An interactive information system for tracking student academic progress and aid in course scheduling

1985

Farahzad Behi

University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation

Behi, Farahzad, "An interactive information system for tracking student academic progress and aid in course scheduling" (1985). Retrospective Theses and Dissertations. 4816.

https://stars.library.ucf.edu/rtd/4816

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
AN INTERACTIVE INFORMATION SYSTEM FOR TRACKING
STUDENT ACADEMIC PROGRESS AND
AID IN COURSE SCHEDULING

BY

FARAHZAD BEHI
B.S., North Carolina State University, 1979

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Summer Term
1985
ABSTRACT

An interactive information system has been developed for the Computer Engineering Department to promote faster and more accurate means of obtaining academic information on a student or a group of students. It was also developed to provide departmental administrative personnel with information to guide course scheduling. The system has the capability of providing the following information concerning undergraduate students enrolled in the Computer Engineering program of the College of Engineering, University of Central Florida.

1. Basic information about the student, such as, address, telephone number, classification, etc.

2. A student's completed and in-progress courses

3. The grade and other information, such as, equivalent transfer credits or UCF resident credits for each course

4. Name and social security number of all the students working under the same advisor

5. Name and social security number of all the students in the same year of college (classification)

6. Name and social security number of all students that have taken or are taking a certain course

7. Name and number of all students with same classification taking the same course

The system provides the user with the additional capabilities to:
1. Update the information in the student's record by interactive operation

2. Delete a student from the data base

3. Print the sorted name and social security numbers of all the students that have a record in the data base
ACKNOWLEDGEMENTS

The author wishes to thank the members of his supervisory committee: Dr. John E. Biegel and Dr. Gary E. Whitehouse for their assistance in the preparation of this research report.

The author is particularly grateful to his advisor and committee chairman, Dr. Christian S. Bauer, whose friendship and inspirational guidance gave encouragement throughout the class work and the preparation of this research report.
# TABLE OF CONTENTS

**LIST OF FIGURES** .................................................. vii

**Chapter**

I. INTRODUCTION ...................................................... 1

II. DATA BASE MANAGER INFORMATION SYSTEMS AND THEIR RELATION TO STUDENTS' ACADEMIC DATA BANK - A LITERATURE SURVEY ................................................. 4

III. DATA BASE MANAGER PACKAGE USED .......................... 10
dBASE III ............................................................ 10

IV. SYSTEM ANALYSIS .................................................. 13
 Data Base Required .................................................. 13
 Student Record - Structure and Contents ....................... 14

V. PROGRAM DESCRIPTIONS AND FUNCTIONS ....................... 17

  MENU.PRG Program .................................................. 17
  ALLPRO.PRG ........................................................ 19
  Procedure "INITVAR" ................................................. 19
  Procedure "ADDSSTU" ............................................... 20
  Procedure "GETDATA" ................................................. 20
  Procedure "FINDSTU" ............................................... 22
  Procedure "DELSSTU" ............................................... 24
  Procedure "OUTCHOIC" ............................................. 24
  Procedure "FGCORS" ............................................... 27
  Procedure "FGADVI" ............................................... 27
  Procedure "FGCLASS" .............................................. 30
  Procedure "FGCORCLS" ............................................ 30
  Procedure "SORTALST" ............................................ 31

VI. SYSTEM INTEGRITY AND SECURITY ............................... 32

VII. CONCLUSIONS AND RECOMMENDATIONS ........................... 34
Appendices
A. SYSTEM STRUCTURE DIAGRAM .......................... 38
B. LISTING OF PROGRAMS ................................. 40
C. OUTPUT SAMPLES ........................................ 72
D. BACKUP OF THE DATABASE/RECOVERY OF THE DATABASE .... 77
E. GENERAL USER INSTRUCTIONS .......................... 79
REFERENCES .................................................. 82
LIST OF FIGURES

1. Data Base Manager .............................................. 5
2. Structure for Data Base "MASTER" .............................. 15
3. Course Field Example ........................................... 16
4. Main Menu ............................................................. 18
5. Flowchart for the "INITVAR" Procedure ....................... 19
6. Flowchart for the "ADDSTU" Procedure ....................... 21
7. Flowchart for the "GETDATA" Procedure .................... 23
8. Flowchart for the "FINDSTU" Procedure ..................... 25
9. Flowchart for the "DELSTU" Procedure ....................... 26
10. Flowchart for the "OUTCHOIC" Procedure .................. 28
11. Flowchart for the "FGCORS" Procedure ...................... 29
12. Flowchart for the "SORTALST" Procedure .................. 31
CHAPTER I

INTRODUCTION

In order for a student to graduate from the University of Central Florida, the student must satisfy certain requirements. The basis of these requirements entails completing a prescribed set of courses which leads to proficiency in the student's major. This curriculum is a product of university requirements, college requirements and departmental requirements, thus barring any complications such as switching majors, failing courses, etc.

An engineering student must complete approximately fifty university courses in order to receive a bachelor's degree.

