**DAY10- ASSIGNMENT**

BATCH NAME: SAP ABAP HANA

NAME: GIDUGU RANI

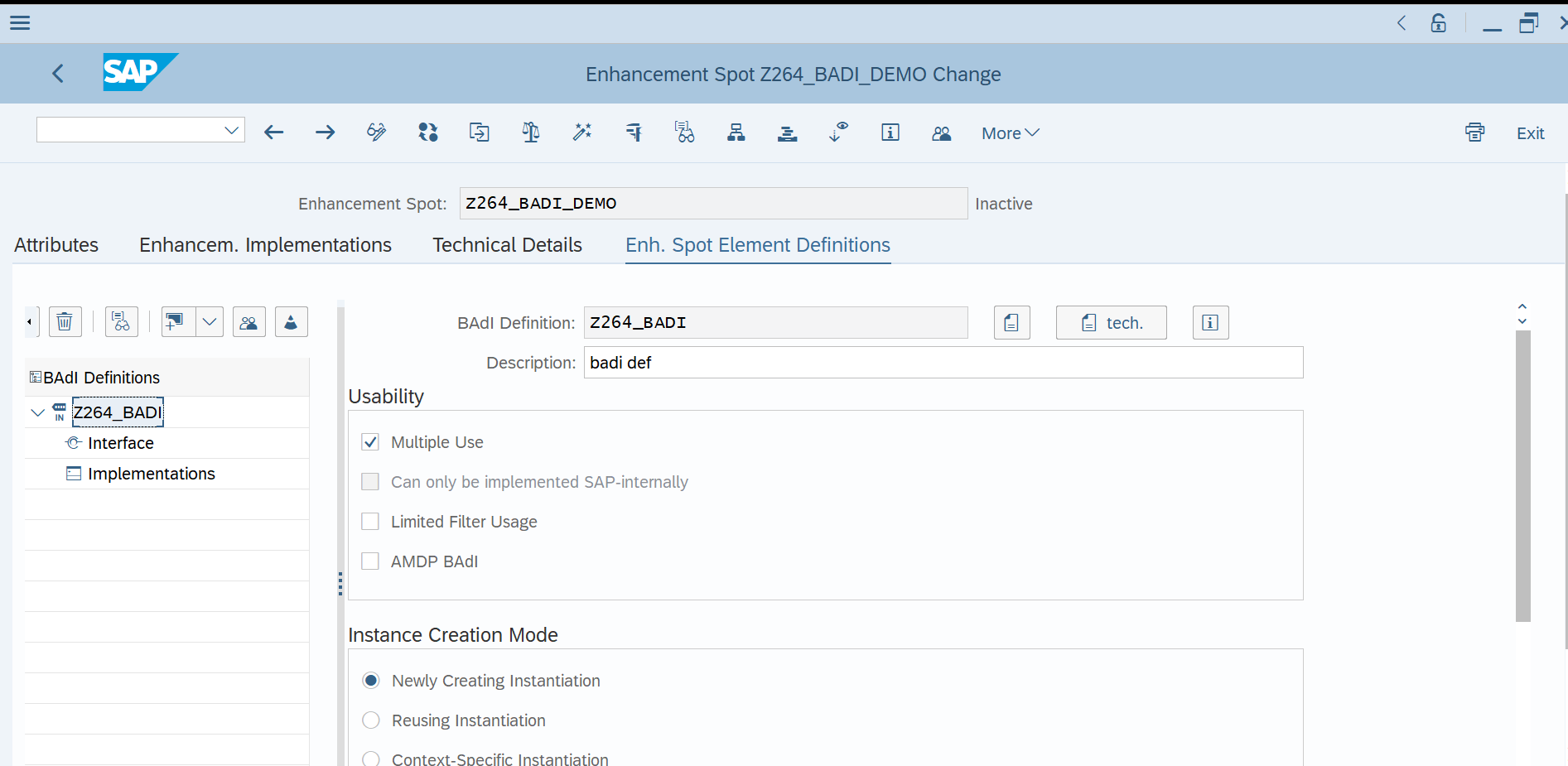
