CS 1024
SUMMARY SHEET 2

Basic Arithmetic Operations

1. ADD ... TO .... GIVING ... [ROUNDED] [ON SIZE ERROR ...] [END-ADD]
2. SUBTRACT ... FROM .... GIVING ... [ROUNDED] [ON SIZE ERROR ...] [END-SUBTRACT]
3. DIVIDE ... BY ... GIVING... [ROUNDED] [ON SIZE ERROR ...] [END-DIVIDE]
4. DIVIDE ... INTO ... GIVING ...[ROUNDED] [ON SIZE ERROR] ... [END-DIVIDE]
5. DIVIDE .......... GIVING ... [ROUNDED] REMAINDER ... [ON SIZE ERROR ...] [END-DIVIDE]
6. MULTIPLY ... BY ... GIVING ... [ROUNDED] [ON SIZE ERROR ...] [END-MULTIPLY]

COMPUTE Statement

COMPUTE  field name [ROUNDED] =  expression  [ON SIZE ERROR ...] [END-COMPUTE]

ORDER OF EVALUATION:
1. Parenthesis
2. Unary minus sign (negation)
3. Exponentiation
4. Multiplication and division, left to right
5. Addition and subtraction, left to right

Selection Structure

1. Basic form of IF statement:
   IF condition THEN
   {statements}
   ELSE
   {statements}
   END-IF
2. One-sided IF statement:
   IF condition THEN
   {statements}
   END-IF
3. EVALUATE statement:
   EVALUATE  field-name
   WHEN ... [THROUGH ....]
   {statements}
   WHEN OTHER
   {statements}
   END-EVALUATE

Relation Conditions

>       GREATER THAN
<=      LESS THAN OR EQUAL TO
<       NOT GREATER THAN
>=      GREATER THAN OR EQUAL TO
NOT >   NOT GREATER THAN
NOT <=  NOT LESS THAN
NOT =   NOT EQUAL TO
Collating Sequence

For ASCII: Lowest to Highest
- Blank
- Special characters
- Integers 0 - 9
- Special characters
- Uppercase letters A-Z
- Lowercase letters a-z

Loop Structure

1. Out-of-line loop:
   PERFORM  paragraph-name  UNTIL condition
2. In-line loop:
   PERFORM UNTIL  condition
   {statements}
   END-PERFORM
3. Nested loop pattern:
   a) With in-line loops:
      PERFORM UNTIL condition-1
      :   PERFORM UNTIL condition-2
      :   END-PERFORM
      :   END-PERFORM
   b) With out-of-line loops:
      PERFORM  outer-loop-name  UNTIL condition-1
      :   outer-loop-name.
      PERFORM inner-loop-name  UNTIL condition-2
      :   inner-loop-name.
      {statements}

Rules for MOVE Operation

1. Numeric move occurs when both the sending and the receiving fields are both numeric
   a) Movement is from right to left.  b) Unfilled higher-order positions are replaced with zeros.
   c) Truncation of higher-order digits occurs if the receiving field is not large enough to hold the results.
   Decimal portion:
   a) Movement is left to right, beginning at the decimal point.
   b) Unfilled low-order positions are filled with zeros.
2. Nonnumeric move occurs when the receiving field is nonnumeric
   a) Left to right  b) Unfilled low-order positions are replaced with spaces
   c) Truncation of low-order character occurs if the receiving field is not large enough to hold the results.