



1. Does every ABAP/4 have a modular structure?

Yes.

2. What is Modularization and its benefits?

If the program contains the same or similar blocks of statements or it is required to process the same function several times, we can avoid redundancy by using modularization techniques. By modularizing the ABAP/4 programs we make them easy to read and improve their structure. Modularized programs are also easier to maintain and to update.

3. Name the ABAP/4 Modularization techniques.

- Source code module.
- Subroutines.
- Functions.

4. How can we create callable modules of program code within one ABAP/4 program?

- By defining Macros.
- By creating include programs in the library.

5. M is the attribute type of the module program.

6. Is it possible to pass data to and from include programs explicitly?

No. If it is required to pass data to and from modules it is required to use subroutines or function modules.

7. What are subroutines?

Subroutines are program modules, which can be called from other ABAP/4 programs or within the same program.

8. What are the types of Subroutines?

- Internal Subroutines: The source code of the internal subroutines will be in the same ABAP/4 program as the calling procedure (internal call).



- External Subroutines: The source code of the external subroutines will be in an ABAP/4 program other than the calling procedure.

9. It is not possible to create an ABAP/4 program, which contains only Subroutines (T/F).

False.

10. A subroutine can contain nested form and endform blocks. (T/F)

False.

11. Data can be passed between calling programs and the subroutines using Parameters.

12. What are the different types of parameters?

Formal Parameters: Parameters, which are defined during the definition of subroutine with the FORM statement.

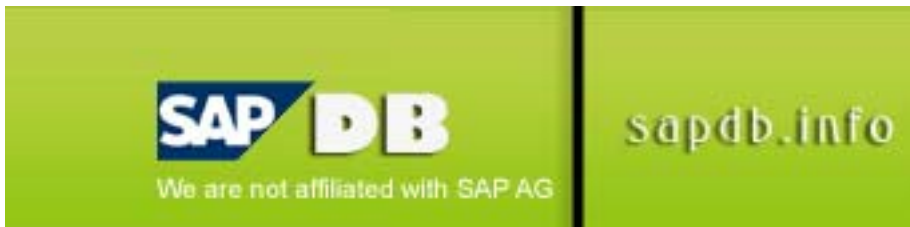
Actual Parameters: Parameters which are specified during the call of a subroutine with the PERFORM statement.

13. How can one distinguish between different kinds of parameters?

- Input parameters are used to pass data to subroutines.
- Output parameters are used to pass data from subroutines.

14. What are the different methods of passing data?

- Calling by reference: During a subroutine call, only the address of the actual parameter is transferred to the formal parameters. The formal parameter has no memory of its own, and we work with the field of the calling program within the subroutine. If we change the formal parameter, the field contents in the calling program also changes.
- Calling by value: During a subroutine call, the formal parameters are created as copies of the actual parameters. The formal parameters have memory of their own. Changes to the formal parameters have no effect on the actual parameters.
- Calling by value and result: During a subroutine call, the formal parameters are created as copies of the actual parameters. The formal parameters have their own memory space. Changes to the formal parameters are copied to the actual parameters at the end of the subroutine.



15. The method by which internal tables are passed is By Reference.

16. How can an internal table with Header line and one without header line be distinguished when passed to a subroutine?

Itab [] is used in the form and endform if the internal table is passed with a header line.

17. What should be declared explicitly in the corresponding ABAP/4 Statements to access internal tables without header lines & why?

Work Area. This is required as the Work Area is the interface for transferring data to and from the table.

18. A subroutine can be terminated unconditionally using EXIT. (T/F)

True.

19. A subroutine can be terminated upon a condition using CHECK Statement.

20. Function Modules are also external Subroutines. (T/F).

True.

21. What is the difference between the function module and a normal ABAP/4 subroutine?

In contrast to normal subroutines function modules have uniquely defined interface. Declaring data as common parts is not possible for function modules. Function modules are stored in a central library.

22. What is a function group?

A function group is a collection of logically related modules that share global data with each other. All the modules in the group are included in the same main program. When an ABAP/4 program contains a CALL FUNCTION statement, the system loads the entire function group in with the program code at runtime. Every function module belongs to a function group.

23. What is the disadvantage of a call by reference?

During a call by reference damage or loss of data is not restricted to the subroutine, but will instantly lead to changes to the original data objects.

24. A function module can be called from a transaction screen outside an ABAP/4 program. (T/F).



True.

25. What is an update task?

It is an SAP provided procedure for updating a database.

26. What happens if a function module runs in an update task?

The system performs the module processing asynchronously. Instead of carrying out the call immediately, the system waits until the next database update is triggered with the 'COMMIT WORK' command.

27. The function modules are created and stored in the Function Library.

28. When a function module is activated syntax checking is performed automatically. (Y/N)

True.

29. What is the use of the RAISING exception?

The raising exception determines whether the calling program will handle the exception itself or leave the exception to the system.

30. What is the difference between internal tables and extract datasets?

- The lines of an internal table always have the same structure. By using extract datasets, you can handle groups of data with different structure and get statistical figures from the grouped data.
- You have to define the structure of the internal table at the beginning. You need not define the structure of the extract dataset.
- In contrast to internal tables, the system partly compresses exact datasets when storing them. This reduces the storage space required.
- Internal tables require special work area for interface whereas extract datasets do not need a special work area for interface.

31. It is possible to assign a local data object defined in a subroutine or function module to a field group. (T/F).

False.

32. What is the difference between field-group header and other field groups?



The header field group is a special field group for the sort criteria. The system automatically prefixes any other field groups with the header field group.

33. Can a field occur in several field groups.

Yes. But it leads to unnecessary data redundancy.

34. When sorting the extract dataset the fields used as default sort key lie in the Header field group.

35. What does the insert statement in extract datasets do?

It defines the fields of a field group.

36. What does the extract statement do in extract datasets?

The data is written to virtual memory by extract commands.

37. A field-groups statement or an insert statement reverses storage space and transfers values. (T/F).

False.

38. While using extract datasets it is required to have a special workarea for interface (T/F)

False.

39. The LOOP-ENDLOOP on extract datasets can be used without any kind of errors (T/F)

False. It causes runtime errors.

40. The Maximum no of key fields that can be used in a header is 50.

41. While sorting field groups we cannot use more than one key field (T/F).

False.

42. While sorting, if the main storage available is not enough, the system writes data to an external help file. The SAP profile parameter, which determines this help file, is DIR_SORTTMP.

-

43. The extract statements in field groups can be used before or after processing the sort statements. (T/F)

FALSE.



We are not affiliated with SAP AG

sapdb.info