

SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY  
SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

COURSE TITLE: ADVANCED APPLICATION PROGRAMMING  
CODE NO. EDP229  
PROGRAM BUSINESS PROGRAMMER  
SEMESTER FOUR  
DATE JANUARY, 1994  
AUTHOR FRAN DEW

NEW      REVISION  X

APPROVED \_\_\_\_\_ DATE \_\_\_\_\_  
DEAN, BUSINESS AND HOSPITALITY

ADVANCED APPLICATION PROGRAMMING  
COURSE NAME

EDP229  
CODE NO.

Total credit time: 90 hours

Prerequisites: EDP112

**I PHILOSOPHY/GOALS:**

This course continues the study of structured analysis and programming techniques utilizing COBOL, begun in the previous semester. Emphasis is placed upon structured design, top-down developments, program constructs, pseudocode, structure charts, etc. These tools are applied to a variety of case studies involving file maintenance (including index sequential files) and table look-ups.

**II STUDENT PERFORMANCE OBJECTIVES:**

Upon successful completion of this course, the student will be able to:

- a Build on COBOL programming features learned in the previous semester
- b Process arrays and handle tables
- c Process indexed files and interact with the computer
- d Utilize features such as Report Writer, COPY and CALL
- e Process relative files

**III TOPICS TO BE COVERED**

- 1. Report Writer Module
- 2. Interactive Processing
- 3. Single-level Arrays and Tables
- 4. Multiple-level Arrays and Tables
- 5. Using Advanced Debugging Aids and Improving Program Performance
- 6. The COPY and CALL statements
- 7. Sequential File Processing (review)
- 8. Indexed File Processing
- 9. Relative File Processing

#### IV LEARNING ACTIVITIES

1. Report Writer Module Ch 20

Upon successful completion of this unit, the student will be able to:

- a use the Report Writer Module for printing reports

2. Interactive Processing Ch 17

Upon successful completion of this unit, the student will be able to:

- a design screen layouts so that the operator can interact with a computer at a terminal
- b handle input and output, and manipulate text

3. Single-level Arrays and Tables Ch 12

Upon successful completion of this unit, the student will be able to:

- a access and manipulate data stored in an array of table
- b establish a series of single-level items
- c use commands for table look-up

4. Multiple-level Arrays and Tables Ch 13

Upon successful completion of this unit, the student will be able to:

- a establish a series of double-level items
- b store and look up data in a double-level table or array
- c access and manipulate data defined with a triple level

5. Using Advanced Debugging Aids and Improving Program Performance Ch 18

Upon successful completion of this unit, the student will be able to:

- a use advanced techniques to debug programs
- b improve program efficiency

ADVANCED APPLICATION PROGRAMMING  
COURSE NAME

EDP229  
CODE NO.

6. The COPY and CALL statements Ch 19

Upon successful completion of this unit, the student will be able to:

- a copy standard parts of a program from a library
- b execute subroutines

7. Sequential File Processing (review) Ch 14

Upon successful completion of this unit, the student will be able to:

- a explain master file processing concepts
- b update sequential files using disk as a master file
- c update sequential disk files in place with a REWRITE statement

8. Indexed File Processing Ch 16

Upon successful completion of this unit, the student will be able to:

- a randomly process disk files
- b create and access indexed disk files

9. Relative File Processing Ch 21

Upon successful completion of this unit, the student will be able to:

- a create, update and use relative files for reporting
- b organize relative files

ADVANCED APPLICATION PROGRAMMING  
COURSE NAME

EDP229  
CODE NO.

**V EVALUATION METHODS**

Tests (3 @ 20%)	60%
Assignments	20%
Project	15%
Participation	5%
	----
	100%

Grading: A+ 90 and over  
A 80 and over  
B 70 and over  
C 60 and over  
R under 60

**VI REQUIRED STUDENT RESOURCES**

Text: "Structured COBOL Programming" 6th Edition  
by Stern and Stern  
1991

available in the Campus Shop

**VII SPECIAL NOTES**

Assignments received after the due date are subject to a zero grade, unless the student has prior permission from the instructor to hand the assignment in at a later date.

Students with special needs, such as physical limitations, visual impairments, hearing impairments, or learning disabilities, are encouraged to discuss required accommodations, confidentially, with the instructor.

Your instructor reserves the right to modify the course as is deemed necessary to meet the needs of students.

