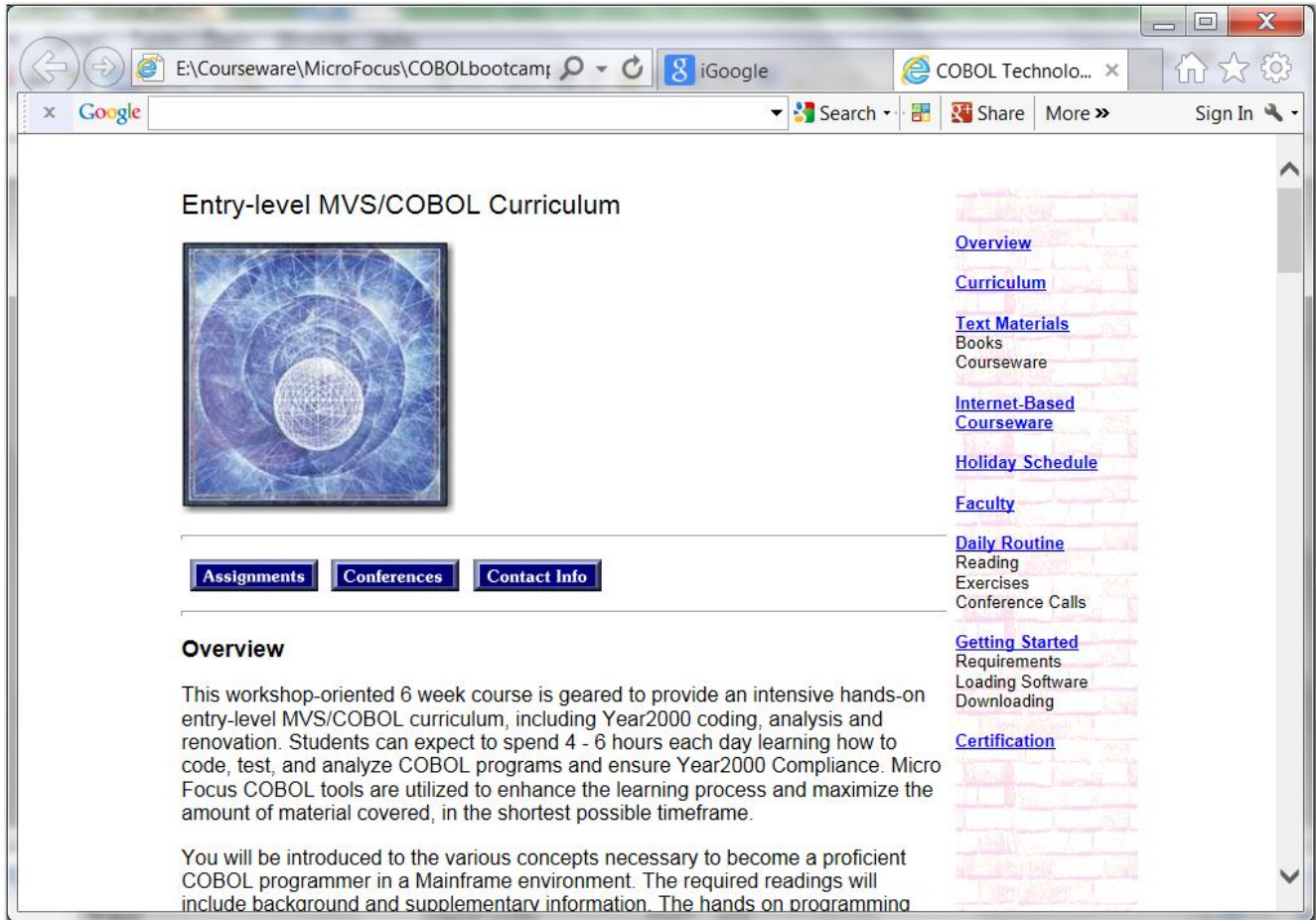



The package depicted below is for an online Entry-Level MVS/COBOL course delivered by Micro Focus, Inc.



