# APPENDIX A

# Composite Language Skeleton

# SECTION V: COMPOSITE LANGUAGE SKELETON

## 1. GENERAL DESCRIPTION

This section contains the composite language skeleton of Standard COBOL. It is intended to display complete and syntactically correct formats.

The leftmost margin on pages V-2 through V-4 and pages V-8 through V-19 is equivalent to margin A in a COBOL source program. The first indentation after the leftmost margin is equivalent to margin B in a COBOL source program.

On pages V-20 through V-33 the leftmost margin indicates the beginning of the format for a new COBOL verb. The first indentation after the leftmost margin indicates continuation of the format of the COBOL verb. The appearance of the italic letter *S*, *R*, *I*, or *W* to the left of the format for the verbs CLOSE, OPEN, READ, and WRITE indicates the Sequential I-O module, Relative I-O module, Indexed I-O module, or Report Writer module in which that general format is used.

The following is a summary of the formats shown on pages V-2 through V-40:

- Page V-2: General format for Identification Division
- Pages V-3 and V-4: General format for Environment Division
- Pages V-5 through V-7: General formats for file control entry
- Page V-8: General format for Data Division
- Pages V-9 through V-12: General formats for file description entry
- Pages V-13 and V-14: General formats for data description entry
- Pages V-15 and V-16: General formats for communication description entry
- Pages V-17 and V-18: General formats for report description entry and report group description entry
- Page V-19: General format for Procedure Division
- Pages V-20 through V-33: General formats for COBOL verbs
- Page V-34: General format for COPY and REPLACE statements
- Pages V-35 and V-36: General format for conditions
- Page V-37: General format for qualification
- Page V-38: Miscellaneous formats
- Page V-39: General format for nested source programs
- Page V-40: General format for a sequence of source programs

# GENERAL FORMAT FOR IDENTIFICATION DIVISION

```
IDENTIFICATION DIVISION.

                                   ┌       ┌│COMMON │┐         ┐
PROGRAM-ID.  program-name          │IS    {│        │}  PROGRAM │
                                   └       └│INITIAL│┘         ┘

[AUTHOR.   [comment-entry] ... ]

[INSTALLATION.   [comment-entry] ... ]

[DATE-WRITTEN.   [comment-entry] ... ]

[DATE-COMPILED.   [comment-entry] ... ]

[SECURITY.   [comment-entry] ... ]
```

## GENERAL FORMAT FOR ENVIRONMENT DIVISION

```
[ENVIRONMENT DIVISION.

[CONFIGURATION SECTION.

[SOURCE-COMPUTER.  [computer-name  [WITH DEBUGGING MODE].]]

[OBJECT-COMPUTER.  [computer-name

    [                         (WORDS     )]
    [MEMORY SIZE integer-1   <CHARACTERS >]
    [                         (MODULES   )]

    [PROGRAM COLLATING SEQUENCE IS alphabet-name-1]

    [SEGMENT-LIMIT IS segment-number].]]

[SPECIAL-NAMES.  [[ implementor-name-1

     (IS mnemonic-name-1 [ON STATUS IS condition-name-1 [OFF STATUS IS condition-name-2]])]
     )IS mnemonic-name-2 [OFF STATUS IS condition-name-2 [ON STATUS IS condition-name-1]](]
     )ON STATUS IS condition-name-1 [OFF STATUS IS condition-name-2]                     (] ...
     (OFF STATUS IS condition-name-2 [ON STATUS IS condition-name-1]                     )]

    [ALPHABET alphabet-name-1 IS

         (STANDARD-1                                               )]
         |STANDARD-2                                               |]
         |NATIVE                                                   |]
         |implementor-name-2                                       |] ...
        <                                                           >]
         | (             [(THROUGH)                 ]  )           |]
         | <literal-1    [<THRU   >  literal-2      ]  > ...       |]
         | (             [{ALSO literal-3} ...      ]  )           |]

    [SYMBOLIC CHARACTERS {{symbolic-character-1} ...  {IS } {integer-1} ... } ...
    [                                                 {ARE}
         [IN alphabet-name-2]} ] ...

    [CLASS class-name-1 IS {literal-4 [{THROUGH} literal-5]} ... ] ...
    [                                 [{THRU   }          ]

    [CURRENCY SIGN IS literal-6]

    [DECIMAL-POINT IS COMMA].]]]
```

## GENERAL FORMAT FOR ENVIRONMENT DIVISION

```
[INPUT-OUTPUT SECTION.

 FILE-CONTROL.

    {file-control-entry} ...

[I-O-CONTROL.
                                                  {  { [END OF] {REEL} }                   }]
                                                  {  {          {UNIT} } OF file-name-2    }]
    [[        [   {file-name-1       }]           {  { integer-1 RECORDS }                 }]
    [[RERUN   [ON {                  }]  EVERY    {                                        }] ...
    [[        [   {implementor-name-1}]           {  integer-2 CLOCK-UNITS                 }]
    [[                                            {  condition-name-1                      }]

                                                                  ]
     [      [RECORD    ]                                          ]
     [SAME  [SORT      ]  AREA FOR file-name-3  {file-name-4} ... ] ...
     [      [SORT-MERGE]                                          ]

     [MULTIPLE FILE TAPE CONTAINS {file-name-5 [POSITION integer-3]} ... ] ... .]]]]
```

# GENERAL FORMAT FOR FILE CONTROL ENTRY

SEQUENTIAL FILE:

```
SELECT  [OPTIONAL]  file-name-1

    ASSIGN TO {implementor-name-1} ...
              {literal-1         }

    [RESERVE integer-1  [AREA ]]
    [                   [AREAS]]

    [[ORGANIZATION IS]  SEQUENTIAL]

    [PADDING CHARACTER IS {data-name-1}]
    [                     {literal-2  }]

    [RECORD DELIMITER IS {STANDARD-1        }]
    [                    {implementor-name-2}]

    [ACCESS MODE IS SEQUENTIAL]

    [FILE STATUS IS data-name-2].
```

RELATIVE FILE:

```
SELECT  [OPTIONAL]  file-name-1

    ASSIGN TO {implementor-name-1} ...
              {literal-1         }

    [RESERVE integer-1  [AREA ]]
    [                   [AREAS]]

    [ORGANIZATION IS]  RELATIVE

    [                {SEQUENTIAL   [RELATIVE KEY IS data-name-1]}]
    [ACCESS MODE IS  {                                          }]
    [                {{RANDOM }                                 }]
    [                {{DYNAMIC}     RELATIVE KEY IS data-name-1 }]
    [FILE STATUS IS data-name-2].
```

# GENERAL FORMAT FOR FILE CONTROL ENTRY

## INDEXED FILE:

```
SELECT  [OPTIONAL]  file-name-1

   ASSIGN TO  {implementor-name-1}  ...
              {literal-1         }

   [RESERVE integer-1  [AREA ]]
                       [AREAS]

   [ORGANIZATION IS]  INDEXED

   [                  {SEQUENTIAL}]
   [ACCESS MODE IS    {RANDOM    }]
   [                  {DYNAMIC   }]

   RECORD KEY IS data-name-1

   [ALTERNATE RECORD KEY IS data-name-2  [WITH DUPLICATES]] ...

   [FILE STATUS IS data-name-3].
```