EMI ID:46255264

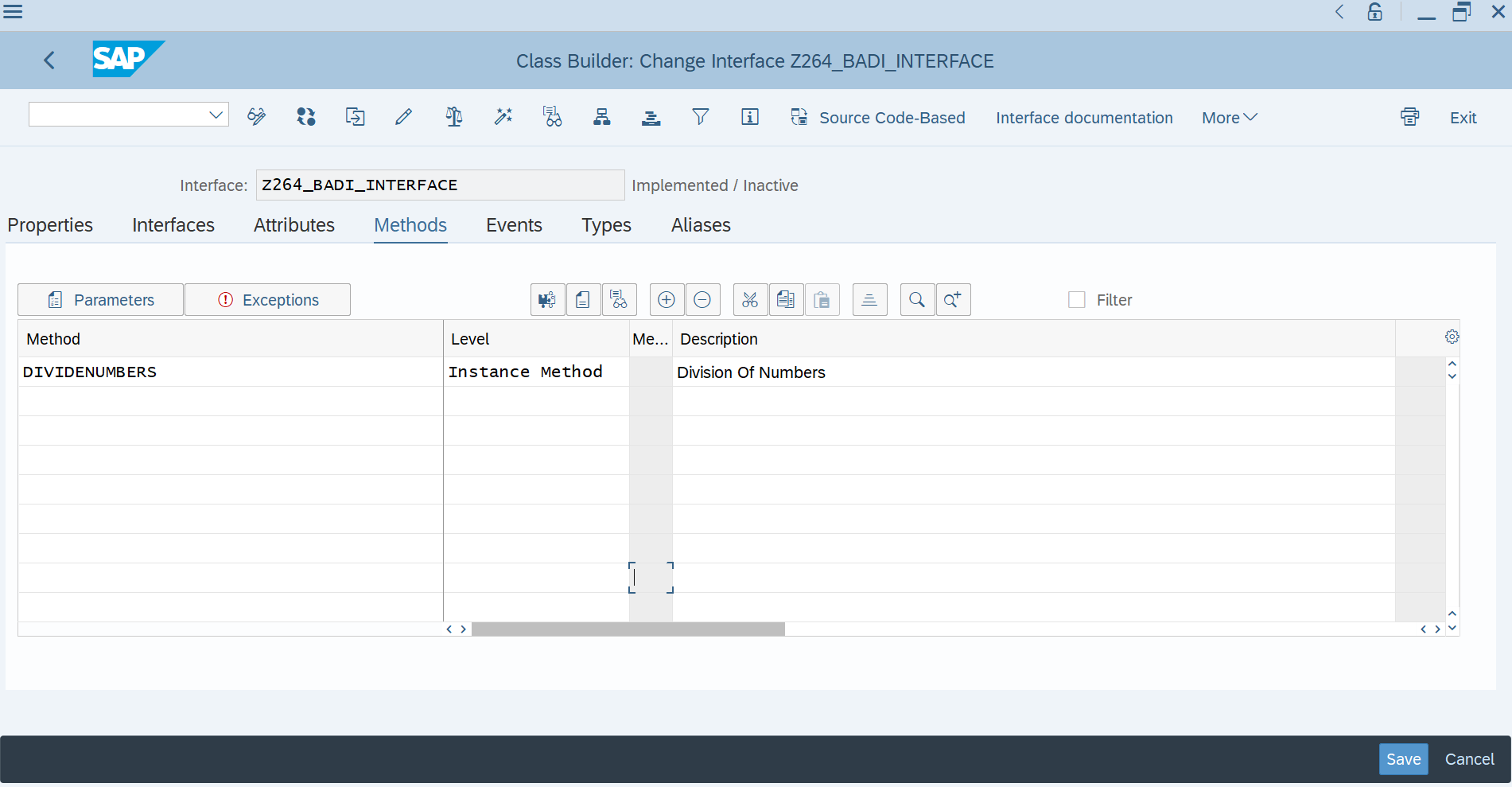
EMP EMAIL ID: GIDUGU.RANI@CAPGEMINI.COM