To help the student decide which courses to take, pre-registration trial and advisement interviews are set up between university advisors and students in their respective departments. Ideally, the department has a complete and up-to-date record on the student containing all the courses completed or in-progress, plus the grades and other information on those courses, such as transfer equivalent credits, grades, electives and number of credit hours. However, this is not always the case. There are occasions when the student's record is not complete or cannot be found at the time of advisement. Furthermore, other information on a student such as address, telephone number, advisor's name and classification should be available at the time of need. Currently, the process involves
obtaining information on the personal and academic record of the student is done by getting the paper file of the student from the department's file cabinet. Also, obtaining information on a group of students, such as, the number of students that have taken a particular course, the number of students that are in the same year in college, the number and the names of students that have the same advisor, and a combination of these criteria are very time-consuming tasks when using information filed on paper. These are items of information that aid the faculty in the course scheduling for later semesters.

In order to have a more accurate record on a student and to accelerate the process of information search, an interactive information system has been developed. Since it is not within the scope of a single master's research report to develop an information system for the entire university, the target of this project is the department of Computer Engineering undergraduate degree program. However, the expansion of this system to include graduate degree students has been considered during the development of the system.

Administrative personnel in the department of Computer Engineering defined the following basic requirements of such an advisement information system:

1. User-oriented: requires little previous knowledge of computers
2. Fast access and easy to use system
3. Capable of updating information concerning students and their academic record: must facilitate easy addition, deletion, change or simply examination of any record of information

4. Well-documented for both users and future programming expansion and changes

5. Capable of printing a report on the acquired information

This information system will allow secretaries and faculty to obtain information on students faster and easier. It also aids the faculty in scheduling courses for later semesters. And, although this system will require the secretaries to update and maintain the students' records, the time saved by using the system to obtain such information will more than justify the investment in establishing and maintaining the system.
A data base management system is essentially "a group or a collection of programs that connect the user to one or more collections of information" (Townsend 1984). The collection of information, or pool, is called a data base (see Figure 1). Program applications for data base management systems include general ledgers, inventory control, accounts receivable, inquiry and mailing list systems, cataloging, order entry systems, bibliographies, and essentially any application in which data files are built, updated, analyzed and reported. The student tracking system is such an application.

The process of advising students on the basis of courses completed versus courses required is an element of the functional type of decision known as operational control. Operational control is defined as "the process of assuring that specific tasks are carried out effectively and efficiently" (Anthony 1970). Ideally, operational control decision-making is supported by the same data bank which supports other functional types of decision-making in the university, including administrative control, management control and strategic planning (Hussain 1973). In other words, an advisement information system should have access to the university files which contain student information and course information, i.e., the
Figure 1. The Data Base Manager.
registrar's files. However, the programs in this research report were developed for a microcomputer system (IBM personal computer) and such an access was not possible. Even if the programs were developed for the school's mainframe system (IBM 4341) or other large systems, access to the registrar's files was not possible because of security purposes.

"Student Graduation Progress Report," by Daryl Monroe (1968), is an example of an information system developed for the purpose of obtaining graduation checks which accesses the university's records. The content of student records and of other related files are somewhat related to the contents of the data base developed for the information system discussed in this research report, keeping in mind the difference between the computer systems used.

In Monroe's article, the detailed information available for each student analyzed was described as:

1. Student's name and number
2. Faculty advisor, major degree, concentration, degree plan
3. Course description, course number, grade, credit hours and grade point for all required courses enrolled in to date
4. A course description, course number, grade and credit hours for all required courses necessary for this degree plan, but not enrolled in to date
5. Summary information for all required courses including total hours attempted, total grade points earned and a grade point average in the required course area
6. Summary information in the areas referred to as general requirements (CORE), and all the details for the courses in this specific area.

7. A summary of all elective courses taken is provided, including total hours attempted, total quality points and a grade point average in the electives only.

8. A summary of all courses taken which count toward graduation, not including non-credit courses; this summary includes all the details considered on the summary of other areas.

9. The status of orientation is normally indicated, either with an "S" for satisfactory, a "U" for unsatisfactory, or as "not taken for credit".

10. The total hours required for graduation of a specific student are indicated.

11. A total for the number of hours of failure is provided.

12. A total for the number of hours of duplication of courses is provided.

This student advisement information system was written for Mesa College in the SPS language for the IBM 1620 computer. The system is reportedly effective in the area of student advisement because of its accuracy, and because it saves large amounts of staff time.

Other similar systems were developed for schools which provide detailed information about the student and courses.

A student information system by Montgomery (1972), which is considered to be the predecessor of the aforementioned system, attempted to satisfy the requirement parallel to the system developed for Mesa College using a single FORTRAN program, run in batch mode.

Kaimann (1973), in Structural Information Files, described a proposed structure for a data base which contained information about
students, teachers, courses and schools (this effort was mainly aimed at secondary schools). In Kaimann's system, student records contained the following information (1973):

1. The check character, identifying the record type
2. The unique number assigned to each student
3. Demographic and other information
4. Curriculum information for courses
   a. the course home address
   b. the grades that the student received and the date that the course was taken
   c. the front tie relating to the succeeding student enrolled in the course
5. Data overflow address
6. Category linkage
7. Synonym linkage

The "home address" of the course would have been initially calculated from a unique number assigned to the course. Likewise, records for students, teachers and schools have home addresses initially computed from a unique number assigned to each student, teacher and school, respectively.

