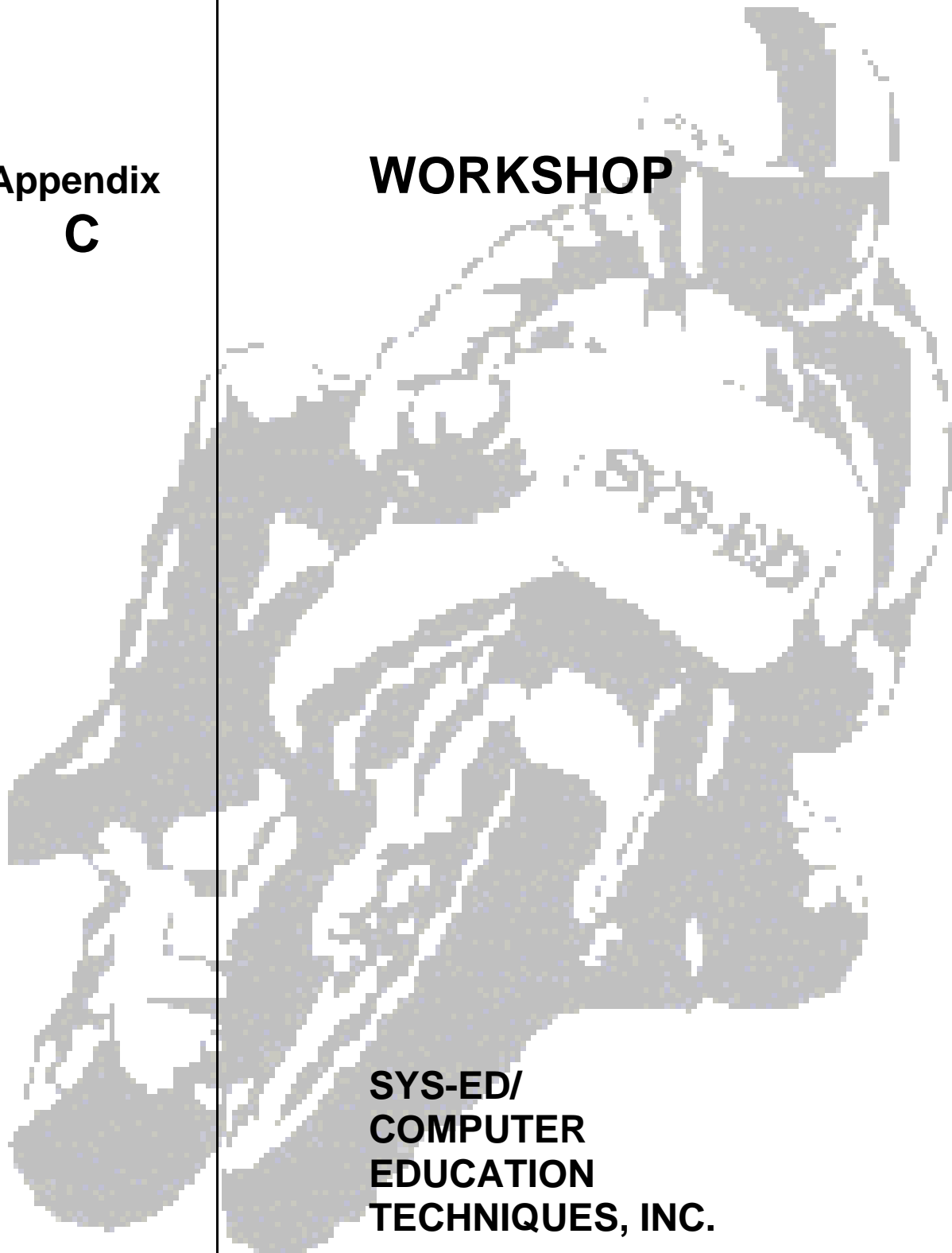


**Appendix
C**

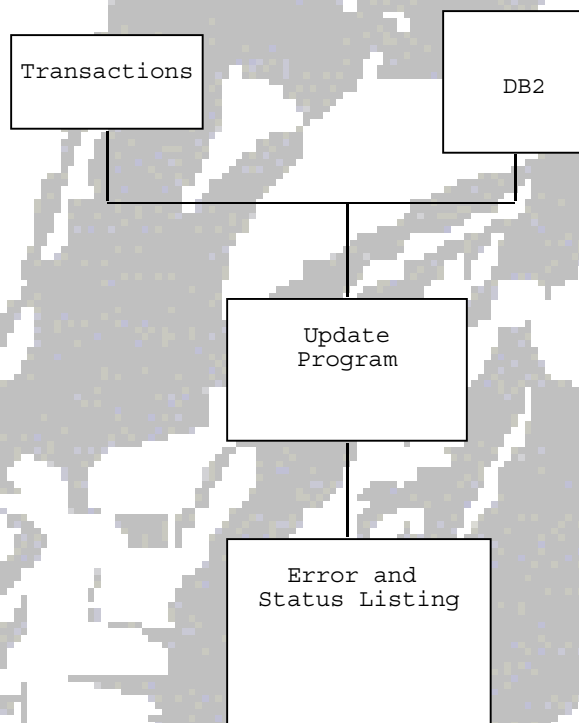
WORKSHOP



**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Workshop 1**Objectives:**

- C Execute DCLGEN to generate the SQL DECLARE TABLE and host variables for a COBOL program. The DB2I Interactive utilities will be used to perform this function.
- C Perform the process of precompiling, compiling, linking, and binding a COBOL/DB2 application program. This can be done with the DB2I Preparation panels, supplied JCL or Endeavor.
- C Code the logic to handle nulls in a COBOL program.
- C Code the 4 DML statements in a COBOL program: SELECT, INSERT, UPDATE and DELETE.
- C Code for error handling conditions.

Program Flowchart

Preparation

Every installation has a different method for coding and testing programs.

The following may be different in your shop.

- C Create a PDS to store your COBOL program with a LRECL of 80 and a RECFM of FB.
DSN: uid.DB2.COBOL.
- C Create a PDS to store your Load Modules with a BLKSIZE of 32767 and a RECFM of U.
DSN: uid.DB2.LOAD.
- C Create a PDS to store your DBRM with a LRECL of 80 and a RECFM of FB.
DSN: uid.DB2.DBRM.

The DB2 subsystem that we will be using in this course is _____.

Program Description

- C The program reads a sequential dataset that contains transactions to update or read the database.
- C The input file is a sequential dataset with a LRECL of 150.
- C The record layout of this dataset is:

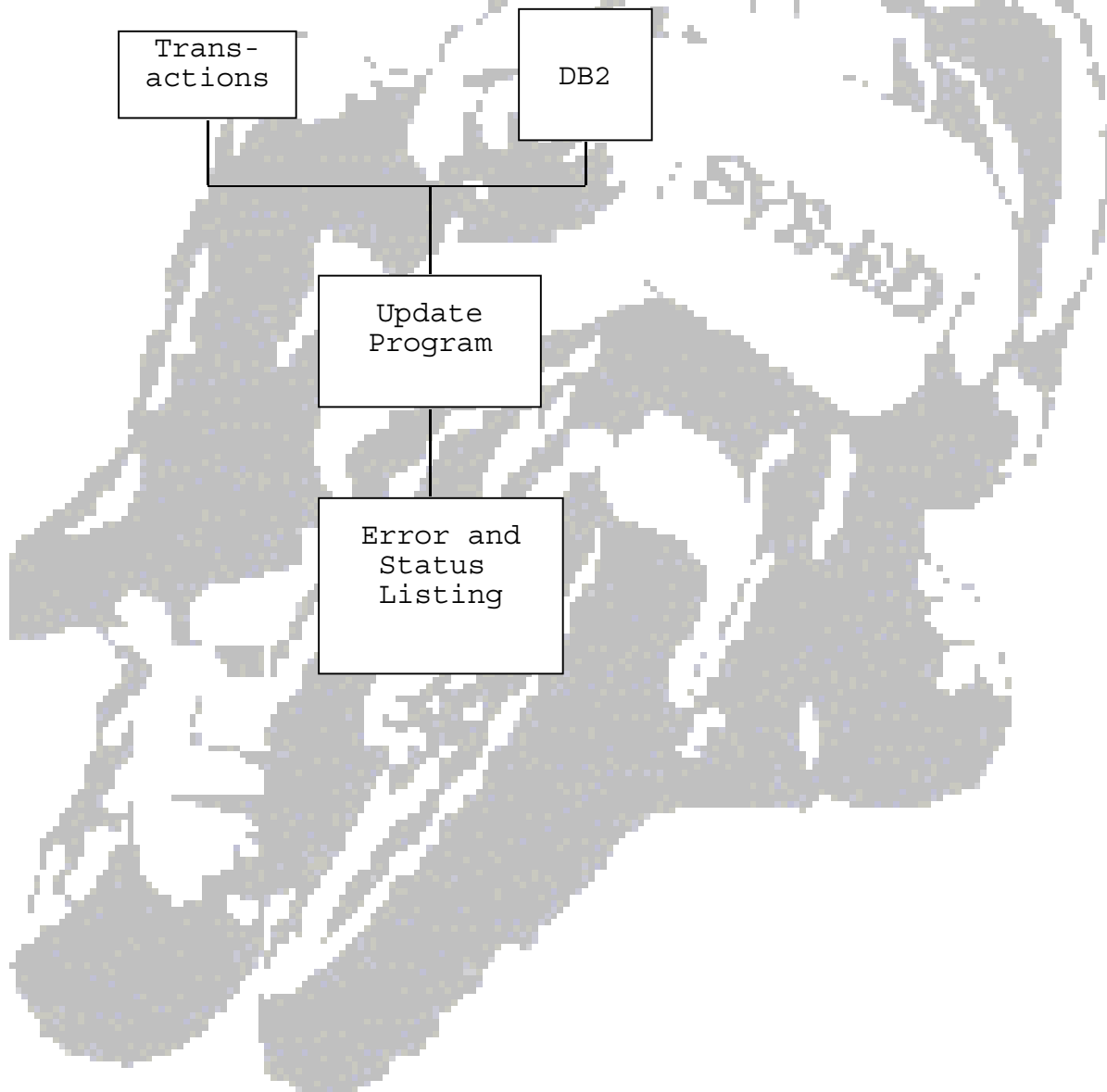
TRANSACTION CODE	1	Add,Delete,Change,Inquiry
EMPLOYEE ID	9	SSN
EMPLOYEE NAME	25	NAME OF EMPLOYEE (Blank for Inquiry)
DEPARTMENT	2	DEPARTMENT NUMBER (Blank for Inquiry)
SALARY	9	9 DIGIT SALARY S9(7)V99 (Blank for Inquiry)
- C All display output should be sent to SYSOUT with a DISPLAY statement. This output would contain inquiry listings, status and error (SQLCODE) codes.
- C For any negative SQLCODEs that are generated, a meaningful report should be displayed. This includes the SQLCODE, pertinent SQLCA data and a call to DSNTIAR.
- C For each of the TRANSACTION CODES in the input file, code a COBOL routine to perform the function. For the inquiry, display the information to SYSOUT.
- C For the update, only update non blank fields on the transaction record. Note, that the key (SSN) cannot be updated.

Workshop 2

Objectives:

- C Code and debug a program that uses cursor access.
- C Perform for end of data processing.
- C Code for error handling conditions.

Program Flowchart



Program Description

C The program reads a sequential dataset that contains transactions to update or read the database.

C The input file is a sequential dataset with a LRECL of 150.

The record layout of this dataset is:

TRANSACTION CODE	1	Add,Delete,Change,Inquiry
EMPLOYEE ID	9	SSN
EMPLOYEE NAME	25	NAME OF EMPLOYEE (Blank for Inquiry)
DEPARTMENT	2	DEPARTMENT NUMBER (Non-Blank for Inquiry)
SALARY	9	9 DIGIT SALARY S9(7)V99 (Blank for Inquiry)

C All display output should be sent to SYSOUT with a DISPLAY statement. This output would contain inquiry listings, status and error (SQLCODE) codes.