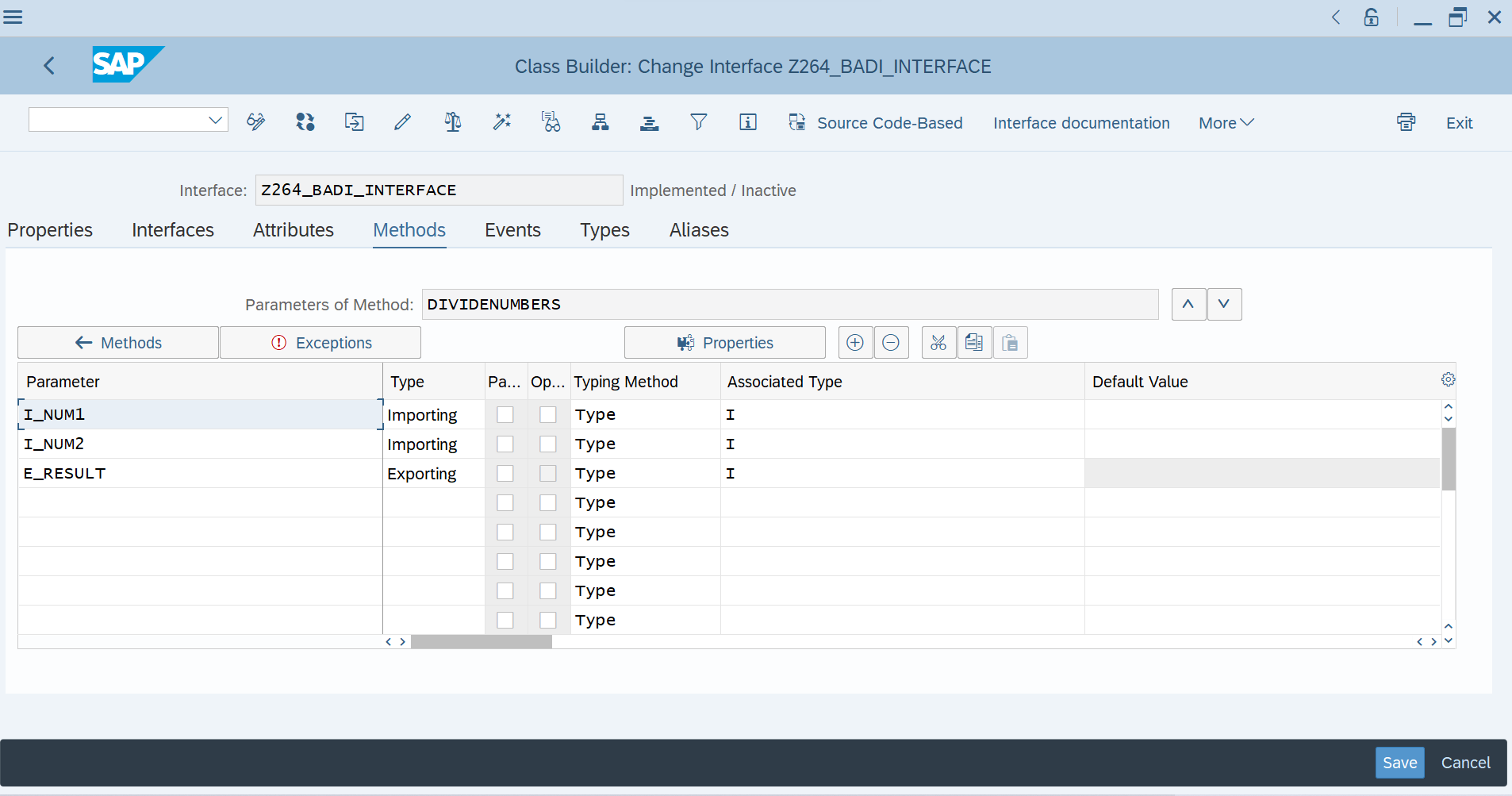
**Assignment 3 – Business Add in (BADI)-Single Use**