## SORT OR MERGE FILE:

```
SELECT file-name-1  ASSIGN TO  {implementor-name-1}  ...  .
                               {literal-1         }
```

# GENERAL FORMAT FOR FILE CONTROL ENTRY

REPORT FILE:

```
SELECT  [OPTIONAL]  file-name-1

    ASSIGN TO  {implementor-name-1}  ...
               {literal-1         }

    [RESERVE integer-1  [AREA ]]
                        [AREAS]

    [[ORGANIZATION IS]  SEQUENTIAL]]

    [PADDING CHARACTER IS  {data-name-1}]
                           {literal-2  }

    [RECORD DELIMITER IS  {STANDARD-1         }]
                          {implementor-name-2}

    [ACCESS MODE IS SEQUENTIAL]

    [FILE STATUS IS data-name-2].
```

## GENERAL FORMAT FOR DATA DIVISION

```
[DATA DIVISION.

[FILE SECTION.

⎡file-description-entry {record-description-entry} ...            ⎤     ⎤
⎢sort-merge-file-description-entry {record-description-entry} ... ⎥ ... ⎥
⎣report-file-description-entry                                    ⎦     ⎦

[WORKING-STORAGE SECTION.

⎡77-level-description-entry⎤     ⎤
⎣record-description-entry  ⎦ ... ⎦

[LINKAGE SECTION.

⎡77-level-description-entry⎤     ⎤
⎣record-description-entry  ⎦ ... ⎦

[COMMUNICATION SECTION.

[communication-description-entry [record-description-entry] ... ] ... ]

[REPORT SECTION.

[report-description-entry {report-group-description-entry} ... ] ... ]]
```

# GENERAL FORMAT FOR FILE DESCRIPTION ENTRY

SEQUENTIAL FILE:

```
FD  file-name-1

    [IS EXTERNAL]

    [IS GLOBAL]

    [BLOCK CONTAINS  [integer-1 TO]  integer-2 {RECORDS   }]
    [                                          {CHARACTERS}]

    [          {CONTAINS integer-3 CHARACTERS                                          }]
    [RECORD    {IS VARYING IN SIZE [[FROM integer-4] [TO integer-5] CHARACTERS]        }]
    [          {    [DEPENDING ON data-name-1]                                         }]
    [          {CONTAINS integer-6 TO integer-7 CHARACTERS                             }]

    [LABEL  {RECORD IS  }  {STANDARD}]
    [       {RECORDS ARE}  {OMITTED }]

    [VALUE OF  {implementor-name-1 IS  {data-name-2}} ...]
    [          {                       {literal-1  }}    ]

    [DATA  {RECORD IS  }  {data-name-3} ...]
    [      {RECORDS ARE}                   ]

    [LINAGE IS  {data-name-4}  LINES  [WITH FOOTING AT  {data-name-5}]
    [           {integer-8  }         [                 {integer-9  }]

          [LINES AT TOP  {data-name-6 }] [LINES AT BOTTOM  {data-name-7 }]]
          [              {integer-10  }] [                 {integer-11  }]]

    [CODE-SET IS alphabet-name-1].
```

# GENERAL FORMAT FOR FILE DESCRIPTION ENTRY

RELATIVE FILE:

```
FD  file-name-1

    [IS EXTERNAL]

    [IS GLOBAL]

    [BLOCK CONTAINS  [integer-1 TO]  integer-2 {RECORDS   }]
                                               {CHARACTERS}]

    [        {CONTAINS integer-3 CHARACTERS                                          }]
    [        {IS VARYING IN SIZE [[FROM integer-4] [TO integer-5] CHARACTERS]        }]
    [RECORD  {     [DEPENDING ON data-name-1]                                        }]
    [        {CONTAINS integer-6 TO integer-7 CHARACTERS                             }]

    [LABEL  {RECORD IS  } {STANDARD}]
    [       {RECORDS ARE} {OMITTED }]

    [VALUE OF {implementor-name-1 IS {data-name-2}} ...]
    [                                {literal-1  }     ]

    [DATA {RECORD IS  }  {data-name-3} ...] .
    [     {RECORDS ARE}                   ]
```

## GENERAL FORMAT FOR FILE DESCRIPTION ENTRY

INDEXED FILE:

```
FD  file-name-1

    [IS EXTERNAL]

    [IS GLOBAL]

    [BLOCK CONTAINS  [integer-1 TO]  integer-2 {RECORDS   }]
                                               {CHARACTERS}

    [        {CONTAINS integer-3 CHARACTERS                                                         }]
    [RECORD  {IS VARYING IN SIZE [[FROM integer-4] [TO integer-5] CHARACTERS]                      }]
    [        {    [DEPENDING ON data-name-1]                                                        }]
    [        {CONTAINS integer-6 TO integer-7 CHARACTERS                                            }]

    [LABEL {RECORD IS  } {STANDARD}]
    [      {RECORDS ARE} {OMITTED }]

    [VALUE OF {implementor-name-1 IS {data-name-2}} ...]
    [                                {literal-1  }     ]

    [DATA {RECORD IS  }  {data-name-3} ...] .
    [     {RECORDS ARE}                   ]
```

# GENERAL FORMAT FOR FILE DESCRIPTION ENTRY

SORT-MERGE FILE:

```
SD  file-name-1
                                                                                       ⎫⎤
    ⎡         ⎧ CONTAINS integer-1 CHARACTERS                                          ⎪⎥
    ⎢ RECORD  ⎨ IS VARYING IN SIZE [[FROM integer-2] [TO integer-3] CHARACTERS]        ⎬⎥
    ⎢         ⎪      [DEPENDING ON data-name-1]                                        ⎪⎥
    ⎣         ⎩ CONTAINS integer-4 TO integer-5 CHARACTERS                             ⎭⎦

    [DATA {RECORD IS  } {data-name-2} ... ] .
          {RECORDS ARE}
```

REPORT FILE:

```
FD  file-name-1

     [IS EXTERNAL]

     [IS GLOBAL]

     [BLOCK CONTAINS  [integer-1 TO]  integer-2 {RECORDS   }]
                                                {CHARACTERS}

     [RECORD    {CONTAINS integer-3 CHARACTERS             }]
                {CONTAINS integer-4 TO integer-5 CHARACTERS}

     [LABEL  {RECORD IS  } {STANDARD}]
             {RECORDS ARE} {OMITTED }

     [VALUE OF {implementor-name-1 IS {data-name-1}} ...]
                                      {literal-1  }

     [CODE-SET IS alphabet-name-1]

     {REPORT IS  } {report-name-1} ...    .
     {REPORTS ARE}
```