C For any negative SQLCODEs that are generated, a meaningful report should be displayed. This includes the SQLCODE, pertinent SQLCA data and a call to DSNTIAR.

C For each of the TRANSACTION CODES in the input file, code a COBOL routine to perform the function. For the inquiry, display the information to SYSOUT.

C For the update, only update non blank fields on the transaction record. Note, that the key (SSN) cannot be updated.

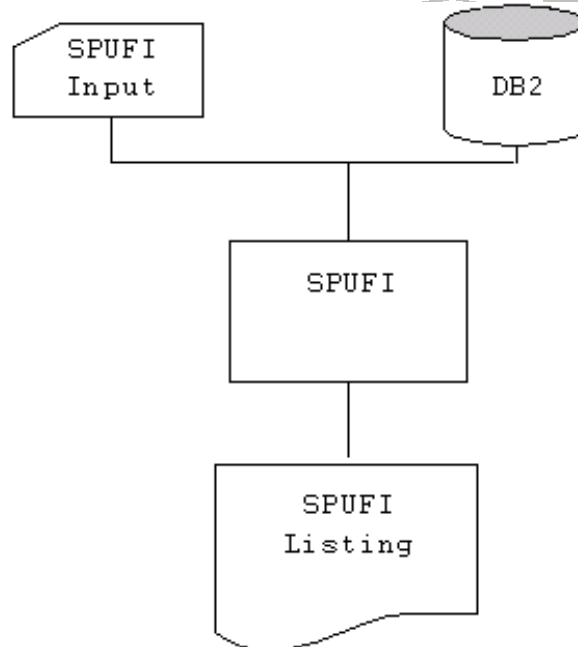
C For the inquiry, check the value in DEPARTMENT.

C For a blank DEPARTMENT, display the single employee using the EMPLOYEE ID as the key (previous workshop).

C For a valid DEPARTMENT, display all employees in the requested DEPARTMENT.

Workshop 3**Objectives:**

Use SPUFI to develop SQL that uses an inner join, full outer join, left outer join, and right outer join.

Program Flowchart**Program Description**

Using the tables, DSN8610.EMP and DSN8610.DEPT, code and test the following SQL statements:

- C Regular inner join using DEPTNO and WORKDEPT as the keys.
- C Using the keyword INNER JOIN, code an inner join using DEPTNO and WORKDEPT as the keys.
- C Full outer join using DEPTNO and WORKDEPT as the keys.
- C Left outer join using DEPTNO and WORKDEPT as the keys.
- C Right outer join using DEPTNO and WORKDEPT as the keys.

Workshop 4

Objectives:

- C Code, install and test a user-defined function.
- C This lab can either use COBOL or C.

Program Description

- C Write a user defined function to locate the last blank in a string.
- C The function is to return the position of the blank.
- C Issue the CREATE FUNCTION to store that functions definition in the database.
- C Code an SQL statement to use and test the user defined function.

Workshop 5**Objectives:**

- C Code a stored procedure program.
- C Define the stored procedure to DB2.
- C Invoke the stored procedure from a COBOL program.

Parameters

Department

Return

Result Set

Program Description

- C Code a stored procedure to retrieve all rows in DSN8610.EMP that has a DEPTNO equal to the passed department.
- C Store the result table to be returned to the invoking program.
- C Issue the CREATE PROCEDURE to store the definition of the stored procedure in DB2.
- C Test the stored procedure by invoking it from a COBOL program. Display the returned result set.

Workshop 6**Objectives:**

- C Code a trigger.
- C Define the trigger to DB2.
- C Invoke the trigger from a SPUFI program.

Program Description

Create the following two tables with the following columns:

EMPTBL
EMPLOYEE
EMPLOYEE NAME

AUDITTBL
USER
INSERTDATE
INSERTTIME

- C Code a trigger that is invoked every time a row is inserted to EMPTBL.

When a new record is added to EMPTBL, the name of the user, insert date, and insert time are added to AUDITTBL in a new record.
- C Issue the CREATE TRIGGER to store the definition of the trigger in DB2.
- C Test the trigger by invoking it from a SPUFI program.

Display the AUDITTBL to confirm the triggers execution.