**Hint:**

1. **SE18 should be used to define a BADI**
2. **SE19 should be used to implement the BADI Definition.**



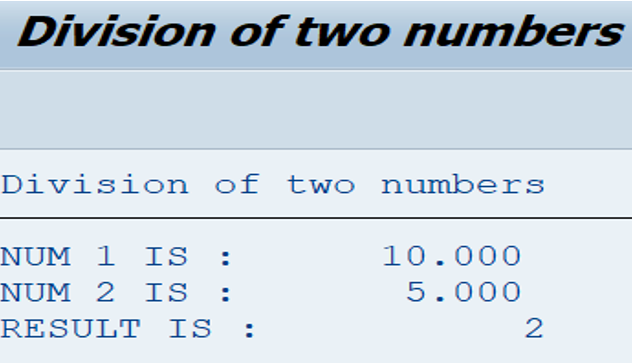




**CODE:**

\*&---------------------------------------------------------------------\*  
\*& Report Z264\_BADIREPORT  
\*&---------------------------------------------------------------------\*  
\* Description:                                                         \*  
\*                                                                      \*  
\* Author:                                                              \*  
\*                                                                      \*  
\* Create date:                                                         \*  
\*                                                                      \*  
\*&---------------------------------------------------------------------\*  
\* Modification log:                                                    \*  
\* -------------------------------------------------------------------- \*  
\* Date        User name      ID       Change Request   Case ref.       \*  
\*             Description                                              \*  
\*&---------------------------------------------------------------------\*  
REPORT Z264\_BADIREPORT.  
PARAMETERS: NUM1 TYPE I,  
            NUM2 TYPE I.  
DATA RES TYPE I.  
DATA OBJ TYPE REF TO Z264\_BADI\_INTERFACE.  
  
CALL METHOD CL\_EXITHANDLER=>get\_instance  
  EXPORTING  
    exit\_name                     = 'Z264\_BADI\_INTERFACE'  
  CHANGING  
    instance                      =   OBJ  
  .  
IF sy-subrc <> 0.  
  MESSAGE 'Error while calling method' TYPE 'I'.  
ENDIF.  
\*CALL METHOD OBJ->divide  
\*  EXPORTING  
\*    i\_num1      = NUM1  
\*    i\_num2      = NUM2  
\*  IMPORTING  
\*    e\_result    = RES  
\*  EXCEPTIONS  
\*    div\_by\_null = 1  
\*  .  
CALL METHOD OBJ->DIVIDENUMBERS  
  EXPORTING  
    I\_NUM1    = NUM1  
    I\_NUM2    = NUM2  
  IMPORTING  
    E\_RESULT  = RES  
  EXCEPTIONS  
    NULLVALUE = 1  
    OTHERS    = 2  
  .  
IF SY-SUBRC <> 0.  
 MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO  
            WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  
ENDIF.  
IF sy-subrc = 1.  
  MESSAGE 'Division by zero' TYPE 'I'.  
ELSE.  
    WRITE:/ 'NUM 1 IS : ' , NUM1.  
    WRITE:/ 'NUM 2 IS : ' , NUM2.  
    WRITE:/ 'RESULT IS : ' , RES.  
ENDIF.

