UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS
Department of Computer Science
and
Information Systems

Repeat Assessment Paper

Academic Year: 2006/2007 **Semester:** 2
Module Title: Business Oriented Programming Languages **Module Code**: CS4556
Duration of Exam: 2.5 hours **% of Total Marks:** 100%
Lecturer : Michael Coughlan **Paper marked out of :** 100

Instructions to Candidates.

3 questions. Attempt (either Question 1 **OR** Question 2) **AND** Question 3.
All questions carry equal weight (i.e. 50 marks)

Question 1	Programming	(50 marks)
	OR	
Question 2	Program Design	(50 marks)
	AND	
Question 3	General theory and programming	(50 marks)

Q1. TV License File Update

At the end of each week *An Post* applies a transaction file of updates to the TVLicenses file creating a new TVLicenses file containing the updated records. The kinds of update applied are: inserting the records of new license holders, registering a new date of license for those who have renewed their licenses, and deleting the license records of those who no longer require them.

The TVLicenses file and the NewTVLicenses file are sequential files ordered on ascending LicenseNumber. Records in these files have the following description:

TVLicenses file and New TVLicenses file record

FIELD	TYPE	LENGTH	VALUES
LicenseNumber	9	7	0000001-9999999
HolderName	X	25	-
HolderAddress	X	55	-
DateOfLicense	9	8	YYYYMMDD

The transaction file is a sequential file ordered upon ascending LicenseNumber. We are guaranteed that there will be a maximum of one of transaction record per TVLicense file record (that is multiple transactions will not be applied to the same master record).
There are three types of record in the transaction file:

- Deletion records - identified by the value 1 in the TypeCode field
- Insertion records - identified by the value 2 in the TypeCode field,
- DateOfLicense update records – identified by the value 3 in the TypeCode field

Deletion record

FIELD	TYPE	LENGTH	VALUES
TypeCode	9	1	1
LicenseNumber	9	7	0000001-9999999

Insertion record

FIELD	TYPE	LENGTH	VALUES
TypeCode	9	1	2
LicenseNumber	9	7	0000001-9999999
HolderName	X	25	-
HolderAddress	X	55	-
DateOfLicense	9	8	YYYYMMDD

DateOfLicense update record

FIELD	TYPE	LENGTH	VALUES
TypeCode	9	1	3
LicenseNumber	9	7	0000001-9999999

Write a program that applies the records in the transaction file to the TVLicenses file to create a new, updated, NewTVLicenses file. For DateOf License-updates the Year part of the DateOfLicense field should be incremented by 1.

Should an error be detected when trying to apply a transaction, a record containing the LicenseNumber and one of the following error messages should be displayed –

Insert error – record with that license number already exists
Delete error – no record with that license number found
DOL error – no record with that license number found

Marks for the problem solution will be allocated as follows -

1. File Section declarations (8 marks)
2. Process Transaction file and TVLicenses files ok (12 marks)
3. Apply transactions to create NewTVLicenses file ok (22 marks)
4. Detect errors and display appropriate error messages ok (8 marks)

Q2. PROGRAM DESIGN
(only attempt this question if you have <u>not attempted</u> Question 1.)

Referring to the program specification in Question 1 -

(a) Create a Program Structure Diagram representing your solution to the problem. (16 marks)

(b) Write out the executable operations required for the program. (14 marks)

(c) Write out the iteration and selection conditions required. (6 marks)

(d) Assign the executable operations and the iteration and selection conditions to the appropriate places in the Program Structure Diagram. (14 marks)

Q3. COBOL GENERAL

(a) The File Description (FD) entries shown below are for two INDEXED files where DirectorId-DF is the Primary Key and DirectorName-DF the Alternate Key of the DirectorFile and where MovieId-MF is the Primary key and MovieName-MF and DirectorID-MF are the Alternate Keys of the MovieFile. The Alternate Keys of both files allow duplicates.

Write a code fragment that accepts the Director's name from the user and then displays the names of all the films directed by that Director (12 marks)

```
FD MovieFile.
01 MovieRec.
   02 MovieID-MF          PIC 9(6).
   02 MovieName-MF        PIC X(60).
   02 DirectorID-MF       PIC 9(5).

FD DirectorFile.
01 DirectorRec.
   02 DirectorID-DF       PIC 9(5).
   02 DirectorName-DF     PIC X(50).
```

(b) Write the SELECT and ASSIGN clauses for the two files described in (a) above. (5 marks)

(c) Suppose that in addition to the files described in (a) above two additional Indexed files exist – a Link file and an Actor file. LinkID-LF is the Primary Key of the LinkFile and ActorID-LF and MovieID-LF are its Alternate Keys. ActorID-AF is the Primary key of the ActorFile and ActorName-AF is its Alternate Key. FD descriptions for these files are given below.

Write a code fragment that accepts an Actor's name from the user and then displays the names of all the movies in which that Actor appeared. (15 marks)

```
FD LinkFile.
01 LinkRec.
   02 LinkID-LF           PIC 9(7).
   02 MovieID-LF          PIC 9(6).
   02 ActorID-LF          PIC 9(5).

FD ActorFile.
01 ActorRec.
   02 ActorID-AF          PIC 9(7).
   02 ActorName-AF        PIC X(50).
```

(d) COBOL sub-programs ordinarily exhibit what is termed "State memory". Briefly describe what is meant by "state memory" and identify the COBOL language elements that allow you to create programs that do not have state memory. (6 marks)

(e) Briefly describe the purpose of the FILE STATUS clause when used in a file's SELECT and ASSIGN clause. (4 marks)

(f) The first step in decoding a simple letter substitution cipher is to count the occurrences of each substituted letter in the encoded text.

Referring to the data descriptions below, write a program fragment that uses the INSPECT to count the number of occurrences of each letter in the MessageText, and then stores the counts in the appropriate LetterCount element. (8 marks)

```
01 MessageText  PIC X(10000).

01 LetterTable "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
   02 Letter OCCURS 26 TIMES PIC X .

01 LetterCountTable.
   02 LetterCount PIC 9(5).
```