INFO3105 Week 3 Class 1

Review

- Numeric Edited fields
- Lab4

Introduction to Summary Report Programming with COBOL

Definition of a Control Break – "***A change of category used to trigger a subtotal. For example, if data are sub totaled by province, a control break occurs when BC changes to ON***." Lots of COBOL programs/reports are written as **control break** programs. The first pre-requisite to writing a control break program has nothing to do with the program itself; rather it's the data that you need to be concerned with. The data must be sorted on the control break field; therefore if we were using the example given in the above definition, the data would have had to be sorted on province. Below we see the layout of the control break program we'll be working on next:

```
  PAGE   1                          ABC CORPORATION                          01/17/2014
                                 SALESPERSON BY BRANCH

BRANCH: 100
LAST NAME       FIRST NAME    GROSS SALES       RETURNS         NET SALES     COMMISSION   R
Laueren           Evan         $63,222.23      $4,244.32        $58,977.91     $1,963.96
Orlando           Randolph     $70,814.29        $322.58        $70,491.71     $2,072.46
Rowan             Eileen       $77,317.12        $838.83        $76,478.29     $3,059.13
DeGaetano         Catherine    $41,516.79        $231.82        $41,284.97     $1,589.47
Flynn             Ashley       $77,374.53        $127.38        $77,247.15     $2,572.33
Hau               Jayne        $70,896.27         $13.29        $70,882.98     $1,694.10
Steele            Karen        $77,360.06        $128.99        $77,231.07     $2,602.69
Baker             Anna         $70,834.11        $443.37        $70,390.74     $2,660.77
Appel             Anne         $70,813.31        $210.11        $70,603.20     $2,577.02
Patchik           Joseph       $21,952.56        $117.54        $21,835.02       $842.83
Banasiak          Nancy        $70,840.52      $1,113.91        $69,726.61     $2,091.80
TOTAL BRANCH 100              $712,941.79      $7,792.14       $705,149.65    $23,726.56
                              -------------    ----------     -------------   -----------
  PAGE   2                          ABC CORPORATION                          01/17/2014
                                 SALESPERSON BY BRANCH

BRANCH: 200
LAST NAME       FIRST NAME    GROSS SALES       RETURNS         NET SALES     COMMISSION   R
Mosak             Brian        $31,272.01        $236.73        $31,035.28     $1,170.03
Doherty           Derek        $31,211.71        $124.23        $31,087.48     $1,025.89
Anderson          Susan        $81,512.17        $910.77        $80,601.40     $2,933.89   *
Ziccardi          Brandon      $70,861.70        $251.03        $70,610.67     $2,492.56
```

Notice that the report is written so that all of the salespeople from a particular branch are listed on a single page. Which field would the data for this report sorted by? Well obviously it's **the branch number field**. Below is some of the data used by this report, see if you can locate the branch information in it:

```
----+----1----+----2----+----3----+----4----+----5--
02153Laueren          Evan       6322223424432100033
07700Orlando          Randolph   7081429032258100029
02613Rowan            Eileen     7731712083883100040
02690DeGaetano        Catherine  4151679023182100038
02158Flynn            Ashley     7737453012738100033
26002Hau              Jayne      7089627001329100023
26916Steele           Karen      7736006012899100033
24930Baker            Anna       7083411044337100037
24927Appel            Anne       7081331021011100036
26868Patchik          Joseph     2195256011754100038
24302Banasiak         Nancy      7084052111391100030
04632Mosak            Brian      3127201023673200037
05929Doherty          Derek      3121171012423200033
05976Anderson         Susan      8151217091077200036
08887Ziccardi         Brandon    7086170025103200035
04852Marcus           Aili       7089209109829200035
01746Brockmann        Nelson     2190102012192200028
04743Corbett          Lynn       7088322011198200021
09223VanLanen         Joanna     7739644012307200021
10216King             Bryan      3127598111941200029
10559McGannon         Margaret   6307920021961200034
10507Canario          Margaret   7082191066713200038
19304Cavaretta        Katherine  6300961200216200033
22643Orlando          Catherine  7089186032099200040
26927Appel            Judy       8476177300336200012
24781Kimble           Kathleen   3120509012204200032
26867Fricker          Kathleen   7734299044727200038
```

The data here shows salespeople for branches 100 and 200

And here is the file layout for this data:

```
* COPY BOOK FOR SALESPERSON MASTER FILE
 01 SALESPERSON-MASTER.
       05 SALESPERSON-NO              PIC 9(5).
       05 SALESPERSON-LAST-NAME       PIC X(15).
       05 SALESPERSON-FIRST-NAME      PIC X(10).
       05 SALESPERSON-GROSS-SALES     PIC 9(5)V99.
       05 SALESPERSON-RETURN-SALES    PIC 9(4)V99.
       05 SALESPERSON-BRANCH-NO       PIC 9(3).
       05 SALESPERSON-COMM-RATE       PIC V9999.
```

So returning to the output we see we need to produce at least 3 different types of output:

1. Heading information (at the top of page)
2. Detailed information (actual sales person data)
3. Summary information (branch totals).