C.J. Date's *An Introduction to Data Base Systems* (1975), presents an example concerning an information system for internal education systems of a large industrial company. Date's example uses IMS (Information Management System/360), an IBM program product and DL/I (Data Language One), an IMS data sub-language to manipulate
the information in the data base described. Details of the information contained in the data base are very similar to other data base systems described previously.

The system developed in this research report provides similar information on the students and courses. But, it does not provide as much detail as some of the larger systems discussed before.

Because of the obvious application of the data base management packages in the market, the system developed in this research report uses dBASE III data base manager for the implementation of the student's academic progress tracking system. The package was purchased by the University of Central Florida-Brevard Campus.
CHAPTER III
DATA BASE MANAGER PACKAGE USED

**dBASE III**

Before writing the programs for the student tracking information system, consideration was given to the programming language that should be used. Because of the applicability of the data base management software packages to this task, the dBASE III data base management software package was used. dBASE III is one of the most sophisticated data base managers available today for microcomputers such as the IBM personal computer. dBASE III is a descendent of dBASE II but has expanded capabilities.

Since several files are generally used in a given application and the relationship between information in different files is not stored in the system, dBASE III is not a true data base management system in the strictest sense of the word. dBASE III is more like a file management system with relational features added. dBASE III does contain its own programming language, permitting a user to develop extremely powerful and complex programs that meet demanding applications like general ledger, job costing and inventory control.

Some of the features of dBASE III include:

1. A large degree of program and data independence. Data structures can be altered without the necessity of many program changes

2. Data can easily be added, edited, deleted, sorted, indexed or reported using a minimum of programming
3. Reports can easily be created from data in a data base using mathematical operations like multiplication and division; subtotals and totals can be easily generated.

4. Professional screen formats can be created for data entry; you design the entire screen, and data entry only requires the operator to "fill in the blanks".

5. The internal language is extremely powerful with single commands for indexing, sorting and reporting; often programs can be developed at five or ten times the speed of developing equivalent BASIC programs.

6. The internal language is a structured language; structured programs are far easier to write and update than unstructured programs.

dBase III limitations are as follows:

1. Maximum number of records in a data base: 1 billion
2. Maximum number of characters in a data base: 2 billion
3. Maximum size of a data base record: 4,000 bytes
4. Maximum size of a memo record: 512K characters
5. Maximum number of fields in a data base: 128
6. Maximum size of a character string: 254 characters
7. Size of all date variables and fields: 8 bytes
8. Size of all logical variables and fields: 1 byte
9. Maximum size of a memo field: 4,096 characters
10. Maximum size of a number: 19 bytes
11. Maximum number of open files of all types: 15
12. Maximum number of open data base files: 10
13. Maximum number of open index files for each open data base file: 7
14. Maximum number of open format files for each open data base file: 1
15. Numeric accuracy: 15.9 digits
16. Largest number: $10^{\text{308}}$
17. Smallest number: $10^{-307}$
18. Maximum number of active memory variables: 256
19. Maximum size of memory variable area: 6,000

The system requirements for the dBASE III are as follows: In order for dBASE III to run on the 16-bit microcomputer (IBM PC), the minimum of 256K bytes of main memory, an operating system equivalent to the MS/PC DOS version 2.0 or later, and two 360K floppy disk drives (double-sided) or one 360K floppy disk drive and a hard disk drive are required.
CHAPTER IV

SYSTEM ANALYSIS

The student academic tracking system uses a large data base file and the combination of two program files to perform its task. These programs are written in dBASE III syntax. The program called "MENU.PRG" is the menu driver of the system. It displays a menu and calls the procedures from the other program, "ALLPRO.PRG," to execute the chosen tasks. The "ALLPRO.PRG" program contains the number of procedures which can be called when the procedure file (ALLPRO.PRG) is open. The "MENU.PRG" and "ALLPRO.PRG" programs will be discussed in later chapters. Appendix A shows an overview of the system structure.

Data Base Required

Throughout the remaining discussion about the student tracking information system programs, the word "DATA BASE" should be understood as meaning an ordered set of logically similar records, each record having exactly the same structure. There is only one data base which is accessed by the student tracking information system programs, although other index files on the unique information in this data base are used for faster access to the records. The name of the data base file is "MASTER." It contains the number of records, each record pertaining to a particular student, but with all records possessing exactly the same structure.
Student Record - Structure and Contents

Data base files were created through the use of the CREATE command. The name, type and size of each field was structured as shown in Figure 2. There are seventy-two fields for each record. Data elements of every field are defined as character strings (type) and the maximum size of each field (width) varies depending on the information it contains. There are forty-nine courses, each assigned to a field, plus three extra fields for other courses that the student might take. Two 70-character fields are reserved for notes. Another eighteen fields were assigned to contain information such as name, social security number, address, etc. The numbers under the width column of Figure 2 reveal the number of character strings reserved for the data element under the field name of the corresponding record. The field reserved for each course contains ten character strings. Possible contents of each position in the course field with their corresponding interpretations are (see Figure 3a):

