\*&---------------------------------------------------------------------\*  
\*& Report ZVAISH\_OOPS\_METHODS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZVAISH\_OOPS\_METHODS.  
  
\*                 #1 Use of STATIC Attributes  
  
\*Class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\* class-data : num type I.  
\* methods : M1.  
\*  
\*  ENDCLASS.  
\*  
\*  CLASS C1 IMPLEMENTATION.  
\*    METHOD M1.  
\*      NUM = NUM + 1.  
\*      WRITE NUM.  
\*      ENDMETHOD.  
\*    ENDCLASS.  
\*  
\*  
\* START-OF-SELECTION. "without start of selection object is not created  
\*  
\*    data: obj type REF TO C1.  
\*    CREATE OBJECT obj. "error comes in this line that statmt is not accessible  
\*    c1=>num = 5.  
\*    obj->m1( ).  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#2 different ways of calling the METHOD with one parameter.  
  
\*class c2 DEFINITION.  
\*  public SECTION.  
\*  data: v1 type I value '10'.  
\*  methods: M1 importing I\_input1 type I.  
\*  endclass.  
\*  
\*  class c2 IMPLEMENTATION.  
\*  
\*    METHOD M1.  
\*      v1 = v1 + I\_input1.  
\*      I\_input1 = 8. "cannot be modifies as it is reference type.  
\*      write v1.  
\*      ENDMETHOD.  
\*  
\*    ENDCLASS.  
\*  
\*    START-OF-SELECTION.  
\*    data: obj type ref to c2.  
\*    create object obj.  
\*\*    obj->M1( 10 ).  
\*\*    call method obj->M1 EXPORTING I\_input1 = '90'.  
\*\*    call method obj->M1( I\_input1 = 10 ).  
\*    call method obj->M1( 30 ).  
\*  
  
\*      #3 import parameters passed by ref.  
  
\*PARAMETERS p\_num type i.  
  
\*  
\*class C1 definition.  
\*  public SECTION.  
\*  methods M1 IMPORTING I\_input1 type I  VALUE(input2) type I.  
\*  
\*  
\*  
\*  ENDCLASS.  
\*  
\*  class c1 IMPLEMENTATION.  
\*    METHOD M1.  
\*    INPUT2 = 7.  
\*      WRITE : 'input 2 value is', input2.  
\*  
\*      ENDMETHOD.  
\*  
\*    ENDCLASS.  
\*    START-OF-SELECTION.  
\*    data obj type REF TO c1.  
\*    create object obj.  
\*  
\*call method obj->M1 exporting I\_input1 = 10 input2 = 60.  
  
  
\* #4 use of EXPORT AND CHANGING PARAMETERS OF A METHOD  
\*  
PARAMETERS p\_a(10) type c.  
data: lv\_tax type p decimals 2,  
      lv\_salary type p DECIMALS 2.  
  
class c1 DEFINITION.  
  public section.  
  methods tax\_calc importing grade type c  
    exporting itax type p  
      changing salary type p.  
  ENDCLASS.  
  
  class c1 IMPLEMENTATION.  
    method tax\_calc.  
      case grade.  
        when 'A01'.  
          itax = salary \* '0.2'.  
          when 'A02'.  
            itax = salary \* '0.1'.  
            when others.  
              itax = salary \* '0.15'.  
          endcase.  
      ENDMETHOD.  
    endclass.  
  
    START-OF-SELECTION.  
    data: obj type REF TO c1.  
          create object obj.  
  
          lv\_salary = 10000.  
          lv\_tax = 0.  
          WRITE:/5 'Before method call, salary and tax are' ,  
          lv\_salary ,  
          lv\_tax .  
          call method obj->tax\_Calc  
          exporting grade = p\_a  
            importing  
              itax = lv\_tax  
              changing  
                salary = lv\_salary.  
           WRITE:/5 'After method call, salary and tax are' ,  
                                      lv\_salary ,  
                                      lv\_tax.  
  
  
\*\* #5 use of returning parameters in method  
  
\*class c1 DEFINITION.  
\*  public SECTION.  
\*  methods m1 IMPORTING num1 type i  
\*    exporting num2 type i  
\*      RETURNING VALUE(result) type I  
\*      EXCEPTIONS e1.  
\*  ENDCLASS.  
\*  
\*  class c1 IMPLEMENTATION.  
\*    method m1.  
\*      if num1 LT 20.  
\*        message i007(zvaish\_messages) with 'value should be more than 20' raising e1.  
\*      else.  
\*        num2 = num1 + 10.  
\*      result = num2.  
\*      ENDIF.  
\*      ENDMETHOD.  
\*    ENDCLASS.  
\*    START-OF-SELECTION.  
\*    data w\_num type i.  
\*  
\*    data obj type REF TO c1.  
\*    create object obj.  
\*  
\*\*    obj->m1( 10 )  "is it only for just sending values.  
\*    call method : obj->m1  
\*    exporting num1 = 100  
\*    receiving result = w\_num.  
\*  
\*  
\*        write w\_num.  
  
  
  
\*  
\*\*\*\*\*\*\*\*\*\*\*#6 STATIC METHODS can on;y use static attributes  
  
\*   class c1 DEFINITION.  
\*     public section.  
\*     class-data: stnum type i value 10.  
\*     data: insnum type i value 20.  
\*  
\*     class-methods: stmeth.  
\*     methods: instmeth.  
\*  
\*     endclass.  
\*  
\*     class c1 IMPLEMENTATION.  
\*       method stmeth.  
\*         write: stnum.  
\*\*         write: insnum.  
\*         ENDMETHOD.  
\*  
\*         method instmeth.  
\*           write : stnum.  
\*           write: insnum.  
\*           ENDMETHOD.  
\*  
\*       endclass.  
\*       START-OF-SELECTION.  
\*       data obj type REF TO c1.  
\*         call method c1=>stmeth.  
\*       create object obj.  
\*  
\*       call method obj->instmeth.  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*#7 EXCEPTION RAISING\*\*\*\*\*\*\*\*\*\*  
  
\*data: w\_num type I.  
\*class c1 definition.  
\*  public section.  
\*  methods  M1 importing num1 type I  
\*       exporting num2 type I  
\*      returning VALUE(result) type I  
\*      exceptions e1.  
\*  
\*  endclass.  
\*  
\*  class c1 IMPLEMENTATION .  
\*    method m1.  
\*      if num1 LT 10.  
\*        message w007(zvaish\_messages) with 'num shoulb be more than 10' raising e1.     "in message class 007 has & that is we can dynamically use that field and insert our own messages suchnas exception .  
\*  
\*      else.  
\*        num2 = num1 \* 5.  
\*result = num2.  
\*        ENDIF.  
\*      ENDMETHOD.  
\*    ENDCLASS.  
\*  
\*    PARAMETERS: p\_no type I.  
\*    START-OF-SELECTION.  
\*  
\*    data obj type ref to c1.  
\*    create object obj.  
\*    obj->M1( p\_no ).  
\*    obj->m1( p\_no ).  
\*   call method obj->m1 exporting num1 = p\_no  
\*  
\*                     receiving result = w\_num.  
\*  
\*    if sy-subrc = 4.  
\*      write 'unable to calculate'.  
\*      else.  
\*        write w\_num.  
\*        ENDIF.