So what you need to do is lay the report out to see where everything fits, historically programmers would use something called a Printer Spacing Chart **see page 95** of the text for an example. Then once the design of the report is done, you can map the columns to the layout in COBOL, for instance here is the 1st Heading Line for the report from page 1:

```
01  WS-HEADING-LINE-1.
    05  FILLER              PIC X(5)    VALUE SPACES.
    05  FILLER              PIC X(6)    VALUE "PAGE  ".
    05  WS-HL1-PAGENO       PIC Z9.
    05  FILLER              PIC X(29)   VALUE SPACES.
    05  FILLER              PIC X(15)   VALUE "ABC CORPORATION".
    05  FILLER              PIC X(27)   VALUE SPACES.
    05  WS-HL1-MONTH        PIC 9(2).
    05  FILLER              PIC X(1)    VALUE "/".
    05  WS-HL1-DAY          PIC 9(2).
    05  FILLER              PIC X(1)    VALUE "/".
    05  WS-HL1-YEAR         PIC 9(4).
    05  FILLER              PIC X       VALUE SPACES.
```

We see here there are some fields designated with the keyword "**FILLER**". This tells the compiler basically what we have is some space to allocate but we can't reference it directly as variable. Use filler when the data isn't important to the logic of the program. Constant strings are embedded in quotation marks this is known traditionally as **hard coding** the data. The keyword **SPACES** is used to put blanks in each character designated by the PIC X(..) clause. So instead of counting out 5 actual spaces we can just write it as **PIC X(5) VALUE SPACES**. One other point to take note of is the picture clauses used here. PIC 9 is for numeric data, PIC X is for alphanumeric, and PIC Z9 is for formatted numeric data, it indicates if there is number is not large enough to encompass the entire picture clause just place spaces (instead of zeroes) in the leading characters.

Returning to the listing on page 1, how many different lines of output do you need to create? The answer is 7 (there are actually 9 but the 8th and 9th aren't visible here, see below for the final lines example).

We saw how to write the individual lines out in our labs 1-3. In labs 1-3 the output was basically the same as our input. Here the last two report columns do not exist in the data but are actually to be calculated by our program based on the data. The net sales

column is simply gross sales subtract returns. The commission column is calculated as net sales multiplied by a commission rate (last column of the input data).

To summarize, the 9 different line layouts you need to create are:

1. Heading line 1 – page no, and date
2. Heading line 2 – constant with Salesperson By Branch text
3. Heading line 3 – Branch indicator
4. Heading line 4 – Individual Column Headings
5. Detail line – containing individual salesperson data
6. Branch Total – containing branch totals (formatted) for each columns
7. Single Underlines – typical in a report like this under the branch totals (see below)
8. Grand Totals – containing accumulation of Branch Totals (formatted)
9. Double Underlines – indicates the end of Report (just use the "=" instead of '-')

End of the Report

Roe	Eileen	$70,824.38	$118.83	$70,705.55	$2,785.80
Holtz	Rachel	$63,072.26	$127.09	$62,945.17	$2,517.81
Hazard	Mary	$70,807.16	$1,121.74	$69,685.42	$2,209.03
Ford	Dalia	$77,331.45	$1,326.30	$76,005.15	$1,565.71
Doherty	Elizabeth	$81,593.90	$211.72	$81,382.18	$2,465.88
Stockover	Nancy	$70,823.37	$112.03	$70,711.34	$2,149.62
TOTAL BRANCH 500		$804,061.01	$5,190.57	$798,870.44	$26,649.82
COMPANY TOTALS		$6,036,605.92	$65,693.15	$5,970,912.77	$195,565.65

Detail Line Formatting

The first four headings are relatively straight forward to layout. Line 5 (the detail line) will ask you to format the numeric fields and the last two fields need to be "Computed". **Page 195** of the text lists all of the different ways to format data. We'll be using something similar to the last one with the $ signs but without CR designation.

COBOL only lets you do math on numeric fields**, NOT on numeric edited** fields. So you'll have to typically follow this pattern with numeric fields that require calculations:

1. move the input to a working storage field
2. do some math on the working storage field
3. move the working storage field to the numeric edited field

Let's see the syntax for doing the 3 steps by looking at what goes on in the detail line with the Net Sales column: Assuming we have done our READ, we would calculate our Net Sales field as follows:

- Calculate the difference of the Gross Sales and Sales Returns :
  COMPUTE WS-NET-SALES = SALESPERSON-GROSS-SALES -
  SALESPERSON-RETURN-SALES

- Move the working storage field (unedited) to the edited field
    - `05  WS-NET-SALES            PIC S9(7)V99   VALUE ZERO.`
    - `..`
    - `..`
    - `05  WS-SL-SLSP-NET          PIC $$$,$$$.99.`
    - `..`
    - `..`
    - `MOVE WS-NET-SALES TO WS-SL-SLSP-NET`

When calculating commissions we need to worry about rounding and we can do this with the keyword ROUNDED (page 32 of the text) as follows:

```
COMPUTE
   WS-COMMISSION-EARNED ROUNDED = (SALESPERSON-GROSS-SALES -
   SALESPERSON-RETURN-SALES) * SALESPERSON-COMM-RATE
END-COMPUTE
```

Lab 5 - 2%

- Submit to the FOL dropbox (no need to get the full program working yet):
    1. the source code for the 9 different output lines (including output field definitions)
    2. the source code for a paragraph called SALESPERSON-CALCULATIONS that contains all of the compute statements for the report (remember you'll need both branch and grand totals)

- Also, read **pages 316-318** of the text.