Entry-level MVS/COBOL Curriculum



[Assignments](#) [Conferences](#) [Contact Info](#)

Overview

This workshop-oriented 6 week course is geared to provide an intensive hands-on entry-level MVS/COBOL curriculum, including Year2000 coding, analysis and renovation. Students can expect to spend 4 - 6 hours each day learning how to code, test, and analyze COBOL programs and ensure Year2000 Compliance. Micro Focus COBOL tools are utilized to enhance the learning process and maximize the amount of material covered, in the shortest possible timeframe.

You will be introduced to the various concepts necessary to become a proficient COBOL programmer in a Mainframe environment. The required readings will include background and supplementary information. The hands on programming

- [Overview](#)
- [Curriculum](#)
- [Text Materials](#)
 - Books
 - Courseware
- [Internet-Based Courseware](#)
- [Holiday Schedule](#)
- [Faculty](#)
- [Daily Routine](#)
 - Reading
 - Exercises
 - Conference Calls
- [Getting Started](#)
 - Requirements
 - Loading Software
 - Downloading
- [Certification](#)

A sample session on control break logic.

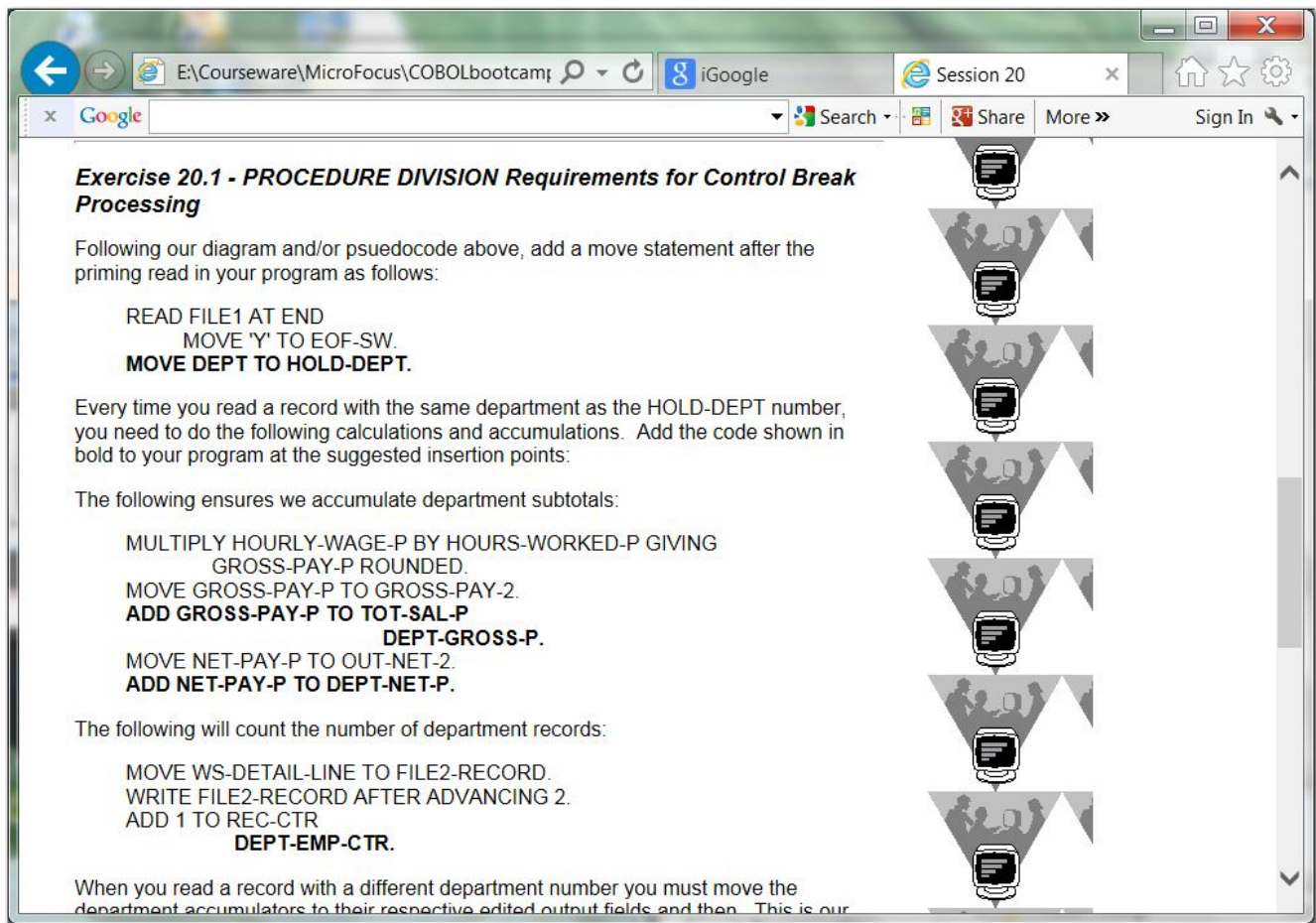
Session 20

In this session we are going to finish up the PROCEDURE DIVISION of PROG12 to add the control break logic required to produce the departmental summaries. As a reminder, we'll review the processing requirements described in the previous Session. First, review the flowchart illustration:

```
graph TD; Start(( )) --> ReadA[READ A RECORD]; ReadA --> Move[MOVE DET TO WS-DEPT]; Move --> ReadA; ReadA --> Eof{EOF?}; Eof -- Y --> WriteDetail1[/WRITE DETAIL/]; WriteDetail1 --> WriteSubtotal1[/WRITE SUBTOTAL/]; WriteSubtotal1 --> WriteFinalTotals[/WRITE FINAL TOTALS/]; Eof -- N --> IsDept{IS DEPT = WS-DEPT?}; IsDept -- Y --> WriteDetail2[/WRITE DETAIL/]; WriteDetail2 --> Start; IsDept -- N --> WriteSubtotal2[/WRITE SUBTOTAL/]; WriteSubtotal2 --> Start;
```

