Capgemini

Modularization Techniques - Subroutines

Lesson Objectives

In this lesson, participants will learn

- Subroutines

Modularization Techniques

A modularization unit is a part of a program that encapsulates a particular function.

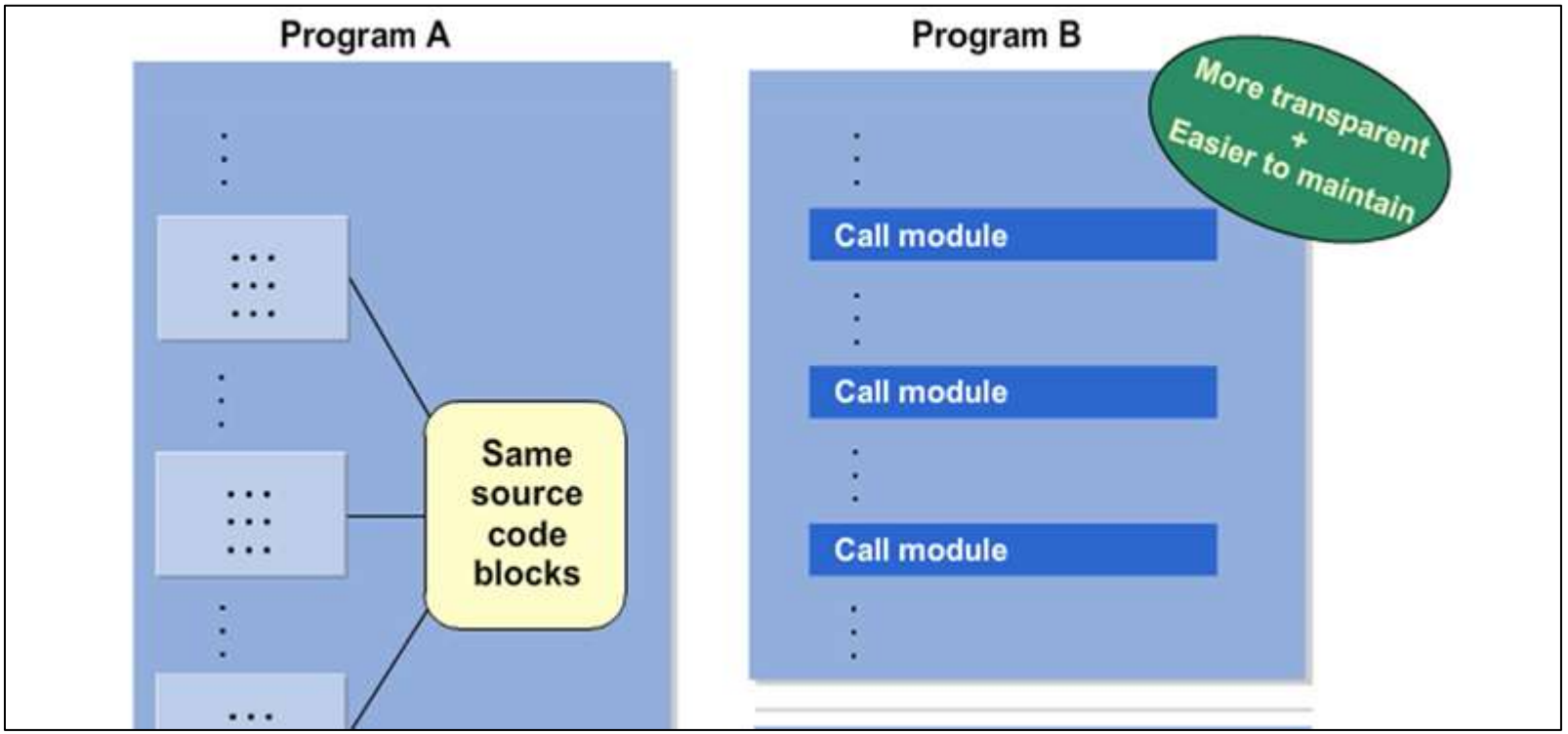
Part of the source code is stored in a module to improve the transparency of the program, as well as to use the corresponding function in the program several times without having to re-implement the entire source code on each occasion.

The improvement in transparency is a result of the program becoming more function-oriented.

It divides the overall task into sub-functions, which are the responsibility of corresponding modularization units.

Modularization makes it easier to maintain programs, because you only need to make changes to the function or corrections in the respective modularization units, and not at various points in the main program.

