

```

1  *&-----*
2  *& Report zdpm practice1
3  *&-----*
4  *&
5  *&-----*
6  REPORT zdpm_practice1.
7
8
9  " subroutines ( from...endform ---> perform ).
10
11  *They are used to modularize the code and used for the code reusability
12  *as it has both global and local scope.
13  *we can have nested subroutines
14
15  *we can pass the values in the subroutine by 3 ways
16  *1. pass by value.
17  *2. pass by reference
18  *3. pass by value and return.
19
20
21
22  * global declarations.
23  DATA: n1   TYPE int4 VALUE 1,
24         n2   TYPE int4 VALUE 2,
25         tot  TYPE int4.
26
27  data: lt_mara type table of mara,
28        ls_mara type mara.
29
30  START-OF-SELECTION.
31      PERFORM sum USING n1 n2. " calling the sum subroutine. --> by value
32
33      PERFORM sum1 USING n1 n2. " calling the sum1 subroutine. --> by reference.
34
35      PERFORM sum2 USING n1 n2 CHANGING tot. " calling the sum1 subroutine. --> by
reference and return.
36      WRITE /: tot.
37
38      PERFORM diff USING n1 n2 CHANGING tot. " calling the sum1 subroutine. --> by value
and return.
39      WRITE /: tot.
40
41      PERFORM exit_handler USING n1. " using exit keyword.
42
43      PERFORM continue_handler USING n1 n2 CHANGING tot. " using continue keyword.
44      IF tot > 0. WRITE /: tot. ENDIF.
45
46      PERFORM check_stmt USING n1 n2 CHANGING tot. " using check statement.
47      IF tot > 0. WRITE /: tot. ENDIF.
48
49  skip 4.
50  uline 2.
51      PERFORM data_static. "difference between data and static
52  skip 3.
53  uline 2.
54
55  perform get_Data. "local used inside but must be declared before.
56  write:/'after local calling', n1.
57
58  perform get_table_data tables lt_mara. " fetch the mara table data.
59
60  perform get_wa_data using ls_mara. " fetch the mara table workarea.
61  end-of-SELECTION.
62
63
64
65  * 1. pass by value.
66  FORM sum USING VALUE(p1)
67              VALUE(p2).
68
69      DATA : total TYPE int4. " local variable with local scope.
70      total = p1 + p2 .
71      WRITE /: total.
72  ENDFORM.

```

```

73
74
75 * 2. pass by reference
76 FORM sum1 USING p1 TYPE int4
77             p2 TYPE int4.
78
79 DATA: total TYPE int4.
80 total = p1 + p2.
81 WRITE /: total.
82 ENDFORM.
83
84
85 * 3. pass by reference and return --> changing parameter
86 FORM sum2 USING p1 TYPE int4
87             p2 TYPE int4
88             CHANGING
89             total TYPE int4.
90
91 total = p1 + p2.
92 ENDFORM.
93
94
95 * 3.1 pass by value and return --> changing parameter
96 FORM diff USING VALUE(p1)
97             VALUE(p2)
98             CHANGING
99             total TYPE int4.
100
101 total = p1 - p2.
102 ENDFORM.
103
104
105 * additional thing need to know in subroutines.
106
107 * 1. Exit
108 FORM exit_handler USING p1 TYPE int4.
109
110 IF p1 > 10.
111     p1 = 12.
112 ELSE.
113     WRITE /'Exit'.
114     EXIT.
115 ENDIF.
116
117 ENDFORM.
118
119 * 2. CONTINUE. --> allowed only within loops (DO, WHILE, LOOP, SELECT).
120 FORM continue_handler USING p1 TYPE int4
121                         p2 TYPE int4
122                         CHANGING
123                         total TYPE int4.
124 DO 1 TIMES.
125     IF p1 > 10 AND p2 < 10.
126         total = p1 + p2.
127     ELSE.
128         WRITE: /'continue'.
129         CONTINUE.
130     ENDIF.
131 ENDDO.
132 ENDFORM.
133
134
135 * 3. check statement --> verify data based on condition
136 FORM check_stmt USING VALUE(p1)
137                     VALUE(p2)
138                     CHANGING VALUE(total).
139 WRITE /:'Check'.
140 CHECK p1 GE 10.
141 CHECK p2 LE 10.
142
143 total = p1 + p2.
144
145 ENDFORM.
146

```

```

147
148 * 4. data and static data in subroutines.
149 " it can be only used in forms, functions and methods.
150
151 FORM data_static.
152
153 data p1 type c length 5 value 'ABCDE'.
154 STATICS val TYPE c LENGTH 5 VALUE '12345'. " static declaration.
155
156 WRITE : / 'original data', p1.
157 p1 = '432'.
158 WRITE : / 'Changed data', p1.
159 WRITE : / 'Changed data', p1.
160
161
162 WRITE: / 'original value', val.
163 val = 'GHI'.
164 WRITE: / 'changed value', val.
165 WRITE: / 'changed value', val.
166
167 ENDFORM.
168
169 * 5. use of global and local work-area in subroutine.
170 form get_data.
171 local n1.
172 n1 = '100'.
173 write: / 'get_Data local :' , n1.
174 ENDFORM.
175
176
177 * 6 . passing a table in subroutine
178 form get_table_data tables lt_mara.
179 select * from mara into table lt_mara.
180 endform.
181
182 * 7 . passing a workarea assiging field symbol in subroutine
183 form get_wa_data using wa type any.
184 FIELD-SYMBOLS: <fs_wa>.
185 select single * from mara into wa.
186 ASSIGN COMPONENT 'MATNR' of STRUCTURE wa to <fs_wa>.
187 write / <fs_wa>.
188 endform.

```