## GENERAL FORMAT FOR DATA DESCRIPTION ENTRY

FORMAT 1:

```
level-number  [data-name-1]
              [FILLER     ]

     [REDEFINES data-name-2]

     [IS EXTERNAL]

     [IS GLOBAL]

     [{PICTURE}  IS character-string]
     [{PIC    }                     ]

     [                {BINARY        }]
     [                {COMPUTATIONAL }]
     [[USAGE IS]      {COMP          }]
     [                {DISPLAY       }]
     [                {INDEX         }]
     [                {PACKED-DECIMAL}]

     [[SIGN IS]  {LEADING }  [SEPARATE CHARACTER]]
     [           {TRAILING}                      ]

     [OCCURS integer-2 TIMES                                             ]
     [                                                                   ]
     [     [{ASCENDING }  KEY IS  {data-name-3} ... ] ...                ]
     [     [{DESCENDING}                            ]                    ]
     [                                                                   ]
     [      [INDEXED BY  {index-name-1} ... ]                            ]
     [                                                                   ]
     [OCCURS integer-1 TO integer-2 TIMES DEPENDING ON data-name-4       ]
     [                                                                   ]
     [     [{ASCENDING }  KEY IS  {data-name-3} ... ] ...                ]
     [     [{DESCENDING}                            ]                    ]
     [                                                                   ]
     [      [INDEXED BY  {index-name-1} ... ]                            ]

     [{SYNCHRONIZED}  [LEFT ]]
     [{SYNC        }  [RIGHT]]

     [{JUSTIFIED}  RIGHT]
     [{JUST     }       ]


     [VALUE IS literal-1].
```

## GENERAL FORMAT FOR DATA DESCRIPTION ENTRY

FORMAT 2:

$$\text{66 data-name-1 } \underline{\text{RENAMES}} \text{ data-name-2 } \left[ \left\{ \begin{matrix} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{matrix} \right\} \text{ data-name-3} \right]$$

FORMAT 3:

$$\text{88 condition-name-1 } \left\{ \begin{matrix} \underline{\text{VALUE}} \text{ IS} \\ \underline{\text{VALUES}} \text{ ARE} \end{matrix} \right\} \left\{ \text{literal-1 } \left[ \left\{ \begin{matrix} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{matrix} \right\} \text{ literal-2} \right] \right\}$$

## GENERAL FORMAT FOR COMMUNICATION DESCRIPTION ENTRY

FORMAT 1:

```
CD  cd-name-1

                            [[SYMBOLIC QUEUE IS data-name-1]
                                 [SYMBOLIC SUB-QUEUE-1 IS data-name-2]
                                 [SYMBOLIC SUB-QUEUE-2 IS data-name-3]
                                 [SYMBOLIC SUB-QUEUE-3 IS data-name-4]
                                 [MESSAGE DATE IS data-name-5]
                                 [MESSAGE TIME IS data-name-6]
                                 [SYMBOLIC SOURCE IS data-name-7]
    FOR  [INITIAL]  INPUT        [TEXT LENGTH IS data-name-8]
                                 [END KEY IS data-name-9]
                                 [STATUS KEY IS data-name-10]
                                 [MESSAGE COUNT IS data-name-11]]
                            [data-name-1, data-name-2, data-name-3,
                                 data-name-4, data-name-5, data-name-6,
                                 data-name-7, data-name-8, data-name-9,
                                 data-name-10, data-name-11]
```

## GENERAL FORMAT FOR COMMUNICATION DESCRIPTION ENTRY

FORMAT 2:

```
CD  cd-name-1 FOR OUTPUT

    [DESTINATION COUNT IS data-name-1]

    [TEXT LENGTH IS data-name-2]

    [STATUS KEY IS data-name-3]

    [DESTINATION TABLE OCCURS integer-1 TIMES

        [INDEXED BY  {index-name-1} ... ]]

    [ERROR KEY IS data-name-4]

    [SYMBOLIC DESTINATION IS data-name-5].
```

FORMAT 3:

```
CD  cd-name-1
                         ┌[[MESSAGE DATE IS data-name-1]                       ┐
                         │                                                     │
                         │     [MESSAGE TIME IS data-name-2]                   │
                         │                                                     │
                         │     [SYMBOLIC TERMINAL IS data-name-3]              │
                         │                                                     │
                         │     [TEXT LENGTH IS data-name-4]                    │
    FOR  [INITIAL]  I-O  │                                                     │
                         │     [END KEY IS data-name-5]                        │
                         │                                                     │
                         │     [STATUS KEY IS data-name-6]]                    │
                         │                                                     │
                         │ [data-name-1, data-name-2, data-name-3,             │
                         │                                                     │
                         └     data-name-4, data-name-5, data-name-6]          ┘
```

## GENERAL FORMAT FOR REPORT DESCRIPTION ENTRY

```
RD  report-name-1

    [IS GLOBAL]

    [CODE literal-1]

    [{CONTROL IS  } {{data-name-1} ...             }]
    [{CONTROLS ARE} {FINAL [data-name-1] ...       }]

    [      [LIMIT IS  ]            [LINE ]
    [ PAGE [LIMITS ARE] integer-1  [LINES]  [HEADING integer-2]

         [FIRST DETAIL integer-3]  [LAST DETAIL integer-4]

         [FOOTING integer-5] ].
```

## GENERAL FORMAT FOR REPORT GROUP DESCRIPTION ENTRY

FORMAT 1:

```
01  [data-name-1]

    [LINE NUMBER IS {integer-1  [ON NEXT PAGE]}]
    [               {PLUS integer-2           }]

    [               {integer-3     }]
    [NEXT GROUP IS  {PLUS integer-4}]
    [               {NEXT PAGE     }]

              { {REPORT HEADING}                                }
              { {RH            }                                }
              {                                                 }
              { {PAGE HEADING}                                  }
              { {PH          }                                  }
              {                                                 }
              { {CONTROL HEADING}  {data-name-2}                }
              { {CH             }  {FINAL      }                }
              {                                                 }
    TYPE IS   { {DETAIL}                                        }
              { {DE    }                                        }
              {                                                 }
              { {CONTROL FOOTING}  {data-name-3}                }
              { {CF             }  {FINAL      }                }
              {                                                 }
              { {PAGE FOOTING}                                  }
              { {PF          }                                  }
              {                                                 }
              { {REPORT FOOTING}                                }
              { {RF            }                                }

   [[USAGE IS]  DISPLAY].
```

# GENERAL FORMAT FOR REPORT GROUP DESCRIPTION ENTRY

FORMAT 2:

```
level-number   [data-name-1]

    [                 { integer-1   [ON NEXT PAGE] } ]
    [LINE NUMBER IS   { PLUS Integer-2             } ]

    [[USAGE IS]  DISPLAY].
```

FORMAT 3:

```
level-number   [data-name-1]

    { PICTURE }
    { PIC     }  IS character-string

    [[USAGE IS]  DISPLAY]

    [              { LEADING  }                        ]
    [[SIGN IS]     { TRAILING }  SEPARATE CHARACTER    ]

    [ { JUSTIFIED }          ]
    [ { JUST      }   RIGHT  ]

    [BLANK WHEN ZERO]

    [                  { integer-1   [ON NEXT PAGE] } ]
    [LINE NUMBER IS    { PLUS integer-2             } ]

    [COLUMN NUMBER IS integer-3]

    { SOURCE IS identifier-1                                                 }
    {                                                                        }
    { VALUE IS literal-1                                                     }
    {                                                                        }
    { {SUM  {identifier-2} ...   [UPON {data-name-2} ... ]} ...              }
    {                                                                        }
    {          [            { data-name-3 } ]                                }
    {          [RESET ON    { FINAL       } ]                                }

    [GROUP INDICATE].
```