First Character: credit hours of the course
Second Character: blank
Third Character: "T" for transfer or "U" for UCF
Fourth Character: blank
Fifth Character: "R" for required or "E" for elective
Sixth Character: blank
Seventh and Eighth Characters: "CO" for completed, "DR" for dropped, "IN" for in-progress and "NE" for next semester
Ninth Character: blank
Tenth Character: grade after completion
<table>
<thead>
<tr>
<th>Field</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Field</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SSNO</td>
<td>Character</td>
<td>11</td>
<td>37</td>
<td>EGN 3311</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>LNAME</td>
<td>&quot;</td>
<td>20</td>
<td>38</td>
<td>EGN 3321</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>FNAME</td>
<td>&quot;</td>
<td>25</td>
<td>39</td>
<td>EGN 3331C</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>ADDR</td>
<td>&quot;</td>
<td>30</td>
<td>40</td>
<td>EGN 3373</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>CITY</td>
<td>&quot;</td>
<td>10</td>
<td>41</td>
<td>EGN 3375C</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>STATE</td>
<td>&quot;</td>
<td>10</td>
<td>42</td>
<td>EGN 3343</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>ZIP</td>
<td>&quot;</td>
<td>10</td>
<td>43</td>
<td>EGN 3353C</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>PHON</td>
<td>&quot;</td>
<td>12</td>
<td>44</td>
<td>EGN 3613</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>ADVI</td>
<td>&quot;</td>
<td>20</td>
<td>45</td>
<td>EGN 3703</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>MAJOR</td>
<td>&quot;</td>
<td>15</td>
<td>46</td>
<td>EGN 3363</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>CLASS</td>
<td>&quot;</td>
<td>10</td>
<td>47</td>
<td>EGN 3704</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>ENC 1101</td>
<td>&quot;</td>
<td>10</td>
<td>48</td>
<td>STA 3032</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>ENC 1102</td>
<td>&quot;</td>
<td>10</td>
<td>49</td>
<td>PHY 3101</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>SPC 1014</td>
<td>&quot;</td>
<td>10</td>
<td>50</td>
<td>BSC 1020C</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>EUH 2000</td>
<td>&quot;</td>
<td>10</td>
<td>51</td>
<td>BSC 1030C</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>16</td>
<td>EUH 2001</td>
<td>&quot;</td>
<td>10</td>
<td>52</td>
<td>GEO 1200</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>AMH 2010</td>
<td>&quot;</td>
<td>10</td>
<td>53</td>
<td>GEO 3370</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>18</td>
<td>AMH 2020</td>
<td>&quot;</td>
<td>10</td>
<td>54</td>
<td>GLY 1000</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>19</td>
<td>HUM 2211</td>
<td>&quot;</td>
<td>10</td>
<td>55</td>
<td>ENC 4634</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>HUM 2230</td>
<td>&quot;</td>
<td>10</td>
<td>56</td>
<td>EGN 4624</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>21</td>
<td>POS 2041</td>
<td>&quot;</td>
<td>10</td>
<td>57</td>
<td>EEL 3342</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>22</td>
<td>PSY 2013</td>
<td>&quot;</td>
<td>10</td>
<td>58</td>
<td>EEL 4701</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>23</td>
<td>SYG 2000</td>
<td>&quot;</td>
<td>10</td>
<td>59</td>
<td>ECM 4504</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>24</td>
<td>ANT 2003</td>
<td>&quot;</td>
<td>10</td>
<td>60</td>
<td>ECM 4804</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>25</td>
<td>COP 3215</td>
<td>&quot;</td>
<td>10</td>
<td>61</td>
<td>EEL 4702</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>26</td>
<td>EGN 3210</td>
<td>&quot;</td>
<td>10</td>
<td>62</td>
<td>ECM 4343</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>27</td>
<td>EGN 1111C</td>
<td>&quot;</td>
<td>10</td>
<td>63</td>
<td>ECM 4124</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>28</td>
<td>ECO 2013</td>
<td>&quot;</td>
<td>10</td>
<td>64</td>
<td>ECM 4411</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>29</td>
<td>CHS 1440</td>
<td>&quot;</td>
<td>10</td>
<td>65</td>
<td>DATA STRUC</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>30</td>
<td>PHY 3048</td>
<td>&quot;</td>
<td>10</td>
<td>66</td>
<td>APPOR ELEC</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>31</td>
<td>PHY 3049</td>
<td>&quot;</td>
<td>10</td>
<td>67</td>
<td>DESIG ELEC</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>32</td>
<td>PHY 3049L</td>
<td>&quot;</td>
<td>10</td>
<td>68</td>
<td>MICELL 1</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>33</td>
<td>MAC 3311</td>
<td>&quot;</td>
<td>10</td>
<td>69</td>
<td>MICELL 2</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>34</td>
<td>MAC 3312</td>
<td>&quot;</td>
<td>10</td>
<td>70</td>
<td>MICELL 3</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>35</td>
<td>MAC 3313</td>
<td>&quot;</td>
<td>10</td>
<td>71</td>
<td>NOTE 1</td>
<td>&quot;</td>
<td>70</td>
</tr>
<tr>
<td>36</td>
<td>MAP 3302</td>
<td>&quot;</td>
<td>10</td>
<td>72</td>
<td>NOTE 2</td>
<td>&quot;</td>
<td>70</td>
</tr>
</tbody>
</table>

**TOTAL:** 958

Figure 2. Structure for Data Base "MASTER."
For the data structure, approved elective, design elective and miscellaneous courses in the data base file, nineteen character strings are reserved. The last eight characters reveal the prefix and number of the course (see Figure 3b).

