  
(a) 2.  
(b) 4.  
(c) 1.  
~~(d)~~ 8.
- 111) Data is stored in char form as 8 bits for one char in \_\_\_\_\_ usage.  
(a) DISPLAY  
(b) BINARY  
~~(c)~~ COMP-3  
(d) None
- 112) A data item can not contain a VALUE if the prior item contains a \_\_\_\_\_ with \_\_\_\_\_.  
(a) DBCS and SPACE  
(b) External floating-point  
(c) ZERO and COMP-1  
~~(d)~~ OCCURS and DEPENDING ON
- 113) A VALUE clause for group entries must not be specified if the entry also contains \_\_\_\_\_.  
(a) JUSTIFIED  
(b) SYNCHRONIZED  
(c) USAGE with other than DISPLAY  
~~(d)~~ All the above
- 114) COMP 3 data is stored internally as \_\_\_\_\_.  
(a) Binary  
(b) Half word  
(c) Full word  
~~(d)~~ Packed decimal
- 115) \_\_\_\_\_ Clause specifies the initial contents of a data item.  
(a) Filler  
~~(b)~~ Picture  
~~(c)~~ Value  
(d) None
- 116) If the data item is not defined with initial value with VALUE, then its value is \_\_\_\_\_.  
~~(a)~~ Spaces for alphabetic and Zeros for numerals  
(b) Unpredictable  
(c) Spaces  
(d) None
- 117) VALUE must not be specified if the entry has \_\_\_\_\_.  
~~(a)~~ JUSTIFIED  
(b) Filler  
~~(c)~~ 88 Level  
(d) None
- 118) ALL literal must not be used with \_\_\_\_\_ statements.  
~~(a)~~ CALL  
(b) INSPECT  
~~(c)~~ All of them  
(d) None
- 119) Group items are in \_\_\_\_\_ class.  
(a) Alpha-Numeric  
(b) Numeric  
~~(c)~~ Depends on Subordinate items  
(d) Alphabetic
- 120) The class of elementary items are determined by \_\_\_\_\_ clause.  
(a) Value  
~~(b)~~ Picture  
(c) Numeric  
(d) None

Numeric - 18  
Alpha Nu - 32767  
Alphabetic - 32767

- 121) A numeric data item has maximum of \_\_\_\_\_ digits.  
 (a) 18  
 (b) 9  
 (c) No limit  
 (d) 24
- 122) Level numbers should be same for \_\_\_\_\_ clause.  
 (a) Group  
 (b) Redefines  
 (c) Renames  
 (d) Filler
- 123) 88 Level numbers \_\_\_\_\_ redefined.  
 (a) can not be  
 (b) can be  
 (c) may be  
 (d) None
- 124) REDEFINES allows different data descriptions to use \_\_\_\_\_ storage area.  
 (a) Same  
 (b) different  
 (c) Limited  
 (d) None
- 125) The PROCEDURE DIV is \_\_\_\_\_ in a COBOL source program.  
 (a) Must  
 (b) Optional
- 126) In Hexadecimal notation, the character allowed are \_\_\_\_\_.  
 (a) 0 to 9  
 (b) a thru f  
 (c) A thru F  
 (d) All of the above
- 127) The identifiers in ADD statement must be \_\_\_\_\_ items.  
 (a) Numeric  
 (b) Alphabetic  
 (c) Alphanumeric  
 (d) All of the above
- 128) The result is truncated when \_\_\_\_\_ not given.  
 (a) ADD  
 (b) Rounded  
 (c) Giving  
 (d) None of them
- 129) In \_\_\_\_\_ arithmetic operation, the rounded has no effect.  
 (a) Numeric  
 (b) floating - point  
 (c) Numeric edited  
 (d) DBCS
- 130) The result of a floating point operation is always \_\_\_\_\_.  
 (a) Rounded  
 (b) Truncated  
 (c) Neither Truncated nor Rounded  
 (d) None
- 131) When the size of the fractional result exceeds the number of places of the result, then \_\_\_\_\_ occurs unless ROUNDED is specified.  
 (a) Rounding  
 (b) Truncation  
 (c) movement of Zeros  
 (d) None

132)

The identifiers are

- (a) Group \_\_\_\_\_ items in MOVE statement.
- (b) Elementary
- (c) either group or elementary
- (d) either group or elementary

133)

The identifiers must be \_\_\_\_\_ items in ADD CORRESPONDING.

- (a) Group
- (b) Elementary
- (c) Numeric
- (d) None

134)

Truncation of the result will occur if \_\_\_\_\_

- (a) On Size Error \_\_\_\_\_ is not specified
- (b) Rounded
- (c) Truncation
- (d) None

135)

Which of the below compiler options can affect the use of numeric data items.

- (a) Numeric
- (b) Trunc
- (c) Binary
- (d) All the above

136)

In ADD the PIC character-string can contain the symbols \_\_\_\_\_

- (a) P, S, V
- (b) P & s
- (c) 9, P, S & V
- (d) None

137)

\_\_\_\_\_ are the building blocks of PROCEDURE DIVISION.

- (a) statements
- (b) Paragraphs
- (c) Sections
- (d) None of the above.

138)

Which of the below is correct form of MULTIPLY?

- (a) MULTIPLY VAR1, VAR2 BY VAR3
- (b) MULTIPLY vAR1 GIVING vAR2 BY vAR3
- (c) MULTIPLY BY vAR1
- (d) MULTIPLY vAR1 BY vAR2 GIVING vAR3

139)

Which is correct if the statements is given as below? MULTIPLY VAR1 BY VAR2 GIVING VAR3

- (a) VAR1 & VAR2 shall be elementary numeric items
- (b) VAR1 & VAR2 shall be elementary alpha numeric item
- (c) Either of them should be numeric items
- (d) None

140)

Which of the below DIVIDE is correct?

- (a) DIVIDE VAR1 INTO VAR2
- (b) DIVIDE VAR1 BY VAR2, VAR3
- (c) DIVIDE VAR1, VAR2 ON SIZE ERROR
- (d) DIVIDE VAR1 VAR2 BY VAR3 VAR4

141)

Which data has quotient in the DIVIDE statement? DIVIDE VAR1 INTO VAR2 GIVING VAR3  
REMAINDER VAR4.

- (a) VAR1
- (b) VAR2
- (c) VAR3
- (d) VAR4

142)

COMPUTE statement is \_\_\_\_\_ the separate arithmetic statements.

- (a) More efficient than
- (b) same as

- (c) less efficient than  
(d) None
- 143) Please justify the below is correct or not. COMPUTE AREA = LENGTH \* WIDTH END-COMPUTE.  
(a) correct  
(b) = not allowed  
✓(c) END-COMPUTE is not allowed here  
(d) END-COMPUTE is allowed in the next line.
- 144) \_\_\_\_\_ statement transfers control explicitly to one or more procedures.  
(a) COMPUTE  
✓(b) PERFORM  
(c) CALL  
(d) CALCULATE
- 145) An in-line PERFORM must be delimited by \_\_\_\_\_ phrase.  
(a) END  
✓(b) END-PERFORM  
(c) EXIT-PERFORM  
(d) END-CONTROL
- 146) An in-line PERFORM \_\_\_\_\_ a period inside.  
(a) can have  
✓(b) can not have  
(c) may have  
(d) must have
- 147) What are the possible declarations of File Status variable in COBOL ?
- (a) PIC X(02) OR PIC XX OR PIC 9(02) OR PIC 99.  
 (b) PIC XX  
 (c) PIC 9(02).  
 (d) PIC 999.