UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

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

Assessment Paper

Academic Year: 2008/2009
Module Title: Business Oriented Programming Languages
Duration of Exam: 2.5 hours
Lecturer : Michael Coughlan

Semester: 1
Module Code: CS4556
% of Total Marks: 40%
Paper marked out of : 150

Instructions to Candidates.

Four questions. Attempt **three** questions **only**.

Question 1	COBOL theory	(50 marks)
Question 2	OO-COBOL	(50 marks)
Question 3	Programming	(50 marks)
Question 4	Design	(50 marks)

Attempt three questions only

Q1 COBOL theory

(a) Briefly explain the purpose of the IS INITIAL phrase when used in the PROGRAM-ID of a subprogram, outline the advantages and disadvantages of its use, and identify the COBOL verb that can be applied to a subprogram to temporarily create the same effect. (10 marks)

(b) Using the class outline below comment on the scope, persistence, and number of instances created for each of the variables shown (Var1 .. Var5). (15 marks)

```
CLASS-ID.
WORKING-STORAGE SECTION.
01  Var1    PIC 9(5).

FACTORY.
WORKING-STORAGE SECTION.
01  Var2    PIC 9(5).
METHOD-ID.
WORKING-STORAGE SECTION.
01  Var3    PIC 9(5).
END METHOD.
END FACTORY.

OBJECT.
WORKING-STORAGE SECTION.
01  Var4    PIC 9(5).
METHOD-ID.
WORKING-STORAGE SECTION.
01  Var5    PIC 9(5).
END METHOD.
END OBJECT.
END CLASS.
```

(c) Define the terms "vertical software market" and "horizontal software market" and briefly describe the characteristic differences between the applications that operate in each of these software markets. (8 marks)

(d) What is the purpose of the FILE STATUS clause when used in a file's SELECT and ASSIGN clause? (5 marks)

(e) 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. (12 marks)

```
01 MessageText  PIC X(10000).

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

01 LetterCountTable.
   02 LetterCount Occurs 26 TIMES PIC 9(5).
```

Q2. OO-COBOL

An OO-COBOL program is required that accepts the name of a Premiership football club from the user and then displays its stadium name, the stadium capacity, and the city of its location. The problem should be solved by writing a PremiershipClub class and a main program which uses that class.

The main program should do no more than accept the club name from the user, invoke the method "new" to create the class instance, and then invoke "getStadium", "getCapacity", and "getCity" to get the required information from the class instance. The program should then display the returned values or output an error message if that is appropriate.

The methods templates are defined as shown below.

```
METHOD-ID. "new".
LINKAGE SECTION.
01 InClub          PIC X(25).
01 OpStatus        PIC 9.
* value of 0 indicates operation was successful
PROCEDURE DIVISION USING InClub RETURNING OpStatus.
*because the club name may be passed as lowercase, uppercase,
*or mixed case it should be converted to uppercase before it is used.
END METHOD "new".

METHOD-ID. "getStadium".
LINKAGE SECTION.
01 OutStadium      PIC X(25)
01 OpStatus        PIC 9.
* value of 0 indicates operation was successful
PROCEDURE DIVISION USING OutStadium RETURNING OpStatus.
END METHOD "getStadium".

METHOD-ID. "getCapacity".
LINKAGE SECTION.
01 OutCapacity     PIC 9(6)
01 OpStatus        PIC 9.
* value of 0 indicates operation was successful
PROCEDURE DIVISION USING OutCapacity RETURNING OpStatus.
END METHOD "getCapacity".

METHOD-ID. "getCity".
LINKAGE SECTION.
01 OutCity         PIC X(15)
01 OpStatus        PIC 9.
* value of 0 indicates operation was successful
PROCEDURE DIVISION USING InClub, OutCity RETURNING OpStatus.
END METHOD "getCity".
```

Because the clubs in the Premiership change from year to year the class instance should be instantiated with information obtained from the ordered sequential file "PremiershipClubs.dat". This file is in ascending ClubName order and each record contains the following items;

FIELD	TYPE	LENGTH	VALUE
ClubName	X	25	-
StadiumName	X	25	-
Capacity	9	6	1-999999
City	X	15	-

Marks for the solution will be allocated as follows -

(a)	Main Program	(14 marks)
(b)	Use method "new" to create a new instance and instantiate the instance data	(25 marks)
(c)	Methods getStadium, getCapacity, and getCity	(9 marks)
(d)	Convert club name to upper case	(2 marks)

Q3 Programming

Write a program which accepts the name of the target county from the user and then, using the Census File for reference, displays on the computer screen the ten surnames which occur most frequently in that county. Be aware that the user may enter the county name in upper case, lower case, or a mixture of cases.

