

100 COBOL interview questions to ask your applicants

Questions

1. What is COBOL? Can you explain it like you're explaining it to a five-year-old?
2. What are the four divisions in a COBOL program and what does each one do?
3. Explain what a 'data type' is in COBOL using an example.
4. What's the difference between a 'PIC' clause and a 'VALUE' clause in COBOL?
5. What is the purpose of the 'WORKING-STORAGE SECTION'?
6. How do you move data from one variable to another in COBOL?
7. What is an 'IF' statement in COBOL, and how does it work?
8. Explain what a 'loop' is and how you might create one in COBOL (e.g., using PERFORM).
9. What does 'compiling' a COBOL program mean?
10. What is the purpose of the IDENTIFICATION DIVISION?
11. Can you describe a simple COBOL program that adds two numbers together?
12. What are literals in COBOL?
13. What is the sequence of execution in a COBOL program?
14. What is a paragraph in COBOL, and how is it different from a section?
15. How do you display output to the screen in COBOL?
16. Explain the purpose of the ENVIRONMENT DIVISION.
17. What are some common COBOL verbs you know, and what do they do (e.g., ADD, SUBTRACT, MULTIPLY, DIVIDE)?
18. What is structured programming, and why is it important in COBOL?
19. What is the purpose of 'PICTURE' clause in data definition?
20. What is the difference between numeric and alphanumeric data types in COBOL?
21. What is the use of COPY statement in COBOL?
22. What is the ACCEPT statement used for?
23. Describe how you would read data from a file in COBOL.
24. What are the different types of files in COBOL, such as sequential, indexed, and relative files?
25. How do you handle errors in COBOL programs?
26. Explain the significance of level numbers in COBOL data structures.
27. What is COBOL, in the simplest terms you can imagine?
28. Can you describe the basic structure of a COBOL program?
29. What are the four divisions in a COBOL program, and what does each one do?
30. Explain what a 'paragraph' is in COBOL and why we use them.
31. What's the difference between a 'literal' and a 'variable' in COBOL?
32. How do you declare a variable in COBOL, and what's the purpose of 'PICTURE' clause?
33. What are some common data types used in COBOL (e.g., alphanumeric, numeric)?
34. Explain the purpose of the 'MOVE' statement in COBOL.
35. How do you perform simple arithmetic operations (addition, subtraction) in COBOL?
36. What is the purpose of the 'DISPLAY' statement?
37. How do you compare two values in COBOL (e.g., using 'IF' statement)?
38. Explain what a 'conditional statement' is and how it's used in COBOL.
39. What's the difference between 'IF' and 'EVALUATE' statements?
40. How do you read data from a file in COBOL?
41. Explain the purpose of the 'OPEN' and 'CLOSE' statements.
42. What is a sequential file in COBOL?
43. Can you give an example of how you would write data to a file?
44. What does 'debugging' mean, and how would you approach debugging a simple COBOL program?
45. What are some common COBOL keywords you know, and what do they do?
46. Explain the purpose of comments in a COBOL program and how to write them.
47. What is the purpose of 'PERFORM' statement? Explain it with example
48. Describe what a COBOL compiler does.
49. What is the purpose of 'WORKING-STORAGE SECTION'?
50. How do you define a table (array) in COBOL?
51. What is the role of 'IDENTIFICATION DIVISION'?
52. Explain what 'STRING' and 'UNSTRING' statements are used for.
53. How would you optimize a COBOL program that's running slowly, focusing on file I/O operations?
54. Explain the use of OCCURS clause with DEPENDING ON phrase and how it impacts memory allocation.
55. Describe a scenario where you would use a subprogram in COBOL and how you would pass data to it.
56. How do you handle errors when reading a sequential file in COBOL and what are the different error codes you might encounter?
57. What are the differences between intrinsic functions and user-defined functions in COBOL, and when would you use each?
58. Explain the significance of the VALUE clause in COBOL and provide examples of its usage with different data types.
59. Describe the process of debugging a COBOL program using a debugging tool or technique you are familiar with.
60. How would you implement a binary search algorithm in COBOL to find a specific record in a sorted table?
61. What are the advantages and disadvantages of using indexed files in COBOL, and how do you create and maintain them?
62. Explain the purpose of the REDEFINES clause in COBOL and provide an example of how it can be used to manipulate data.
63. How do you handle date and time calculations in COBOL, including converting between different date formats?
64. Describe a situation where you would use the STRING and UNSTRING statements in COBOL and explain their functionality.
65. What are the different types of COBOL verbs used for conditional execution, and how do they differ in their behavior?
66. Explain the concept of scope in COBOL and how it affects the visibility and accessibility of variables and data items.
67. How do you interact with external systems or databases from a COBOL program, such as using SQL or other interfaces?
68. Describe the process of converting a COBOL program to run on a different platform or operating system.
69. What are the different types of COBOL sections in the DATA DIVISION and what is the purpose of each section?
70. Explain the concept of CALL statement in COBOL and how it's related to program modularity?
71. How can you sort the records within a COBOL program?
72. Explain how to perform calculations with decimal data in COBOL and prevent rounding errors.
73. Describe how to implement a simple menu-driven interface in COBOL to interact with the user.
74. How to use INSPECT statement to analyze and modify strings.
75. Explain the advantages of using structured programming techniques in COBOL development.
76. How do you handle file locking and concurrency issues in a COBOL application?
77. Describe how to perform error handling in COBOL using EVALUATE statement.
78. How do you integrate COBOL with modern technologies such as web services or APIs?
79. Explain a complex COBOL program you designed, focusing on how you optimized it for performance and maintainability.
80. Describe your experience with different COBOL compilers and their impact on code execution.
81. How do you approach debugging a large, legacy COBOL system with limited documentation?
82. Explain your experience with integrating COBOL applications with modern technologies like web services or databases.
83. Describe a time you had to rewrite a significant portion of a COBOL application. What challenges did you face?
84. How do you ensure code quality and adherence to standards in a COBOL development environment?
85. Explain your experience with different file handling techniques in COBOL, such as sequential, indexed, and relative files.
86. Describe your experience with using COBOL in a distributed processing environment.
87. How do you handle error conditions and exceptions in COBOL programs to ensure data integrity?
88. Explain your understanding of the COBOL CALL statement and its usage in modular programming.
89. Describe your experience with performance tuning of COBOL applications, including identifying and resolving bottlenecks.
90. How do you approach the task of migrating a COBOL application to a different platform or environment?
91. Explain your experience with using COBOL precompilers for database access or other specialized functions.
92. Describe a situation where you had to troubleshoot a performance issue in a COBOL batch processing job.
93. How do you stay up-to-date with the latest trends and best practices in COBOL development?
94. Explain your experience with using COBOL to process large volumes of data, such as in a data warehousing environment.
95. Describe your experience in leading a COBOL development team.
96. How do you handle data conversion issues when integrating COBOL systems with other platforms?
97. Explain how you use structured programming techniques to write maintainable COBOL code.
98. Describe your experience with using COBOL in a real-time transaction processing environment.