![Figure 3. Course Field Samples.](image)

If a blank is entered in any position, the corresponding information is not available. For example, the student grade for the course is not available until the course is completed.

Building the student data base (master) and manipulation of information contained in the data base is performed by the "MENU.PRG" and "ALLPRO.PRG" programs in interactive fashion.
CHAPTER V
PROGRAM DESCRIPTIONS AND FUNCTIONS

MENU.PRG Program

This program presents the user with a two-part menu from which he may select operations appropriate to his needs (see Figure 4). The use of this menu entails four distinct functions:

1. Display the menu
2. Obtain the code for the selected function
3. Perform the operation indicated by the selection code
4. Loop to display the menu again

All of these functions are embedded in a loop that will be executed until the user decides to exit from the module in which the menu is contained.

The set of PUBLIC statements in the beginning of the menu program allows the variables in those statements to be recognized by the modules of "ALLPRO.PRG" file. A set of flags are checked after an operation is completed by the "ALLPRO.PRG" program file and the control is returned to the "MENU.PRG" program. These flags control the flow of the operation. Each selection from the menu executes the SET PROCEDURE TO B:ALLPRO statement which opens the procedure file (ALLPRO.PRG) on the floppy disk in drive B. Each selection also ends with the CLEAR ALL statement which closes all files and clears all
Individual Operation Menu:
1. Add a new student
2. Delete a student
3. Modify a student's record
4. Display/print a student's record

Group Information Menu:
5. Find group by advisor name
6. Find group by course name
7. Find group by classification
8. Find group by classification and course name
9. Sorted list of all students
10. Exit to operating system

Select Option:

Figure 4. Main Menu.

the variables. This will prevent the system from having too many files open at the same time. The DO statement starts the execution of the procedure module which is the ALLPRO.PRG file and its name appears in the DO statement. After the completion of the procedure module, the RETURN statement at the end of the procedure module gives the control back to the calling program or MENU.PRG program file. Exit from this program and the whole student tracking system is accomplished by the QUIT statement. This statement exits the dBASE III and lets the operating system take over. The listing of the MENU.PRG program is presented in Appendix B.
This program file contains several procedures to perform the function displayed by "MENU.PRG" program. Each procedure has been developed to be used by as many operations as possible in order to avoid duplication of codes.

Procedure "INITVAR"

The initialization of variable names corresponding to the field names in the data base is done by "INITVAR" procedure. These variables are used to allow the programmer to place a document on the screen to facilitate entry of the data. Each variable contains blank characters equal to its corresponding field, after the initialization procedure. This procedure is only used during the entry of data for new students (records). Figure 5 presents a flowchart for this procedure. The listing is found in Appendix B.

START

INITIALIZE VARIABLES

RETURN

Figure 5. Flowchart for "INITVAR" Procedure.
Procedure "ADDSTU"

This procedure was constructed to direct the data entry for a new student. The social security number is a unique data element for each student and it is also the first entry of the student's record. To prevent any duplication of records, after the social security number of the student is entered, the "ADDSTU" procedure searches the data base for that social security number, a message is printed and re-entry is allowed or data entry is allowed, correspondingly. Then, a call for procedure "GETDATA" is executed, a flag called "ANSWER" is checked and, after "GETDATA" operation is completed, control is returned to the menu program. There are two exit points from this procedure; the first is during the social security entry (if a blank is entered), and the second is when the "ANSWER" flag is checked. This flag is set at the end of "GETDATA," indicating whether another record entry is desired. Figure 6 presents a flowchart for this procedure and the listing can be found in Appendix B.

Procedure "GETDATA"

To build the database "MASTER," the data entry process is performed under the control of the "GETDATA" procedure. The entry of data is through the variables initialized by the "INITVAR" procedure and by a full page editing process. At the end of each page, the user is asked to confirm the entries of that page and the action is taken toward continuation of data entry or re-entry of data.
START
OPEN "MASTER" AND SS# INDEX FILE
PRINT TO SCREEN
INITIALIZE SS#
GET SS#
IS 1ST CHARACTER BLANK?
yes
no
DOES SS# ALREADY EXIST?
no
yes
CALL "GETDATA" PROCEDURE
ANOTHER ENTRY
no
RETURN

Figure 6. Flowchart of the "ADDSTU" Procedure.
During each page of data entry, the user is allowed to move around and make any changes necessary. When all the data for that student (record) is entered, the REPLACE command of dBASE III takes the data elements from variable names and writes them in the data base under the appropriate fields.

The "GETDATA" procedure is also used to display a record on the screen for viewing. If the flag "SCREFLAG" is set, the change of data elements is not allowed. In this case, a record of the student is displayed.