The Census File is an unordered sequential file with fixed length fields. Each record contains a census number, a surname, and a county name.

Record Description

FIELD	TYPE	LENGTH	VALUE
CensusNumber	N	7	0000001-9999999
Surname	X	20	-
CountyName	X	9	AllUpperCase

The on screen report should take the format shown in the "Limerick" example below –

```
Surname Density Report for - LIMERICK

  Rank   Household      Number of
   Pos   Surname        Occurrences
    1   Fitzpatrick       112
    2   Ryan               90
    3   O'Brien            70
    4   Power              65
    5   Molloy             63
    6   Coughlan           52
    7   Burke              51
    8   Collins            45
    9   Madden             38
   10   English            32
```

The target county name is shown in the report headings. The Number of Occurrences field is the count of the number of households with this particular surname in the target county. The surname with the highest count occupies position 1, the next highest is in position 2 and so on.

Marks for the solution will be allocated as follows -

(a)	Data declarations – File & Table	(10 marks)
(b)	Sort and Select	(10 marks)
(c)	Process sorted file and accumulate Surname occurrences	(12 marks)
(d)	Get top ten surnames	(14 marks)
(e)	Display top ten surnames	(4 marks)

Q4 Design

The owner of a local radio station has decided that, in future, the station will pay royalties on all the recorded material it broadcasts. To enable the station to do this you have been asked to design a program to produce a summary file that shows the royalty payments owed to each record company. The summary file must be sequenced on ascending record company name and must only include those companies to whom royalties are owed.

Using the Program Structure Diagram and the file descriptions on the next page for inspiration, write the executable operations and the iteration and selection conditions required for the program and then create a Program Structure Diagram populated with these operations and conditions.

The summary file (RoyaltySum.Dat) is a sequential file ordered on ascending Company-Name. Each record in the file contains the following items;

FIELD	TYPE	LENGTH	VALUE
Company-Name	X	20	-------
Company-Num	9	7	0-9999999
Total-Royalties-Owed	9	8	0.00-999999.99

The summary file must be based on two indexed files. The first is the Artist/Company relationship file (Acr.Dat). This file shows which artists are under contract to which record companies. The record description for this file is as follows;

FIELD	TYPE	LENGTH	KEY	VALUE
Relationship-Num	9	7	PRIMARY	0-9999999
Artist-Num	9	7	-------	0-9999999
Artist-Name	X	20	-------	-------
Company-Num	9	7	-------	0-9999999
Company-Name	X	20	ALT WITH DUPLICATES	-------

The second file is the Opus-Royalty file (Or.Dat). This file contains the royalties owed by the station for broadcasts of each opus. Your program should reset the Royalty-Owed field to zeros after the record has been processed. The record description for this file is as follows;

FIELD	TYPE	LENGTH	KEY	VALUE
Opus-Num	9	8	PRIMARY	0-99999999
Artist-Num	9	7	ALT WITH DUPLICATES	0-99999999
Royalty-Owed	9	6	-------	0.00-9999.99