# GENERAL FORMAT FOR PROCEDURE DIVISION

FORMAT 1:

```
[PROCEDURE DIVISION  [USING  {data-name-1} ... ].

[DECLARATIVES.

{section-name SECTION [segment-number].

     USE statement.

[paragraph-name.

     [sentence] ... ] ... } ...

 END DECLARATIVES.]

{section-name SECTION [segment-number].

[paragraph-name.

     [sentence] ... ] ... } ... ]
```

FORMAT 2:

```
[PROCEDURE DIVISION  [USING  {data-name-1} ... ].

{paragraph-name.

     [sentence] ... } ... ]
```

# GENERAL FORMAT FOR COBOL VERBS

```
ACCEPT identifier-1  [FROM mnemonic-name-1]


                          ( DATE        )
                          ( DAY         )
ACCEPT identifier-2 FROM  { DAY-OF-WEEK }
                          ( TIME        )


ACCEPT cd-name-1 MESSAGE COUNT


ADD  {identifier-1}  ...   TO  {identifier-2 [ROUNDED]} ...
     {literal-1   }

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-ADD]


ADD  {identifier-1}  ...   TO  {identifier-2}
     {literal-1   }            {literal-2   }

   GIVING  {identifier-3 [ROUNDED]} ...

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-ADD]


ADD  {CORRESPONDING}  identifier-1 TO identifier-2 [ROUNDED]
     {CORR         }

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-ADD]


ALTER  {procedure-name-1 TO  [PROCEED TO]  procedure-name-2} ...


CALL  {identifier-1}  [USING {[BY REFERENCE]  {identifier-2} ... } ... ]
      {literal-1   }  [      {BY CONTENT  {identifier-2} ...      }     ]

   [ON OVERFLOW imperative-statement-1]

   [END-CALL]
```

# GENERAL FORMAT FOR COBOL VERBS

```
   CALL  {identifier-1}  [USING {[BY REFERENCE]  {identifier-2} ...}  ...]
         {literal-1   }         {BY CONTENT    {identifier-2} ...   }

      [ON EXCEPTION imperative-statement-1]

      [NOT ON EXCEPTION imperative-statement-2]

      [END-CALL]


   CANCEL {identifier-1}  ...
          {literal-1   }

                          [{REEL} [FOR REMOVAL]]
SW CLOSE  {file-name-1    [{UNIT}              ]   } ...
                          [                    ]
                          [WITH {NO REWIND}    ]
                          [     {LOCK     }    ]

RI CLOSE  {file-name-1  [WITH LOCK]} ...

   COMPUTE  {identifier-1 [ROUNDED]} ...  =  arithmetic-expression-1

      [ON SIZE ERROR imperative-statement-1]

      [NOT ON SIZE ERROR imperative-statement-2]

      [END-COMPUTE]


   CONTINUE


   DELETE file-name-1 RECORD

      [INVALID KEY imperative-statement-1]

      [NOT INVALID KEY imperative-statement-2]

      [END-DELETE]


            {INPUT [TERMINAL]}
   DISABLE  {I-O TERMINAL    }  cd-name-1  [WITH KEY  {identifier-1}]
            {OUTPUT          }                        {literal-1   }
```

# GENERAL FORMAT FOR COBOL VERBS

```
DISPLAY  {identifier-1}  ...   [UPON mnemonic-name-1]   [WITH NO ADVANCING]
         {literal-1   }

DIVIDE  {identifier-1}  INTO  {identifier-2 [ROUNDED]} ...
        {literal-1   }

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-DIVIDE]

DIVIDE  {identifier-1}  INTO  {identifier-2}
        {literal-1   }        {literal-2   }

   GIVING  {identifier-3 [ROUNDED]} ...

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-DIVIDE]

DIVIDE  {identifier-1}  BY  {identifier-2}
        {literal-1   }      {literal-2   }

   GIVING  {identifier-3 [ROUNDED]} ...

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-DIVIDE]

DIVIDE  {identifier-1}  INTO  {identifier-2}  GIVING identifier-3 [ROUNDED]
        {literal-1   }        {literal-2   }

   REMAINDER identifier-4

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-DIVIDE]
```

## GENERAL FORMAT FOR COBOL VERBS

```
DIVIDE {identifier-1}  BY  {identifier-2}  GIVING identifier-3 [ROUNDED]
       {literal-1   }      {literal-2   }

    REMAINDER identifier-4

    [ON SIZE ERROR imperative-statement-1]

    [NOT ON SIZE ERROR imperative-statement-2]

    [END-DIVIDE]


        {INPUT [TERMINAL]}               [           {identifier-1}]
ENABLE  {I-O TERMINAL    }  cd-name-1    [WITH KEY   {literal-1   }]
        {OUTPUT          }


ENTER language-name-1  [routine-name-1].


          {identifier-1}  [       {identifier-2}]
          {literal-1   }  [       {literal-2   }]
EVALUATE  {expression-1}  [ALSO   {expression-2}]  ...
          {TRUE        }  [       {TRUE        }]
          {FALSE       }  [       {FALSE       }]

  {{WHEN

    {ANY                                                                                         }
    {condition-1                                                                                 }
    {TRUE                                                                                        }
    {FALSE                                                                                       }
    {       {{identifier-3             }  [{THROUGH}  {identifier-4             }]}               }
    {[NOT]  {{literal-3                }  [{THRU   }  {literal-4                }]}               }
    {       {{arithmetic-expression-1  }  [           {arithmetic-expression-2  }]}               }

   [ALSO

     {ANY                                                                                          }]      }
     {condition-2                                                                                  }]      }
     {TRUE                                                                                         }]      }
     {FALSE                                                                                        }] ...  } ...
     {       {{identifier-5             }  [{THROUGH}  {identifier-6             }]}                }]      }
     {[NOT]  {{literal-5                }  [{THRU   }  {literal-6                }]}                }]      }
     {       {{arithmetic-expression-3  }  [           {arithmetic-expression-4  }]}                }]      }

   imperative-statement-1} ...

  [WHEN OTHER imperative-statement-2]

  [END-EVALUATE]
```