Another use of the "GETDATA" procedure is for the purpose of modifying a student's record. The process is the same as when adding a new student, except the data elements are taken from the data base and not initialized to blank characters. A listing of this procedure can be found in Appendix B. The flowchart is shown in Figure 7.

Procedure "FINDSTU"

In order to delete, view or modify a student's record, the position of that record in the data base must be determined. The "FINDSTU" procedure locates the record of a student in the data base "MASTER" by either the student's name or the student's social security number. This procedure allows the user to choose either one as a search criteria, in a menu-driven fashion. The sequential search is exercised when a name is chosen for search criteria, and random access search is performed when a social security number is the search criteria. The reason is that the social security number is unique for each student,
Figure 7. Flowchart of the "GETDATA" Procedure.
therefore, the index file could be searched which provides random access to records in the database. But, the last name of each student is not quite unique. Therefore, two fields have to be searched (last name and first-middle names) and matched with the given criteria. The procedure provides an exit option in its menu. If the exit function is chosen, meaning the find operation is not completed, a flag (FLAG1) is set in order for the "MENU.PRG" program to skip the normal path and display the main menu. This precaution prevents the records in the database to be altered accidently. A listing of this procedure is found in Appendix B. Figure 8 shows a flowchart of the "FINDSTU" procedure.

Procedure "DELSTU"

After the student's record is found, this procedure erases that record if the confirmation has been made by the user; otherwise, the record will be left unaltered. The DELETE statement of dBASE III flags the record for deletion and the PACK command actually erases the record. Figure 9 provides a flowchart for this procedure and a listing can be found in Appendix B.

Procedure "OUTCHOIC"

To obtain a report on a student record or to view a student's record, a procedure called "OUTCHOIC" is used. This procedure provides the user with a choice between the printer output or the screen output. Depending on the user's response, certain flags are set, instructions are given and the task is performed. For screen viewing,
Figure 8. Flowchart of the "FINDSTU" Procedure.
Figure 9. Flowchart of the "DELSTU" Procedure.
the procedure "GETDATA" is used to output the record to the screen, but prevents the user from altering any data item. In order to get a hard copy of the student's record, more codes were written to control the printer output. Figure 10 shows a flowchart and Appendix B provides a listing of the "OUTCHOIC" procedure.

Procedure "FGCORS"

This procedure is constructed to find a group of students who have completed or are in the process of completing a particular course. The user is provided with the name and social security number of the students, with the total number of the students or just the total number of the students having taken that course. The only input provided by the user is the prefix and the number of the course under investigation, which must be entered with an underscore line separating the prefix and the number (e.g., ECM_3341). This course name must also be one of the field names in the data base. Once the course is input by the user, the data base field corresponding to the input name is searched sequentially and if there is an entry in a record for that field, the names or the count or both is printed. A blank field is discarded. A flowchart of this procedure is shown in Figure 11. A listing of this procedure is shown in Appendix B.

Procedure "FGADVI"

This procedure is very similar to the "FGCORS" procedure, except that the field of search is the advisor name field (ADVI) in the data base. Since the flowchart is similar to the flowchart of the "FGCORS"
Figure 10. Flowchart of the "OUTCHOIC" Procedure.
Figure 11. Flowchart of the "FGCORS" Procedure.
procedure, please refer to Figure 11 and replace the course name with the advisor's name. A listing of the "FGADVI" program is presented in Appendix B.

**Procedure "FGCLASS"**

This procedure is also very similar to the "FGCORS" procedure, except that the classification field (CLASS) in the data base is searched. The flowchart is similar to Figure 11 except that the classification is entered instead of the course name. A listing is provided in Appendix B.

**Procedure "FGCORCLS"**

This procedure will cross-check the two fields (class field and course field) to obtain the records which satisfy the criteria provided by the user. The user is to provide the classification category (freshman, sophomore, junior or senior) and the course name (prefix and number). Once this is done, the procedure will find all the students that are in the given classification and have taken the given course. The flowchart is similar to the flowchart of the "FGCORS" procedure (shown in Figure 11), except both the course name and classification categories are entered and the search is conducted based on the entries of both fields. A listing of this procedure is found in Appendix B.
Procedure "SORTALST"

To obtain a sorted list of all the students in the data base "MASTER," this procedure is used. The SORT TO command of dBASE III is executed to accomplish this task. This command creates a new data base file from the selected records and fields in the current (in use) data base file. The sorting will be in ascending order and a listing will be printed by the printer. A listing of this procedure is shown in Appendix B. A flow chart is shown in Figure 12.

NOTE: The modification of a student record is performed by using the "FINDSTU" and "GETDATA" procedures. Therefore, a separate procedure to perform this task was not necessary.
CHAPTER VI
SYSTEM INTEGRITY AND SECURITY

Throughout the interactive programs, numerous integrity constraints and security measures have been installed to protect the system from possible user or software errors.

The problem of integrity is the problem of ensuring—insofar as it can be ensured—that the data in the database is accurate at all times. There is a limit on the extent to which this objective can be achieved. In particular, the system cannot check the correctness of every individual value entered into the database... although it clearly can check each such value for plausibility...(Van Rijsbergen 1984).