EDP229 ADVANCED APPLICATIONS PROGRAMMING

ASSIGNMENT ONE BANK ACCOUNT BALANCE

At the SCAAT CREDIT UNION, a group-indicated BANK ACCOUNT BALANCE report is to be created from an account data file. Individual transactions are to be printed, and a balance for each account is to be accumulated and printed. A final over-all total for balances is to be output at the end of the report. Records in error are to be listed in a BANK ACCOUNT BALANCE DATA ERROR report, with the error listed beside the record.

Data validation is to be performed on all numeric fields.

Sort the records in ascending order by date within account number. "Group-indicated" means that the account number will be printed on the first line of each account output, but not on the succeeding lines for that account.

USE REPORT WRITER TO CREATE THE REPORT

Limit the width of the report to 80 characters or less.

DATA

Have a minimum of 15 accounts, with each account starting with an old balance (code 1). Make sure that you print out at least two pages of report. Each account is also to have at least two other transactions. The layout of the input record is as follows:

ACCOUNT NUMBER	5 digits
TRANSACTION DATE	6 digits in the form DDMMYY
AMOUNT	8 digits (two decimal places)
TRANSACTION CODE	1 digit

where transaction code is: 1=old balance  
2=deposit  
3=withdrawal

Use the logic from a COBOL I assignment to assist you.

EDP229 ADVANCED APPLICATION PROGRAMMING

ASSIGNMENT TWO ON-LINE MENU

The SCAAT Boutique is becoming automated. The initial stage includes setting up a standard employee data file. A data entry clerk has been hired to transfer pertinent information from paper files to a computer file.

Write a program to produce an on-line menu which prompts the data entry clerk to input each employee name, employee number, department number, year of employment, job code and annual salary.

Create a data file, sorted in ascending order of employee number. Print out a seniority list of employees, sorted in ascending order of employee number within year of employment. Print a report listing the average annual salary for each job code.

Access to the program is protected by a password entry. After a third unsuccessful attempt to enter the correct password, the user is denied further access to the program.

Put in editing parameters, such as accepting only certain department numbers and job codes. Consider other delimiters that would logically be in a program such as this.

DATA

Have a minimum of 15 employees. The layout of the output record is as follows:

Employee first name	10 characters
Employee last name	10 characters
Employee number	5 digits (00001 through 00050)
Department number	2 digits (Departments 01 through 12)
Year of employment	2 digits (last two digits of year)
Job code	3 digits (codes 001 through 050)
Annual salary	6 digits (no decimal places)

EDP229 ADVANCED APPLICATION PROGRAMMING

ASSIGNMENT THREE

STUDENT-MAJOR

SCAAT College registrar's office requires a STUDENT - MAJOR report on students, their schools, majors and GPAs (grade point averages).

Print the STUDENT-MAJOR report by student first name within student last name within major within school. The SCHOOL CODES and MAJOR CODES of this assignment are laid out in program assignment 7, page 461 of Chapter 12.

DATA

Use at least 40 student records. For the report, print 20 students per page, and within an 80 column width. The layout of the input records is as follows:

Student number	8 digits
Student first name	10 characters
Student last name	15 characters
School code	1 digit
Major code	3 characters
GPA	3 digits (two decimal places)

EDP229    ADVANCED APPLICATION PROGRAMMING

ASSIGNMENT FOUR

WEEKLY PAYROLL

The SCAAT Hotel wants a computer system to take care of its accounting needs. A portion of this system produces the GROSS WEEKLY PAY report for the employees. Pay rates have been stored in a table file, PAYROLL-RATES.DAT. The values in the table are loaded through a subroutine PAYROLL-RATES.COB. A SCAAT-PAYROLL.COB program calls PAYROLL-RATES.COB as a subroutine, and then uses the table values in the calculation of employee wages. A job description table file, JOB-DESCRIPTION.DAT, consists of job codes and descriptions.

The systems analyst is responsible for setting up the payroll rates table, the logic of the program to load the table, and the layout of the error report and payroll report, as well as the logic for the payroll program. The programmer is responsible for the actual coding, debugging and running of the programs.

Because SCAAT Hotel is relatively small, one person is both the Systems Analyst and the Programmer. Congratulations, you have just been hired to do this job.

There are three departments: Housekeeping (code 1), Maintenance (code 2) and Accounting/Administration (code 3). Each department wants a printout for its employees. Accounting/Administration also wants a printout of all departments, as well as over-all totals.

