Logical Expression

1. **AND**
2. **OR**
3. **NOT**

```
COUNTER GE 0
```

```
COUNTER GT 0 AND
FLAG IS INITIAL
```

```
( FLAG1 NE SPACE OR FLAG2
NE SPACE ) AND COUNTER
BETWEEN 0 AND 100
```

```
COUNTER EQ 0 AND NOT
( FLAG1 EQ SPACE AND FLAG2
EQ SPACE )
```

IF Statement

```
IF <logical expression>.
        statements
ENDIF.
```

```
IF <logical expression>.
        statements
ELSE.
        statements
ENDIF.
```

```
IF <logical expression>.
        statements
ELSEIF <logical expression>.
        statements
ELSEIF <logical expression>.
        statements
ELSE.
        statements
ENDIF.
```

CASE statement

- **Execute different statement blocks depending on the contents of particular data fields**
- **WHEN OTHERS:**
  - **contents of <field> does not equal to any of the <value?> contents**

```
CASE <field>.
  WHEN <value1>.
    statements
  WHEN <value2>.
    statements
  WHEN <value3>.
    statements
  WHEN OTHERS.
    statements
ENDCASE.
```

ABAP statement and Keyword(con't)

CHECK statement

```
CHECK <logical expression>.
```

1. CHECK within a loop structure.

- Within Loop:
  to terminate loop
  pass conditionally

```
WHILE COUNTER GT 0.
    statements
    CHECK FLAG NE SPACE.
    statements
ENDWHILE.
```

2. CHECK outside loop structures.

```
statements
CHECK <condition>.
statements
```

ABAP statement and Keyword(con't)

- Terminate loop entirely without any condition
- Terminate a subroutine without any condition

```
EXIT.
```

1 `EXIT` within a loop structure.

```
DO ...
      statements
   IF COUNTER GE 100.
      EXIT.
   ENDIF.
      statements

ENDDO.
```

2 `EXIT` outside loop structure.

```
      statements
IF ... .
   EXIT.
ENDIF
      statements
```

ABAP statement and Keyword(con't)

```
CONTINUE.
```

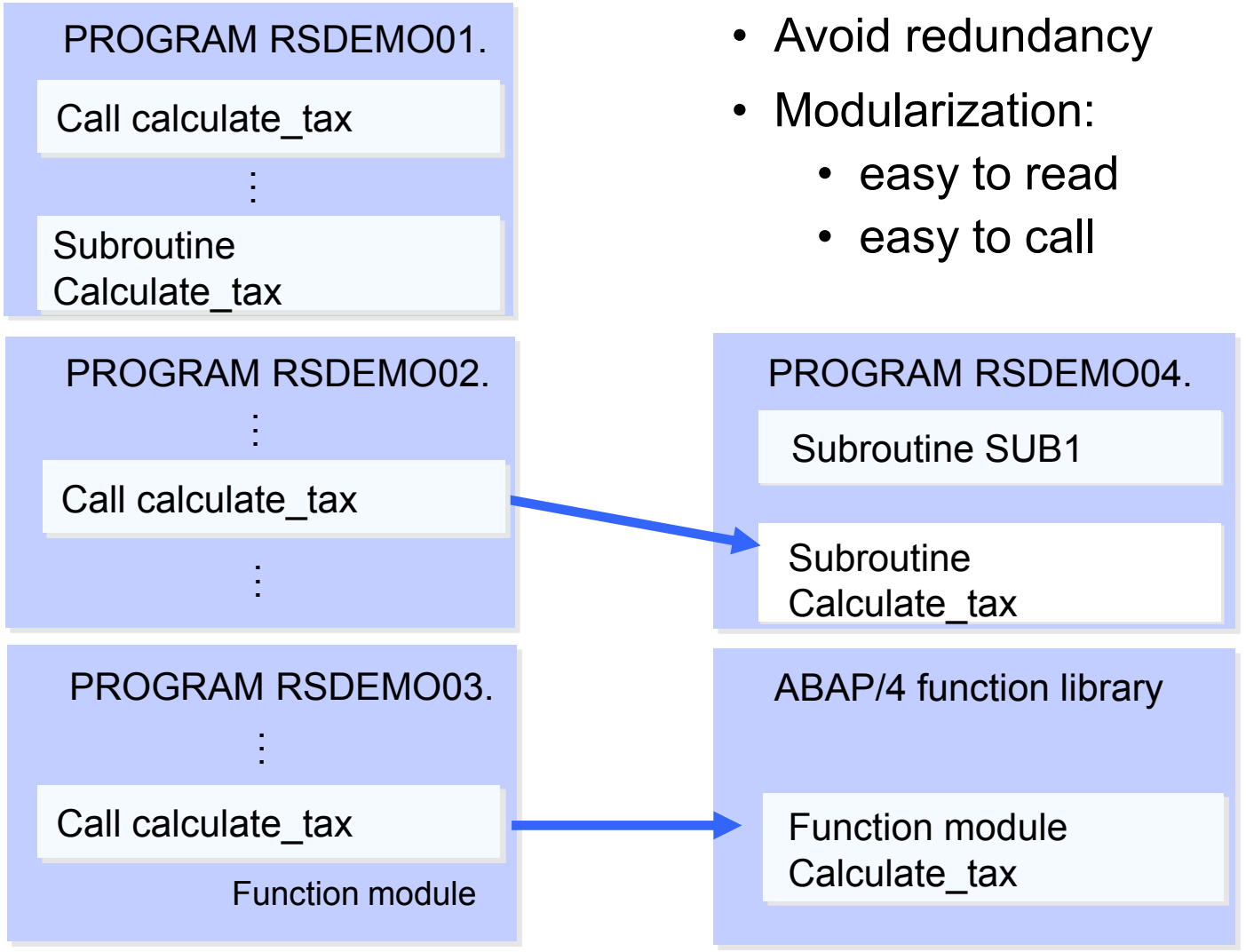
- Terminate loop pass immediately without any condition

```
DO 100 TIMES.
    statements
    IF  SY-INDEX GE 10
    AND SY-INDEX LE 20.

        CONTINUE.

    ENDIF.
    statements
ENDDO.
```

ABAP statement and Keyword(con't)



- Avoid redundancy
- Modularization:
  - easy to read
  - easy to call

- **Define**

  FORM <name> [<parameters>].
  <statement block>
  ENDFORM.

- **Calling internally**

  PERFORM <name> [<parameters>].

- **Calling external subroutine**

  PERFORM <name>(program name) <parameters>
  [IF FOUND]

ABAP statement and Keyword(con't)

```
form f_check_file_exists using filename.
   ..............
endform.
```

```
Parameters:
p_path like rlgrap-filename.
.............................
perform f_check_file_exists using p_path.
```

Content

- Introduction R/3 Basis system overview
- ABAP/4 overview

  Component, Program, ABAP dictionary, Function Module , Message/Transaction code, etc.

- Common Statement / Command

  - Main event in ABAP program

  - General ABAP command

- Transportation for ABAP objects
- Type of problem and how to investigate

  Program bug, ABAP runtime error, Update terminate, Unexpected error message.