Modularization Technique

Modularization is breaking down the application code into smaller units, so that it is easy to maintain. Suppose if we want to implement the same logic like adding two numbers in several places of the same program, then write the logic inside a modularization unit and call the modularization unit wherever we want to add two numbers.

Even if we want to declare the same variables in multiple programs or use the same logic in different programs then code the common part in the modularization unit and use it in different programs.

Advantages of Modularization are as follows.

- Easier to read, maintain and debug.
- Eliminates redundancy of code and increases reusability of code.

Different types of Modularization units called from ABAP programs.

- **Macros** -- If you want to reuse the same set of statements more than once in a program, you can include them in a macro. You can only use a macro within the program in which it is defined.
- **Include Programs** -- Include programs allow you to use the same source code in different programs. They are mainly used for modularizing source code and have no parameter interface.
- **Subroutines** -- Subroutines are normally called internally i.e. called from the same program in which it is declared. But subroutines can also be called from external programs.
- **Function Modules** -- Function Modules are stored in the central library and can be called from any program. Even the function modules(RFC enabled) can be called from non-SAP systems.
- **Methods** -- Methods used ABAP object oriented programming

Different types of Modularization units called from ABAP runtime.

- **Event Blocks** -- Event blocks begin with a event code and ends with next processing block.
- **Dialog Modules** -- Dialog Modules are used in PBO and PAI events of Module Pool programs.