## GENERAL FORMAT FOR COBOL VERBS

```
EXIT
```

```
EXIT PROGRAM
```

```
GENERATE  ⎧data-name-1  ⎫
          ⎩report-name-1⎭
```

```
GO TO  [procedure-name-1]
```

```
GO TO  {procedure-name-1} ...    DEPENDING ON identifier-1
```

```
                                                  ⎧ELSE {statement-2} ... [END-IF]⎫
IF condition-1 THEN  ⎧{statement-1} ...⎫          ⎨ELSE NEXT SENTENCE             ⎬
                     ⎩NEXT SENTENCE    ⎭          ⎩END-IF                         ⎭
```

```
INITIALIZE  {identifier-1} ...

    ⎡              ⎧⎧ALPHABETIC           ⎫                              ⎫     ⎤
    ⎢              ⎪⎪ALPHANUMERIC         ⎪                              ⎪     ⎥
    ⎢ REPLACING    ⎨⎨NUMERIC              ⎬  DATA BY  ⎧identifier-2⎫     ⎬ ... ⎥
    ⎢              ⎪⎪ALPHANUMERIC-EDITED  ⎪           ⎩literal-1   ⎭     ⎪     ⎥
    ⎣              ⎩⎩NUMERIC-EDITED       ⎭                              ⎭     ⎦
```

```
INITIATE  {report-name-1} ...
```

```
INSPECT identifier-1 TALLYING

 ⎧                  ⎧ CHARACTERS [⎧BEFORE⎫ INITIAL ⎧identifier-4⎫] ...                                ⎫ ⎫
 ⎨identifier-2 FOR  ⎨                    ⎩AFTER ⎭         ⎩literal-2   ⎭                                      ⎬ ...⎬ ...
 ⎩                  ⎩ ⎧⎧ALL    ⎫ ⎧identifier-3⎫ [⎧BEFORE⎫ INITIAL ⎧identifier-4⎫] ...⎫ ...        ⎭ ⎭
                      ⎩⎩LEADING⎭ ⎩literal-1   ⎭  ⎩AFTER ⎭         ⎩literal-2   ⎭     ⎭
```

```
INSPECT identifier-1 REPLACING

 ⎧ CHARACTERS BY ⎧identifier-5⎫ [⎧BEFORE⎫ INITIAL ⎧identifier-4⎫] ...                                         ⎫
 ⎪               ⎩literal-3   ⎭  ⎩AFTER ⎭         ⎩literal-2   ⎭                                              ⎪
 ⎨ ⎧ALL    ⎫ ⎧identifier-3⎫ BY ⎧identifier-5⎫ [⎧BEFORE⎫ INITIAL ⎧identifier-4⎫] ...⎫ ...                      ⎬ ...
 ⎪ ⎨LEADING⎬ ⎩literal-1   ⎭    ⎩literal-3   ⎭  ⎩AFTER ⎭         ⎩literal-2   ⎭     ⎭                          ⎪
 ⎩ ⎩FIRST  ⎭                                                                                                ⎭
```

## GENERAL FORMAT FOR COBOL VERBS

```
INSPECT identifier-1 TALLYING
 (                   (CHARACTERS [{BEFORE} INITIAL {identifier-4}] ...                             )    (
 {identifier-2 FOR  {            [{AFTER }         {literal-2   }]                                 } ...{ ...
 (                   ({ALL    } {{identifier-3} [{BEFORE} INITIAL {identifier-4}] ...} ...          )    (
                     ({LEADING} {{literal-1   } [{AFTER }         {literal-2   }]     }

REPLACING
 (CHARACTERS BY {identifier-5} [{BEFORE} INITIAL {identifier-4}] ...                        )
 (              {literal-3   } [{AFTER }         {literal-2   }]                            }...
 ({ALL    } {{identifier-3} BY {identifier-5} [{BEFORE} INITIAL {identifier-4}] ... } ...    )
 ({LEADING} {{literal-1   }    {literal-3   } [{AFTER }         {literal-2   }]     }
 ({FIRST  }


INSPECT identifier-1 CONVERTING {identifier-6} TO {identifier-7}
                                {literal-4   }    {literal-5   }

    [{BEFORE} INITIAL {identifier-4}] ...
    [{AFTER }         {literal-2   }]


MERGE file-name-1 {ON {ASCENDING } KEY {data-name-1} ... } ...
                  {   {DESCENDING}                       }

    [COLLATING SEQUENCE IS alphabet-name-1]

    USING file-name-2 {file-name-3} ...

    {OUTPUT PROCEDURE IS procedure-name-1 [{THROUGH} procedure-name-2]}
    {                                     [{THRU   }                 ]}
    {GIVING {file-name-4} ...                                         }


MOVE {identifier-1} TO {identifier-2} ...
     {literal-1   }


MOVE {CORRESPONDING} identifier-1 TO identifier-2
     {CORR         }


MULTIPLY {identifier-1} BY {identifier-2 [ROUNDED]} ...
         {literal-1   }

    [ON SIZE ERROR imperative-statement-1]

    [NOT ON SIZE ERROR imperative-statement-2]

    [END-MULTIPLY]
```

```
MULTIPLY {identifier-1}  BY  {identifier-2}
         {literal-1   }      {literal-2   }

   GIVING {identifier-3 [ROUNDED]} ...

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-MULTIPLY]

        {INPUT   {file-name-1  [REVERSED          ]} ...  }
        {                      [WITH NO REWIND    ]        }
S OPEN  {OUTPUT {file-name-2  [WITH NO REWIND]} ...       }  ...
        {I-O {file-name-3} ...                             }
        {EXTEND {file-name-4} ...                          }

        {INPUT {file-name-1} ...   }
        {OUTPUT {file-name-2} ...  }
RI OPEN {I-O {file-name-3} ...     }  ...
        {EXTEND {file-name-4} ...  }

W OPEN  {OUTPUT {file-name-1 [WITH NO REWIND]} ... }  ...
        {EXTEND {file-name-2} ...                  }

PERFORM  [procedure-name-1  [{THROUGH}  procedure-name-2]]
                             {THRU   }

   [imperative-statement-1 END-PERFORM]

PERFORM  [procedure-name-1  [{THROUGH}  procedure-name-2]]
                             {THRU   }

   {identifier-1}
   {integer-1   }

   TIMES  [imperative-statement-1 END-PERFORM]

PERFORM  [procedure-name-1  [{THROUGH}  procedure-name-2]]
                             {THRU   }

   [WITH TEST  {BEFORE}]  UNTIL condition-1
               {AFTER }

   [imperative-statement-1 END-PERFORM]
```

## GENERAL FORMAT FOR COBOL VERBS

```
PERFORM  [procedure-name-1  [{THROUGH}  procedure-name-2]]
                             {THRU   }

    [WITH TEST {BEFORE}]
               {AFTER }

    VARYING {identifier-2}  FROM {identifier-3}
            {index-name-1}       {index-name-2}
                                 {literal-1   }

            BY {identifier-4}  UNTIL condition-1
               {literal-2   }

      [AFTER {identifier-5}  FROM {identifier-6}
             {literal-3   }       {index-name-4}
                                  {literal-3   }

            BY {identifier-7}  UNTIL condition-2 ] ...
               {literal-4   }

    [imperative-statement-1 END-PERFORM]


 PURGE cd-name-1


SRI READ file-name-1  [NEXT]  RECORD  [INTO identifier-1]

    [AT END imperative-statement-1]

    [NOT AT END imperative-statement-2]

    [END-READ]


  R READ file-name-1 RECORD  [INTO identifier-1]

    [INVALID KEY imperative-statement-3]

    [NOT INVALID KEY imperative-statement-4]

    [END-READ]
```

```
I  READ file-name-1 RECORD  [INTO identifier-1]

       [KEY IS data-name-1]

       [INVALID KEY imperative-statement-3]

       [NOT INVALID KEY imperative-statement-4]

       [END-READ]


   RECEIVE cd-name-1  {MESSAGE}  INTO identifier-1
                      {SEGMENT}

       [NO DATA imperative-statement-1]

       [WITH DATA imperative-statement-2]

       [END-RECEIVE]


   RELEASE record-name-1  [FROM identifier-1]


   RETURN file-name-1 RECORD  [INTO identifier-1]

       AT END imperative-statement-1

       [NOT AT END imperative-statement-2]

       [END-RETURN]


S  REWRITE record-name-1  [FROM identifier-1]


RI REWRITE record-name-1  [FROM identifier-1]

       [INVALID KEY imperative-statement-1]

       [NOT INVALID KEY imperative-statement-2]

       [END-REWRITE]
```