**OUTPUT:**

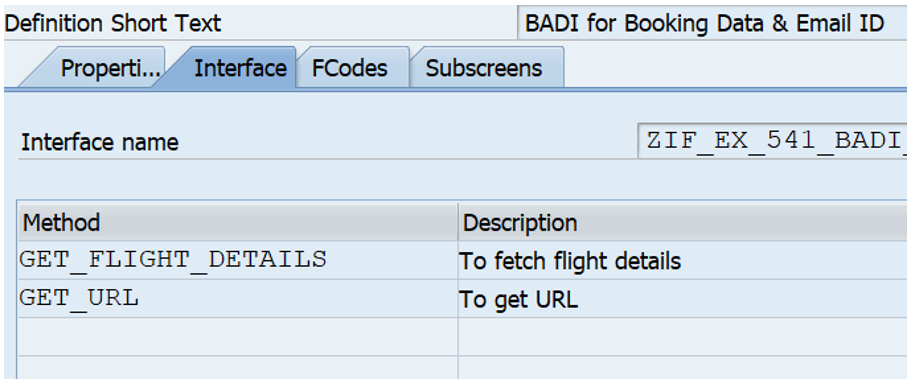


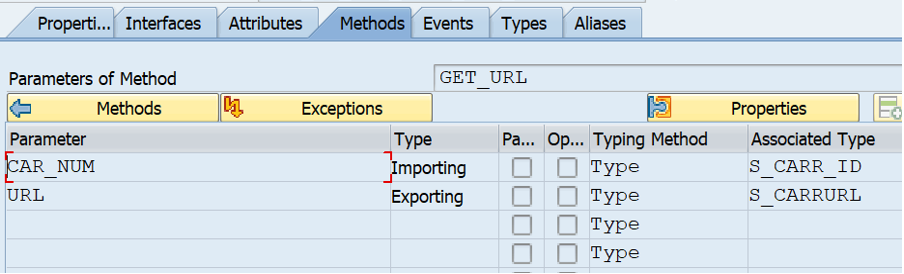
**Assignment 4 – Business Add in (BADI)-Multiple Use**

**Hint:**

**1.SE18 should be used to define a BADI**

**2.SE19 should be used to implement the BADI Definition.**





**CODE:**

*\*&---------------------------------------------------------------------\**  
*\*& Report Z264\_D10Q4*  
*\*&---------------------------------------------------------------------\**  
*\* Description:                                                         \**  
*\*                                                                      \**  
*\* Author:                                                              \**  
*\*                                                                      \**  
*\* Create date:                                                         \**  
*\*                                                                      \**  
*\*&---------------------------------------------------------------------\**  
*\* Modification log:                                                    \**  
*\* -------------------------------------------------------------------- \**  
*\* Date        User name      ID       Change Request   Case ref.       \**  
*\*             Description                                              \**  
*\*&---------------------------------------------------------------------\**  
REPORT Z264\_D10Q4.  
  
TABLES : SFLIGHT,SBOOK,SCARR.  
parameters : p\_carrid type sflight-carrid.  
DATA(TEXT) = 'DL'.  
SELECT KF~CARRID,  
       KF~CONNID,  
       KF~PRICE,  
       KF~PLANETYPE,  
       KB~BOOKID,  
       KB~SMOKER,  
       KB~INVOICE,  
       KS~CARRNAME,  
       KS~CURRCODE  
   FROM SFLIGHT AS KF  
  INNER JOIN SBOOK AS KB ON KF~CARRID = KB~CARRID  
  INNER JOIN SCARR AS KS ON KB~CARRID = KS~CARRID  
  INTO TABLE @DATA(ITAB)  
  UP TO 30 ROWS  
  WHERE KF~CARRID = @TEXT.  
CL\_DEMO\_OUTPUT=>DISPLAY\_DATA( VALUE = ITAB  
                              NAME = 'Booking Details').  
*"-------------------------------------------------------------"*  
LOOP AT ITAB INTO DATA(WA).  
  WRITE :/ WA-CARRID,(10) WA-CONNID,(15) WA-PRICE,(20) WA-PLANETYPE,(25) WA-BOOKID,(30) WA-SMOKER.  
ENDLOOP.  
*"-------------------------------------------------------------"*  
DATA(TEXT1) = 'BA'.  
SELECT SINGLE  
       KS~CARRNAME AS CA,  
       KS~CURRCODE AS CU  
   FROM SCARR AS KS  
   INTO @DATA(ITAB1)  
  WHERE KS~CARRID = @TEXT1.  
  
