Cobol Exercise

Updating a Sequential File

READ, WRITE, PERFORM..UNTIL, EVALUATE, DISPLAY

Introduction

The transaction file (Transfer.Dat) contains records of students who have transferred from one University of Limerick (UL) course to another. The file is sequenced on ascending StudentId. Using this file, update the records in the student file (Students.Dat). Do this by creating a new file (Students.New) which contains the updated records. The students file and the new file are sequential files held in ascending StudentId order.

We are guaranteed that there will not be more than one transaction record for a particular Student.

The program must be able to detect two kinds of error;

1. It must be able to detect when a record in the transaction file does not have a matching record in the student file.
2. It must be able to detect (in mataching records) when the OldCourseCode of the transaction record is not the same as the CourseCode in the student record.

Download Students.Dat and save it to the WorkArea directory on the drive D:

Download Transfer.Dat and save it to the same place.

File Descriptions

The transaction file is a sequential file ordered on ascending StudentId.
The records in the transaction file (Transfer.Dat) have the following description;

Field	Type	Length	Value
Student Id	9	7	0-9999999
Old Course Code	X	4	-
New Course Code	X	4	-

The students file is a sequential file ordered on ascending StudentId.
The records in the students file and the new file have the same description.
Records in these files have the following description;

Field	Type	Length	Value
Student Id	9	7	0-9999999
Student Name	.	.	Group
Surname	X	8	-
Initials	X	2	-
DateOfBirth	.	.	Group
Year	9	2	00-99
Month	9	2	01-12

Day	9	2	01-31
Course Code	X	4	-
Grant	9	4	0000-9999
Gender	X	1	M/F

Suggested Approaches

This is quite a tricky problem so you may have some difficulty devising a solution to it. Here are some suggestions that may help you.

You need to work out what must to be done to solve the problem before you can write the program to solve it. One way of finding out what must be done is to solve the problem yourself on paper and note what you had to do. You can do this by running through some simple test data.

In the example test data which follows **T** represents the records in the transaction file and **S** represents the records in the student file. The numbers represent the StudentIds of the records. A "y" following the transaction number means the CourseCodes are ok, an "n" means that the CourseCodes do not match and an "x" means it is not relevant.

```
T       S
05y     01
15x     05
25y     10
50y     25
        50
```

Go through the test data and answer the following questions;

1. When T < S (i.e. when the transaction StudentId is less than the StudentId in the student file) what file(s) must be read? Must a record be written to the new file?
2. When T = S what file(s) must be read? Must a record be written to the new file? Is there anything else that needs to be taken into consideration?
3. When T > S what file(s) must be read? Must a record be written to the new file?

Sample Solution

When you have written your program and have compiled it and have it working correctly you may wish to compare it with this
sample solution.

WARNING

As always please do not look at the solution until you have finished your own program. At the very least you should make a substantial effort to complete your own attempt at the program before examining the sample solution.

To COBOL Exercises | Please fill out Evaluation Form | Search the Search CSIS Web Site | Selectable Site Contents UL Home Page ▼ Go!

Last updated :April 1997
e-mail : CSISwebeditor@ul.ie