The reports are to include as much information as is necessary. All of the information is to be saved on a file, for future reference. Information on each employee is to be entered by an data entry clerk, with helpful screens to accomplish the data entry correctly. Overtime is calculated over 40 hours, at time and a half.

DATA

The layout of the output record is as follows:

Employee first name	10 characters
Employee last name	10 characters
Employee Number	5 digits (00001 to 00050)
Department Code	1 digit (departments 1 through 3)
Job Code	3 digits (codes 001 through 050)
Time Worked	4 digits (HHMM)

Test out the system using your own data. Set up the .COM file so that I can use my own PAYROLL-RATES.DAT, JOB-DESCRIPTION.DAT and employee files.

EDP229 ADVANCED APPLICATION PROGRAMMING

TERM PROJECT BANKING SYSTEM

The SCAAT Bank of Canada is a newly established bank in Sault Ste. Marie, Ontario. It is a fast-growing company, with many Sault College graduates as employees, and many small businesses and students as part of its customer base. The Bank does a brisk business, processing loans for both entrepreneurial and conservative companies. Customers want their loan requests and transactions processed in a timely fashion. They also want the Bank to be able to tell them their account balances and loan balances quickly.

The Bank needs to maintain online interactive functions concerning customer accounts and loans. Accounts can be added, updated and deleted. Loans can be added, updated and deleted. Customers and bank personnel will want to see a display of account transactions and balances, as well as loan payments and balances.

Your firm has been hired to set up the system, as well as complete the necessary program. Go through the analysis with the end-user. Find out what reports are required and when. Find out what files are to be updated.

In order to help the end-users actually use your system, set up a USER'S GUIDE explaining all functions to the user. This will guide the user through the system easily. Screen layouts and error handling procedures are to be explained.

Yes, I am the end-user.

Class time will be used for questions from the groups. Make sure that you write down all the requests and answers from the end-user. Keep in mind that you may have to lead the end-user to what should be needed.

## DIRECTIVES

A customer can be issued a loan, without having an account at the bank.

Loan payments can be made either through an account or by cash or by cheque drawn on another bank.

Loan payments must be equal to or greater than the predetermined monthly payment.

If a customer does not have overdraft allowance, then the account balance cannot fall below zero.

If the customer does have overdraft allowance, then the account balance cannot fall below the overdraft amount.

## BASIC REQUIRED FUNCTIONS

### ACCOUNTS

addition

update based on deposit or withdrawal transactions and/or payments made on loans through an account

deletion

### LOANS

addition

update based on payments

deletion based on a zero balance for the loan

### LOGS

transaction account entries for deposits, withdrawals, loan payments

payment loan payment entries

## REPORTS

### TRANSACTION report

In ascending order of account number, list all transactions made for each account, with opening balance, deposits and withdrawals, as well as closing balance. Print the account number once. Display all account totals. As well, print over-all Bank totals.

### LOAN-PAYMENT report

In ascending order of loan number, list all payments made on each loan, with initial principle, payment amounts and dates, total payment amount, and outstanding amount. As well, print over-all loan totals.

## SPECIFICATIONS:

### MASTER FILES

#### ACCOUNT

account number	5 digits (first digit signifies either checking or savings account)
customer last name	10 characters
customer first name	10 characters
account balance	7 digits (two decimal places, there can be a negative balance)
date account opened	6 digits (YYMMDD)
overdraft allowance	4 digits (maximum \$1,000)

#### LOANS

loan number	5 digits
customer last name	10 characters
customer first name	10 characters
initial loan amount	6 digits (no decimal places)
term of loan	3 digits (in number of months)
monthly payment amount	6 digits (2 decimal places)
payment due day of month	2 digits
interest rate	4 digits (2 decimal places)
loan balance due	8 digits (2 decimal places)
date loan issued	6 digits (YYMMDD)

### TRANSACTION FILES

The following files will be maintained to keep track of the deposits and withdrawals made against accounts, and payments made on loans. The files will help settle any customer-bank discrepancies and provide an audit trail for accounting and reporting purposes.

#### TRANSACTIONS

account number	5 digits (first digit signifies either checking or savings account)
transaction date	6 digits (YYMMDD)
transaction type	1 character (D=deposit W=withdrawal)
transaction amount	7 digits (two decimal places)

#### PAYMENTS

loan number	5 digits
actual payment day	6 digits (YYMMDD)
actual payment amount	6 digits (two decimal places)