WRITE : / 'Single record',/ ITAB1-CA, ITAB1-CU.  
  
*"-------------------------------------------------------------"*  
  
DATA(WA\_KS) = ITAB[ 12 ].  
CL\_DEMO\_OUTPUT=>DISPLAY\_DATA( VALUE = WA\_KS  
                              NAME = 'Displaying Index Specific Data' ).  
  
*"-------------------------------------------------------------"*  
  
DATA(WA\_KS1) = ITAB[ CARRID = 'DL' CONNID = '0106' ].  
CL\_DEMO\_OUTPUT=>DISPLAY\_DATA( VALUE = WA\_KS1  
                            NAME = 'Displaying Column Specific Data' ).  
  
*"------------------------------------------------ -------------"*  
DATA(TRY) =  
COND STRING(  
WHEN ITAB1-CA = 'British Airways' THEN  
  |BA|  
  ELSE |No Airlines Found| ).  
  
WRITE :/ 'Conditional Statement Output',/ TRY.  
*"-------------------------------------------------------------"*  
TYPES : BEGIN OF TY\_SFLIGHT,  
          CARRID    TYPE SFLIGHT-CARRID,  
          CONNID    TYPE SFLIGHT-CONNID,  
          PRICE     TYPE SFLIGHT-PRICE,  
        END OF TY\_SFLIGHT,  
itab\_flight type table of TY\_SFLIGHT with key carrid.  
  
DATA(gt\_flight) = VALUE itab\_flight(  
( carrid = 'AA' connid = '0106' price = '20000')  
( carrid = 'AA' connid = '0106' price = '30000')  
( carrid = 'AZ' connid = '0206' price = '10000') ).  
DATA: gv\_tot\_age TYPE i,  
       gv\_avg\_age TYPE decfloat34.  
LOOP AT gt\_flight INTO DATA(ls\_flight)  
  GROUP BY ( carrid  = ls\_flight-carrid  
                        size  = GROUP SIZE  
                       index = GROUP INDEX )  
  ASCENDING  
  ASSIGNING FIELD-SYMBOL(<group>).  
  CLEAR: gv\_tot\_age.  
  WRITE: / |Group: { <group>-index }    Carrid: { <group>-carrid WIDTH = 15 }|  
              & |     Number in this Carrid: { <group>-size }|.  
  LOOP AT GROUP <group> ASSIGNING FIELD-SYMBOL(<ls\_member>).  
      WRITE: /13 <ls\_member>-carrid,23 <ls\_member>-connid,30 <ls\_member>-price.  
   ENDLOOP.  
  
Endloop.  
*"-------------------------------------------------------------"*  
*\*data(ks) = myclass=>get\_select( )-carrid.*  
class myclass definition.  
  public section.  
  TYPES : BEGIN OF TY\_SFLIGHT,  
          CARRID    TYPE SFLIGHT-CARRID,  
          CONNID    TYPE SFLIGHT-CONNID,  
          PRICE     TYPE SFLIGHT-PRICE,  
        END OF TY\_SFLIGHT.  
  data : it\_flight type table of ty\_sflight.  
  methods : get\_data  
          importing im\_carrid type sflight-carrid.  
endclass.  
  
class myclass implementation.  
  method get\_data.  
    select carrid connid price  
    from sflight into table it\_flight  
    where carrid = im\_carrid.  
    cl\_demo\_output=>display( it\_flight ).  
    endmethod.  
endclass.  
START-OF-SELECTION.  
data(call\_class) = new myclass( ).  
create object call\_class.  
call\_class->get\_data( EXPORTING im\_carrid = p\_carrid ).

**OUTPUT:**