The flowchart illustrates the control break logic for departmental summaries. It starts with a loop: 'READ A RECORD' followed by 'MOVE DET TO WS-DEPT' and then 'READ A RECORD'. A decision diamond 'EOF?' checks for the end of the file. If 'Y' (Yes), it proceeds to 'WRITE DETAIL', then 'WRITE SUBTOTAL', and finally 'WRITE FINAL TOTALS'. If 'N' (No), it proceeds to another decision diamond 'IS DEPT = WS-DEPT?'. If 'Y' (Yes), it proceeds to 'WRITE DETAIL' and loops back to the start of the record loop. If 'N' (No), it proceeds to 'WRITE SUBTOTAL' and loops back to the start of the record loop.

A sample session on control break logic continued.



The screenshot shows a web browser window with the following content:

Exercise 20.1 - PROCEDURE DIVISION Requirements for Control Break Processing

Following our diagram and/or psuedocode above, add a move statement after the priming read in your program as follows:

```
READ FILE1 AT END  
  MOVE 'Y' TO EOF-SW.  
MOVE DEPT TO HOLD-DEPT.
```

Every time you read a record with the same department as the HOLD-DEPT number, you need to do the following calculations and accumulations. Add the code shown in bold to your program at the suggested insertion points:

The following ensures we accumulate department subtotals:

```
MULTIPLY HOURLY-WAGE-P BY HOURS-WORKED-P GIVING  
  GROSS-PAY-P ROUNDED.  
MOVE GROSS-PAY-P TO GROSS-PAY-2.  
ADD GROSS-PAY-P TO TOT-SAL-P  
  DEPT-GROSS-P.  
MOVE NET-PAY-P TO OUT-NET-2.  
ADD NET-PAY-P TO DEPT-NET-P.
```

The following will count the number of department records:

```
MOVE WS-DETAIL-LINE TO FILE2-RECORD.  
WRITE FILE2-RECORD AFTER ADVANCING 2.  
ADD 1 TO REC-CTR  
DEPT-EMP-CTR.
```

When you read a record with a different department number you must move the department accumulators to their respective edited output fields and then... This is our

The browser window includes a search bar, navigation buttons, and a sidebar with a vertical scroll bar. The page title is "Session 20".