Therefore, the integrity constraints existing in the student tracking information system programs check the plausibility of user inputs.

Security, in this case, refers to the measures taken to guard the information in the database against total or partial loss. In order to preserve the information in the database, each time a database is successfully updated, a copy of the database (backup) must be made on another file. In case of a system crash due to hardware, software or environmental causes resulting in the total or partial loss of the database, the contents of the backup file may be copied into the database. Then, the activity which was in progress during the crash must be started anew. Because the system records data on disk after the completion of obtaining information for each record, the user should not be concerned about saving any data or losing more
than one record modification. Complete directions for making a backup of the data base file and the recovery of information are available in Appendix D.
CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

The interactive student tracking information system provides a convenient, menu-driven system capable of providing a report on a student's academic record and categorizing records of students enrolled in the Computer Engineering program. The reports on a student record will provide the advisor with an overview of the student's academic progress. The categorizing of all the students' records on the basis of given criteria will give the faculty advisors some foresight into course scheduling for future semesters.

Although integrity constraints and security measures have been incorporated in all of the interactive programs, only time and system usage will tell whether or not these measures are sufficient.

This system has been designed considering the possibility of later expansion. Some recommended expansions are:

1. The data base file and application programs could be modified to contain and operate on the courses on a new catalog. This could be accomplished by creating a new data base file that contains the courses for the new catalog. Then, all the procedures could be used for the new data base without any modification with the exception of the procedures containing the name of the course that has been changed for the new catalog. In this case, the procedures could be modified
very easily and in a similar fashion to perform the task based on the courses in the new data base. In the case of addition of a course to the new catalog, a few lines of codes, similar to the existing codes, could be added to the procedures (see Appendix B for listing of procedures).

2. The existing system could perform a search based on one field when the appropriate option is selected. The system could be modified to do search based on multiple fields. For example, the procedure "FGCORS" could be modified to do multiple search based on the name of several courses (field names).

3. Other procedures could be written for the system to be able to calculate the grade point average of a student for different categories of courses, as well as the overall grade point average. For this purpose, the content of the course fields of a student record must contain the grade for that course in a particular position and consistency must be maintained during the adding and modifying process of a student's record.

4. This system could also be expanded to contain information on the course and their prerequisites and to reflect this information on a student's record. Such an expansion requires an extensive amount of coding and a good knowledge of dBASE III package.

To make any modification or expansion to the existing system, a general recommendation is to have a knowledge of dBASE III, maintain consistency, maintain a menu-driven system and keep it simple. The
limitation of this system is basically the limitation of the dBASE III data base manager package (please refer to pages 11 and 12 of this document for more information).

It is hoped that the documentation contained herein will be sufficient to aid any future expansion or programming efforts. Appendix A provides the path of execution of the programs and their interaction with the data base of the student tracking system.
APPENDICES
APPENDIX A

SYSTEM STRUCTURE DIAGRAM
APPENDIX B
LISTING OF PROGRAMS
MAIN MENU DRIVER FOR THE INFORMATION SYSTEM

MENU.PRG

BY F. BEHi 5/28/85

---------------------------------------------

MASSAGE TO USER

Please use CAPITAL letters to answer the prompts

SET PF: INT OFF

SET TAU< OFF

SET SAFETY OFF

NO HELP NEEDED

OPEN PROCEDURE FILE

SET PROCEDURE TO B:ALLPRO

DECLEAR THE FOLLOWING VARIABLES AS GLOBAL

PUBLIC VSSND, VNAME, VNAME, VADDR, VICTY, VSTATE, VZIP, VPHONE, VADV

PUBLIC VMAJOR, VCLASS, VENC1101, VENC1102, VSPC1014, VEUH2000, VEUH2001

PUBLIC VAMH2010, VAMH2020, VHUM2211, VHUM2230, VPOS2041, VSYS2013, VSYS2000

PUBLIC VANT2003, VCAP5215, VEGN3210, VEGN1111C, VEC02003, VPHY3048

PUBLIC VPHY3049, VPHY3049L, VMAC3311, VMAC3312, VMAC3313, VMAC3302, VEGN3311

PUBLIC VEGN3321, VEGN3331C, VEGN3375, VEGN3375C, VEGN3343, VEGN3353C, VEGN3313

PUBLIC VEGN3703, VEGN3563, VEGN3570, VSTA2032, VPHY3101, VBSC1020C, VBSC1030C

PUBLIC VEG10200, VEG10370, VGLY1000, VENC4634, VEGN4624, VEGN3342, VEEL4701

PUBLIC VECM4504, VECM4804, VEEL4702, VECM4343, VECM4124, VECM4411, VMEMSTRUC

PUBLIC VAPPORLEC, VDESILELEC, VMICELL1, VMICELL2, VMICELL3, VMICELL4, VMICELL5

PUBLIC FLAG, PRFLAG, RECFLAG, ADMDFLAG, ADDFLAG

PRINT THE MENU ON THE SCREEN

SET DEVICE TO SCREEN

CLEAR THE SCREEN

PRINT "+++++ INDIVIDUAL OPERATION MENU +++++"

PRINT THE SYSTEM DATE

PRINT DATE()