# GENERAL FORMAT FOR COBOL VERBS

$$\underline{\texttt{SEARCH}}\ \texttt{identifier-1}\ \left[\underline{\texttt{VARYING}}\ \left\{\begin{array}{l}\texttt{identifier-2}\\ \texttt{index-name-1}\end{array}\right\}\right]$$

$$\texttt{[AT }\underline{\texttt{END}}\texttt{ imperative-statement-1]}$$

$$\left\{\underline{\texttt{WHEN}}\ \texttt{condition-1}\ \left\{\begin{array}{l}\texttt{imperative-statement-2}\\ \underline{\texttt{NEXT}}\ \underline{\texttt{SENTENCE}}\end{array}\right\}\right\}\ldots$$

$$\texttt{[}\underline{\texttt{END-SEARCH}}\texttt{]}$$

$$\underline{\texttt{SEARCH}}\ \underline{\texttt{ALL}}\ \texttt{identifier-1}\ \texttt{[AT }\underline{\texttt{END}}\texttt{ imperative-statement-1]}$$

$$\underline{\texttt{WHEN}}\ \left\{\begin{array}{l}\texttt{data-name-1}\ \left\{\begin{array}{l}\texttt{IS }\underline{\texttt{EQUAL}}\texttt{ TO}\\ \texttt{IS =}\end{array}\right\}\ \left\{\begin{array}{l}\texttt{identifier-3}\\ \texttt{literal-1}\\ \texttt{arithmetic-expression-1}\end{array}\right\}\\ \texttt{condition-name-1}\end{array}\right\}$$

$$\left[\underline{\texttt{AND}}\ \left\{\begin{array}{l}\texttt{data-name-2}\ \left\{\begin{array}{l}\texttt{IS }\underline{\texttt{EQUAL}}\texttt{ TO}\\ \texttt{IS =}\end{array}\right\}\ \left\{\begin{array}{l}\texttt{identifier-4}\\ \texttt{literal-2}\\ \texttt{arithmetic-expression-2}\end{array}\right\}\\ \texttt{condition-name-2}\end{array}\right\}\right]\ldots$$

$$\left\{\begin{array}{l}\texttt{imperative-statement-2}\\ \underline{\texttt{NEXT}}\ \underline{\texttt{SENTENCE}}\end{array}\right\}$$

$$\texttt{[}\underline{\texttt{END-SEARCH}}\texttt{]}$$

$$\underline{\texttt{SEND}}\ \texttt{cd-name-1}\ \underline{\texttt{FROM}}\ \texttt{identifier-1}$$

$$\underline{\texttt{SEND}}\ \texttt{cd-name-1}\ \texttt{[}\underline{\texttt{FROM}}\texttt{ identifier-1]}\ \left\{\begin{array}{l}\texttt{WITH identifier-2}\\ \texttt{WITH }\underline{\texttt{ESI}}\\ \texttt{WITH }\underline{\texttt{EMI}}\\ \texttt{WITH }\underline{\texttt{EGI}}\end{array}\right\}$$

$$\left[\left\{\begin{array}{l}\underline{\texttt{BEFORE}}\\ \underline{\texttt{AFTER}}\end{array}\right\}\ \texttt{ADVANCING}\ \left\{\begin{array}{l}\left\{\begin{array}{l}\texttt{identifier-3}\\ \texttt{integer-1}\end{array}\right\}\ \left[\begin{array}{l}\texttt{LINE}\\ \texttt{LINES}\end{array}\right]\\ \left\{\begin{array}{l}\texttt{mnemonic-name-1}\\ \underline{\texttt{PAGE}}\end{array}\right\}\end{array}\right\}\right]$$

$$\texttt{[}\underline{\texttt{REPLACING}}\texttt{ LINE]}$$

$$\underline{\texttt{SET}}\ \left\{\begin{array}{l}\texttt{index-name-1}\\ \texttt{identifier-1}\end{array}\right\}\ \ldots\ \underline{\texttt{TO}}\ \left\{\begin{array}{l}\texttt{index-name-2}\\ \texttt{identifier-2}\\ \texttt{integer-1}\end{array}\right\}$$

## GENERAL FORMAT FOR COBOL VERBS

```
SET  {index-name-3} ...  {UP BY  }  {identifier-3}
                         {DOWN BY}  {integer-2   }


SORT file-name-1  {ON {ASCENDING }  KEY  {data-name-1} ... } ...
                  {   {DESCENDING}                        }

   [WITH DUPLICATES IN ORDER]

   [COLLATING SEQUENCE IS alphabet-name-1]

   {INPUT PROCEDURE IS procedure-name-1  [{THROUGH}  procedure-name-2]}
   {                                     [{THRU   }                  ]}
   {USING  {file-name-2} ...                                          }

   {OUTPUT PROCEDURE IS procedure-name-3  [{THROUGH}  procedure-name-4]}
   {                                      [{THRU   }                  ]}
   {GIVING  {file-name-3} ...                                          }

                         [     {IS EQUAL TO                  }               ]
                         [     {IS =                         }               ]
                         [     {IS GREATER THAN              }               ]
                         [     {IS >                         }               ]
START file-name-1        [ KEY {IS NOT LESS THAN             }  data-name-1  ]
                         [     {IS NOT <                     }               ]
                         [     {IS GREATER THAN OR EQUAL TO  }               ]
                         [     {IS >=                        }               ]

   [INVALID KEY imperative-statement-1]

   [NOT INVALID KEY imperative-statement-2]

   [END-START]


STOP  {RUN      }
      {literal-1}
```