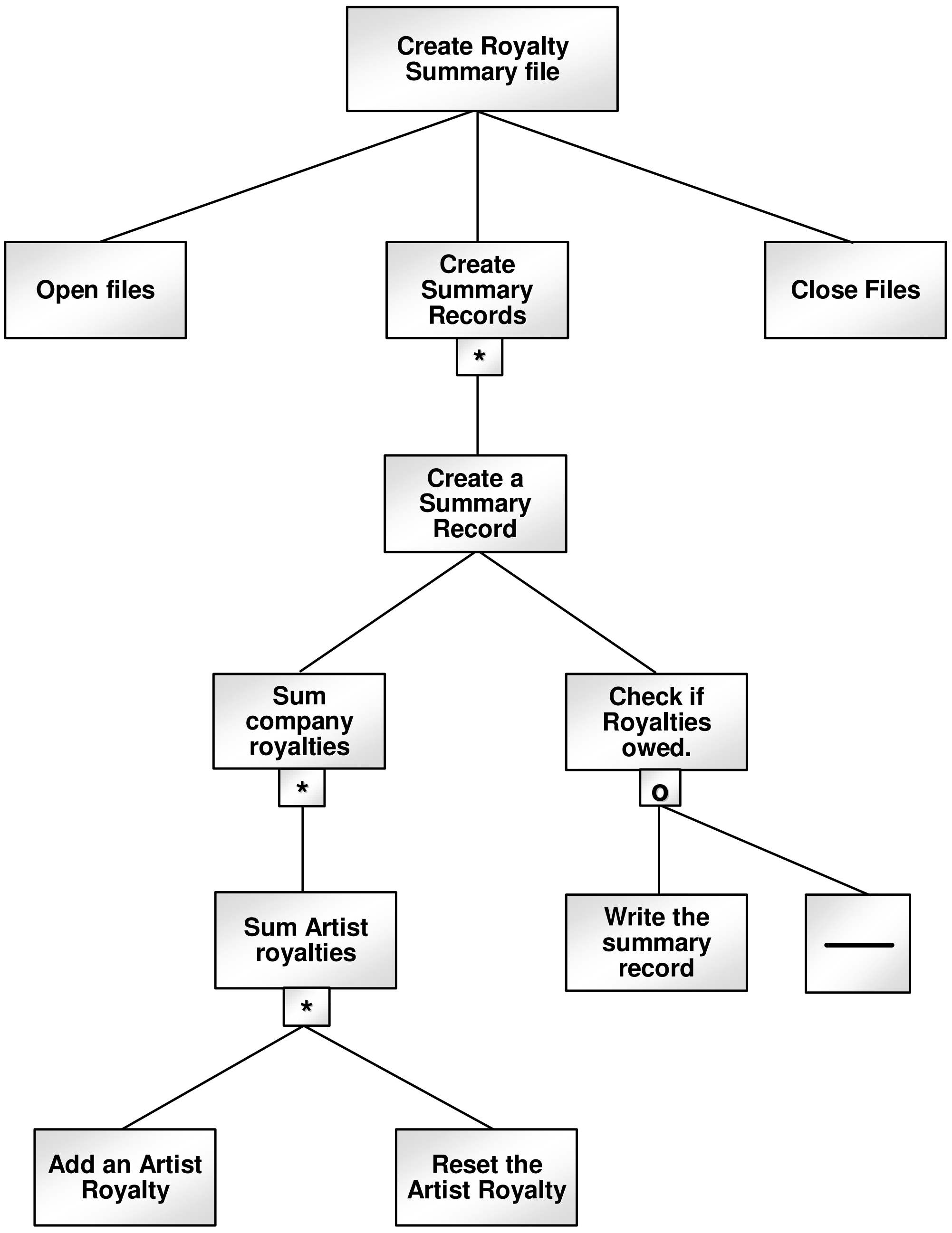
Marks for the solution will be allocated as follows –

(a)	Populated Program Structure Diagram representing your design	(16 marks)
(b)	Executable operations	(28 marks)
(c)	Iteration and Selection Conditions	(6 marks)

```
FD Artist-Company-File.
01 FA-Acr-Record.
   02  Fa-Relationship-Num     PIC 9(7).
   02  Fa-Artist-Num           PIC 9(7).
   02  Fa-Artist-Name          PIC X(20).
   02  Fa-Company-Num          PIC 9(7).
   02  Fa-Company-Name         PIC X(20).

FD Opus-Royalty-File.
01 Fb-Opus-Royalty-Rec.
   02 Fb-Opus—Num              PIC 9(8).
   02 Fb-Artist-Num            PIC 9(7).
   02 Fb-Company-Num           PIC 9(7).
   02 Fb-Royalty-Owed          PIC 9999V99.
      88 Royalty-To-Pay  Values 0.01 Thru 9999.99.

Fd Royalty-Summary-File.
01 Fc-Summary-Rec.
   02 Fc-Company-Name          PIC X(20).
   02 Fc-Company-Num           PIC 9(7).
   02 Fc-Total-Royalties-Owed  PIC 9(6)V99.
```


Create Royalty Summary file
Open files
Create Summary Records
*
Close Files
Create a Summary Record
Sum company royalties
*
Check if Royalties owed.
o
Sum Artist royalties
*
Write the summary record
—
Add an Artist Royalty
Reset the Artist Royalty