Modularization Techniques

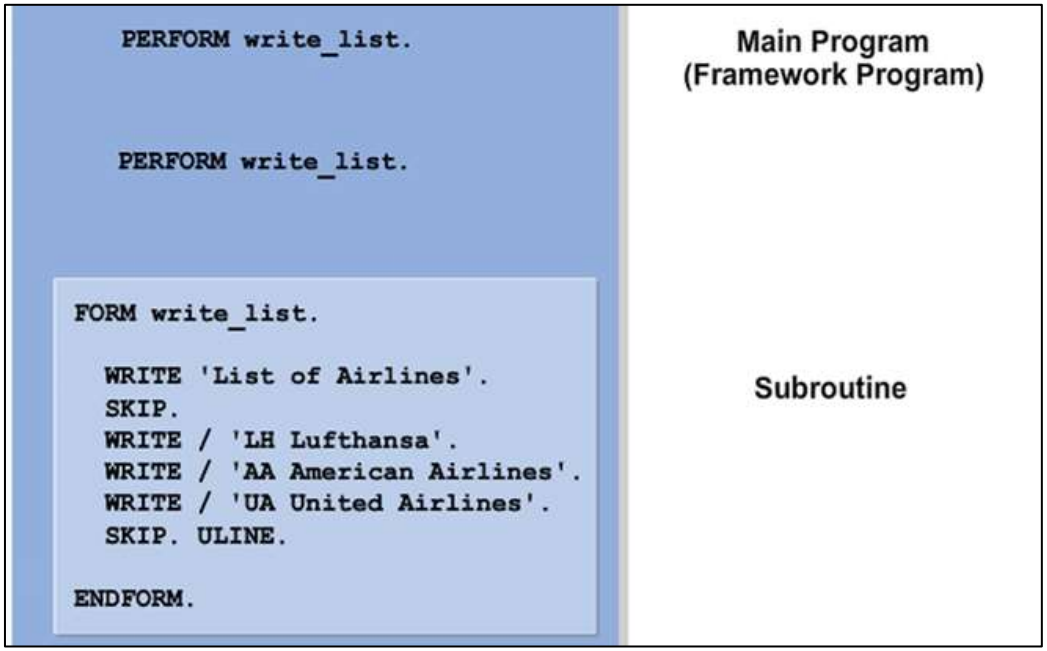


Modularization - Subroutines

Subroutines can be defined in any ABAP program.

A subroutine is a modularization unit within a program.

Subroutines are used to write functions that are used repeatedly within a program.



Modularization - Subroutines

A subroutine is a block of code introduced by FORM and concluded by ENDFORM.

Syntax

FORM <SUBR> [USING ... [VALUE(]<pi>[)] [TYPE <t>|LIKE <f>]...]
[CHANGING... [VALUE(]<pi>[)] [TYPE <t>|LIKE <f>]...].

...

ENDFORM.

- <SUBR> is the name of the subroutine.

Modularization - Subroutines

Calling Subroutines

- PERFORM…… [USING ... <pi>...] [CHANGING... <pi>...].

Subroutine - Example

```
REPORT  Z NO STANDARD PAGE HEADING.
WRITE : / 'Before calling the subroutine'.
PERFORM SUB1.
WRITE : / 'After calling the subroutine'.

FORM SUB1.
  WRITE : / 'Inside sub1'.
ENDFORM.
```

Demo: Create Simple Subroutine

```
1  REPORT  Z NO STANDARD PAGE HEADING.
2  WRITE : / 'Before calling the subroutine'.
3  PERFORM SUB1.
4  WRITE : / 'After calling the subroutine'.
5
6  FORM SUB1.
7    WRITE : / 'Inside sub1'.
8  ENDFORM.
```

Output after execution

```
Before calling the subroutine
Inside sub1
After calling the subroutine
```

Parameter Definition for subroutines

Variables defined in the main program are globally visible within the program and can be changed at any point within the program.

This means that also subroutines defined within the program can change the value of these global variables.

```
REPORT Z.
* a and b are global variables
DATA: a TYPE i value 10,
      b TYPE i value 20.

PERFORM sub.
WRITE: 'After invoking subroutine'.
WRITE : /'Value of a is:' , a.
WRITE : /'Value of b is:' , b.
FORM sub.
  a = 11.
  b = 22.
ENDFORM.
```