## GENERAL FORMAT FOR COBOL VERBS

```
                                               (identifier-2) )
STRING  {(identifier-1)}  ...   DELIMITED BY  { literal-2    } } ...
        {(literal-1   )}                       (SIZE         ) )

   INTO identifier-3

   [WITH POINTER identifier-4]

   [ON OVERFLOW imperative-statement-1]

   [NOT ON OVERFLOW imperative-statement-2]

   [END-STRING]


SUBTRACT  {identifier-1}  ...   FROM  {identifier-3 [ROUNDED]} ...
          {literal-1   }

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-SUBTRACT]


SUBTRACT  {identifier-1}  ...   FROM  {identifier-2}
          {literal-1   }              {literal-2   }

   GIVING  {identifier-3 [ROUNDED]} ...

   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-SUBTRACT]


SUBTRACT  {CORRESPONDING}  identifier-1 FROM identifier-2 [ROUNDED]
          {CORR         }
   [ON SIZE ERROR imperative-statement-1]

   [NOT ON SIZE ERROR imperative-statement-2]

   [END-SUBTRACT]


SUPPRESS PRINTING


TERMINATE  {report-name-1} ...
```

```
UNSTRING  identifier-1

   [                    {identifier-2}  [         {identifier-3}]     ]
   [DELIMITED BY [ALL]  {literal-1   }  [OR [ALL] {literal-2   }] ... ]

   INTO {identifier-4 [DELIMITER IN identifier-5]  [COUNT IN identifier-6]} ...

   [WITH POINTER identifier-7]

   [TALLYING IN identifier-8]

   [ON OVERFLOW imperative-statement-1]

   [NOT ON OVERFLOW imperative-statement-2]

   [END-UNSTRING]


                                                     ({file-name-1} ...)
                                                     (INPUT            )
SRI USE [GLOBAL] AFTER STANDARD {EXCEPTION} PROCEDURE ON <OUTPUT           >
                                {ERROR    }          (I-O              )
                                                     (EXTEND           )


                                             ({file-name-1} ...)
  W USE AFTER STANDARD {EXCEPTION} PROCEDURE ON <OUTPUT           >
                       {ERROR    }           (EXTEND           )


   USE [GLOBAL] BEFORE REPORTING identifier-1


                         (cd-name-1                              )
                         ([ALL REFERENCES OF] identifier-1       )
   USE FOR DEBUGGING ON  <file-name-1                            > ...
                         (procedure-name-1                       )
                         (ALL PROCEDURES                         )
```

```
S WRITE record-name-1  [FROM identifier-1]

    ⎡                      ⎧ ⎧identifier-2⎫  ⎡LINE ⎤ ⎫ ⎤
    ⎢ ⎧BEFORE⎫             ⎪ ⎩integer-1   ⎭  ⎣LINES⎦ ⎪ ⎥
    ⎢ ⎩AFTER ⎭  ADVANCING  ⎨                         ⎬ ⎥
    ⎢                      ⎪ ⎧mnemonic-name-1⎫       ⎪ ⎥
    ⎣                      ⎩ ⎩PAGE           ⎭       ⎭ ⎦

    ⎡AT  ⎧END-OF-PAGE⎫  imperative-statement-1⎤
    ⎣    ⎩EOP        ⎭                        ⎦

    ⎡NOT AT  ⎧END-OF-PAGE⎫  imperative-statement-2⎤
    ⎣        ⎩EOP        ⎭                        ⎦

    [END-WRITE]


RI WRITE record-name-1  [FROM identifier-1]

    [INVALID KEY imperative-statement-1]

    [NOT INVALID KEY imperative-statement-2]

    [END-WRITE]
```

# GENERAL FORMAT FOR COPY AND REPLACE STATEMENTS

$$\underline{\text{COPY}}\ \text{text-name-1}\ \left[\left\{\begin{array}{l}\underline{\text{OF}}\\ \underline{\text{IN}}\end{array}\right\}\ \text{library-name-1}\right]$$

$$\left[\underline{\text{REPLACING}}\ \left\{\left\{\begin{array}{l}\text{==pseudo-text-1==}\\ \text{identifier-1}\\ \text{literal-1}\\ \text{word-1}\end{array}\right\}\ \underline{\text{BY}}\ \left\{\begin{array}{l}\text{==pseudo-text-2==}\\ \text{identifier-2}\\ \text{literal-2}\\ \text{word-2}\end{array}\right\}\right\}\ \ldots\right]$$

$$\underline{\text{REPLACE}}\ \ \{\text{==pseudo-text-1==}\ \ \underline{\text{BY}}\ \ \text{==pseudo-text-2==}\}\ \ldots$$

$$\underline{\text{REPLACE}}\ \underline{\text{OFF}}$$

## GENERAL FORMAT FOR CONDITIONS

### RELATION CONDITION:

$$\left\{\begin{array}{l}\text{identifier-1}\\ \text{literal-1}\\ \text{arithmetic-expression-1}\\ \text{index-name-1}\end{array}\right\}\left\{\begin{array}{l}\text{IS [}\underline{\text{NOT}}\text{] }\underline{\text{GREATER}}\text{ THAN}\\ \text{IS [}\underline{\text{NOT}}\text{] >}\\ \text{IS [}\underline{\text{NOT}}\text{] }\underline{\text{LESS}}\text{ THAN}\\ \text{IS [}\underline{\text{NOT}}\text{] <}\\ \text{IS [}\underline{\text{NOT}}\text{] }\underline{\text{EQUAL}}\text{ TO}\\ \text{IS [}\underline{\text{NOT}}\text{] =}\\ \text{IS }\underline{\text{GREATER}}\text{ THAN }\underline{\text{OR}}\text{ }\underline{\text{EQUAL}}\text{ TO}\\ \text{IS >=}\\ \text{IS }\underline{\text{LESS}}\text{ THAN }\underline{\text{OR}}\text{ }\underline{\text{EQUAL}}\text{ TO}\\ \text{IS <=}\end{array}\right\}\left\{\begin{array}{l}\text{identifier-2}\\ \text{literal-2}\\ \text{arithmetic-expression-2}\\ \text{index-name-2}\end{array}\right\}$$

### CLASS CONDITION:

$$\text{identifier-1 IS [}\underline{\text{NOT}}\text{]}\left\{\begin{array}{l}\underline{\text{NUMERIC}}\\ \underline{\text{ALPHABETIC}}\\ \underline{\text{ALPHABETIC-LOWER}}\\ \underline{\text{ALPHABETIC-UPPER}}\\ \text{class-name-1}\end{array}\right\}$$

### CONDITION-NAME CONDITION:

condition-name-1

### SWITCH-STATUS CONDITION:

condition-name-1

### SIGN CONDITION:

$$\text{arithmetic-expression-1 IS [}\underline{\text{NOT}}\text{]}\left\{\begin{array}{l}\underline{\text{POSITIVE}}\\ \underline{\text{NEGATIVE}}\\ \underline{\text{ZERO}}\end{array}\right\}$$

### NEGATED CONDITION:

$\underline{\text{NOT}}$ condition-1

## GENERAL FORMAT FOR CONDITIONS

### COMBINED CONDITION:

$$\texttt{condition-1} \left\{ \begin{Bmatrix} \underline{\text{AND}} \\ \underline{\text{OR}} \end{Bmatrix} \; \texttt{condition-2} \right\} \ldots$$

### ABBREVIATED COMBINED RELATION CONDITION:

$$\texttt{relation-condition} \left\{ \begin{Bmatrix} \underline{\text{AND}} \\ \underline{\text{OR}} \end{Bmatrix} \; [\underline{\text{NOT}}] \; [\texttt{relational-operator}] \; \texttt{object} \right\} \ldots$$

## GENERAL FORMAT FOR QUALIFICATION

FORMAT 1:

$$\left\{\begin{matrix}\text{data-name-1}\\ \text{condition-name-1}\end{matrix}\right\} \left\{\begin{matrix} \left\{\left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{data-name-2}\right\} \ldots \left[\left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \left\{\begin{matrix}\text{file-name-1}\\ \text{cd-name-1}\end{matrix}\right\}\right] \\ \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \left\{\begin{matrix}\text{file-name-1}\\ \text{cd-name-1}\end{matrix}\right\} \end{matrix}\right\}$$

FORMAT 2:

$$\text{paragraph-name-1} \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{section-name-1}$$

FORMAT 3:

$$\text{text-name-1} \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{library-name-1}$$

FORMAT 4:

$$\underline{\text{LINAGE-COUNTER}} \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{file-name-2}$$

FORMAT 5:

$$\left\{\begin{matrix}\underline{\text{PAGE-COUNTER}}\\ \underline{\text{LINE-COUNTER}}\end{matrix}\right\} \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{report-name-1}$$

FORMAT 6:

$$\text{data-name-3} \left\{\begin{matrix} \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{data-name-4} \left[\left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{report-name-2}\right] \\ \left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \text{report-name-2} \end{matrix}\right\}$$

## MISCELLANEOUS FORMATS

### SUBSCRIPTING:

$$\left\{\begin{array}{l}\text{condition-name-1}\\ \text{data-name-1}\end{array}\right\}\ \ (\ \ \left\{\begin{array}{l}\text{integer-1}\\ \text{data-name-2 } [\{\pm\}\ \text{integer-2}]\\ \text{index-name-1 } [\{\pm\}\ \text{integer-3}]\end{array}\right\}\ \ldots\ )$$

### REFERENCE MODIFICATION:

data-name-1 (leftmost-character-position: [length])

### IDENTIFIER:

$$\text{data-name-1}\ \ \left[\left\{\begin{array}{l}\underline{\text{IN}}\\ \underline{\text{OF}}\end{array}\right\}\ \text{data-name-2}\right]\ \ldots\ \left[\left\{\begin{array}{l}\underline{\text{IN}}\\ \underline{\text{OF}}\end{array}\right\}\left\{\begin{array}{l}\text{cd-name-1}\\ \text{file-name-1}\\ \text{report-name-1}\end{array}\right\}\right]$$

[({subscript} ... )] [(leftmost-character-position: [length])]

## GENERAL FORMAT FOR NESTED SOURCE PROGRAMS

IDENTIFICATION DIVISION.

PROGRAM-ID. program-name-1 [IS INITIAL PROGRAM].

[ENVIRONMENT DIVISION. environment-division-content]

[DATA DIVISION. data-division-content]

[PROCEDURE DIVISION. procedure-division-content]

[[nested-source-program] ...

END PROGRAM program-name-1.]

## GENERAL FORMAT FOR NESTED-SOURCE-PROGRAM

IDENTIFICATION DIVISION.

PROGRAM-ID. program-name-2 $\left[\text{IS} \left\{\left\|\begin{array}{l}\underline{\text{COMMON}}\\ \underline{\text{INITIAL}}\end{array}\right\|\right\} \text{PROGRAM}\right]$ .

[ENVIRONMENT DIVISION. environment-division-content]

[DATA DIVISION. data-division-content]

[PROCEDURE DIVISION. procedure-division-content]

[nested-source-program] ...

END PROGRAM program-name-2.

# GENERAL FORMAT FOR A SEQUENCE OF SOURCE PROGRAMS

```
{IDENTIFICATION DIVISION.

 PROGRAM-ID.  program-name-3  [IS INITIAL PROGRAM].

[ENVIRONMENT DIVISION.  environment-division-content]

[DATA DIVISION.  data-division-content]

[PROCEDURE DIVISION.  procedure-division-content]

[nested-source-program] ...

 END PROGRAM program-name-3.} ...

 IDENTIFICATION DIVISION.

 PROGRAM-ID.  program-name-4  [IS INITIAL PROGRAM].

[ENVIRONMENT DIVISION.  environment-division-content]

[DATA DIVISION.  data-division-content]

[PROCEDURE DIVISION.  procedure-division-content]

[[nested-source-program] ...

 END PROGRAM program-name-4.]
```

## Intrinsic Functions

FUNCTION ACOS (number-1)

FUNCTION ANNUITY (numeric-item-1 integer-item-1)

FUNCTION ASIN (number-1)

FUNCTION ATAN (number-1)

FUNCTION CHAR (integer-item-1)

FUNCTION COS (number-1)

FUNCTION CURRENT-DATE

FUNCTION DATE-OF-INTEGER (integer-1)

FUNCTION DATE-TO-YYYYMMDD (integer▌item-1 [integer-item-2])

FUNCTION DAY-OF-INTEGER (integer-1)

FUNCTION DAY-TO-YYYYDDD (integer▌item-1 [integer-item-2])

FUNCTION FACTORIAL (integer-item-1)

FUNCTION INTEGER (numeric-item-1)

FUNCTION INTEGER-OF-DATE (integer-1)

FUNCTION INTEGER-OF-DAY (integer-1)

FUNCTION INTEGER-PART (numeric-item-1)

FUNCTION LENGTH (data-item-1)

FUNCTION LOG (number-1)

FUNCTION LOG10 (number-1)

FUNCTION LOWER-CASE (AN-data-item-1)

FUNCTION MAX ( {argument-1}▌)

FUNCTION MEAN ( {numeric-item-1}▌)

FUNCTION MEDIAN ( {numeric-item-1}▌)

FUNCTION MIDRANGE ( {numeric-item-1}▌)

FUNCTION MIN ( {argument-1}▌)

FUNCTION MOD (integer-item-1 integer-item-2)

```
FUNCTION NUMVAL (AN-data-item-1)

FUNCTION NUMVAL-C (AN-data-item-1 [char-item-1])

FUNCTION ORD (char-item-1)

FUNCTION ORD-MAX ( {argument-1}▌)

FUNCTION ORD-MIN ( {argument-1}▌)

FUNCTION PRESENT-VALUE (numeric-item-1 {numeric-item-2}▌)

FUNCTION RANDOM ([integer-item-1])

FUNCTION RANGE ( {numeric-item-1}▌)

FUNCTION REM (numeric-item-1 numeric-item-2)

FUNCTION REVERSE (AN-data-item-1)

FUNCTION SIN (number-1)

FUNCTION SQRT (numeric-item-1)

FUNCTION STANDARD-DEVIATION ( {numeric-item-1}▌)

FUNCTION SUM ( {numeric-item-1}▌)

FUNCTION TAN (number-1)

FUNCTION TEST-DATE-YYYYMMDD (integer▌item-1)

FUNCTION TEST-DAY-YYYYDDD (integer▌item-1)

FUNCTION UPPER-CASE (AN-data-item-1)

FUNCTION VARIANCE ( {numeric-item-1}▌)

FUNCTION WHEN-COMPILED

FUNCTION YEAR-TO-YYYY (integer▌item-1 [integer-item-2])
```