PRINT "1.---- ADD A NEW STUDENT"

PRINT "2.---- DELETE A STUDENT"

PRINT "3.---- MODIFY A STUDENT'S RECORD"

PRINT "4.---- DISPLAY/PRINT A STUDENT'S RECORD"

PRINT 3 BLANK LINES

GROUP INFORMATION DRIVER (MENU)

PRINT "+++++ GROUP INFORMATION MENU +++++"

PRINT "5.---- FIND GROUP BY ADVISOR NAME"

PRINT "6.---- FIND GROUP BY COURSE NAME"
* PRINT 3 BLANK LINES

* WAIT FOR INPUT
  WAIT " SELECT OPTION : " TO OPTION
* CHECK VALID INPUT
  DO CASE
    CASE OPTION = "1"
    * EXECUTE THE FOLLOWING PROCEDURES
      DO INITVAR
      DO ADDSTU
    CASE OPTION = "2"
    * EXECUTE "FINDSTU" PROGRAM
      DO FINDSTU
    * CHECK FOR EXISTENCE OF A RECORD
      IF FLAG1=1
        STORE 0 TO FLAG1
      LOOP
    ENDIF
    * EXECUTE "DELSTU" PROGRAM
      DO DELSTU
    CASE OPTION = "3"
    * SET THE MODIFY MODE FLAG
      STORE 1 TO ADMDFLAG
    * EXECUTE "FINDSTU" PROGRAM
      DO FINDSTU
    * CHECK FOR EXISTENCE OF A RECORD
      IF FLAG1=1
        STORE 0 TO FLAG1
      LOOP
    ENDIF
    * RESTORE VARIABLES WITH THE FIELD CONTENTS
    * EXECUTE "GETVARI" PROCEDURE
      DO GETVARI
    * EXECUTE "GETDATA" PROCEDURE
      DO GETDATA
    CASE OPTION = "4"
    * EXECUTE "FINDSTU" PROCEDURE
      DO FINDSTU
    * CHECK FOR EXISTENCE OF A RECORD
      IF FLAG1=1
        STORE 0 TO FLAG1
      LOOP
    ENDIF
    * EXECUTE "OUTCHOIC" PROCEDURE
      DO OUTCHOIC
    * EXECUTE "GETVARI" PROCEDURE
      DO GETVARI
    * EXECUTE "GETDATA" PROCEDURE
      IF PRFLAG=1
STORE 0 TO PRFLAG
DO PRIREP
ELSE
DO GETDATA
ENDIF
SET FUNCTION 10 TO "DO B: MENU"
CASE OPTION = "5"
* EXECUTE "FGADV1" PROCEDURE
DO FGADV1
CASE OPTION = "6"
* EXECUTE "FGCORS" PROCEDURE
DO FGCORS
CASE OPTION = "7"
* EXECUTE "FGCLASS" PROCEDURE
DO FGCLASS
CASE OPTION = "8"
* EXECUTE "FGCLASS" PROCEDURE
DO FGCLASS
CASE OPTION = "9"
* EXECUTE "SORTALL" PROCEDURE
DO SORTALL
CASE OPTION = "E"
CLEAR ALL
* EXIT TO OPERATING SYSTEM
QUIT
* CHECK FOR ILLEGAL OPTION
OTHERWISE
* PRINT MESSAGE
? "ILLEGAL OPTION, PLEASE RESELECT OPTION"
* WAIT FOR INPUT
WAIT
LOOP
ENDCASE
ENDDO
THIS IS THE PROCEDURE FILE. ALL OF THE PROCEDURES THAT ARE USED FOR DIFFERENT OPERATIONS OF THE SYSTEM ARE CONTAINED IN THIS FILE. THE REASON FOR HAVING ONE LARGE FILE NESTED OF MANY SMALL PROGRAM FILES IS TO HAVE AS FEW NUMBER OF FILES OPEN AT ANY ONE TIME AS POSSIBLE.

===============================================

ALLPRO.PRG
BY F. BEHI 6/10/85

PROCEDURE INITVAR
* PROCEDURE TO INITIALIZE VARIABLES
* INITVAR.PRG
* BY F. BEHI 5/31/85

* INITIALIZE VARIABLES
STORE SPACE(20) TO VLNAME
STORE SPACE(25) TO VFNAME
STORE SPACE(25) TO VADDR
STORE SPACE(10) TO VCIY
STORE SPACE(10) TO VSTATE
STORE SPACE(10) TO VZIP
STORE SPACE(12) TO VPHON
STORE SPACE(20) TO VADVI
STORE SPACE(15) TO VMAJOR
STORE SPACE(10) TO VCLASS
STORE SPACE(10) TO VENC1101
STORE SPACE(10) TO VENC1102
STORE SPACE(10) TO VSFC1014
STORE SPACE(10) TO VEUH2000
STORE SPACE(10) TO VEUH2001
STORE SPACE(10) TO VAMH2010
STORE SPACE(10) TO VAMH2020
STORE SPACE(10) TO VHUM2011
STORE SPACE(10) TO VHUM2030
STORE SPACE(10) TO VPOS2041
STORE SPACE(10) TO VPSY2013
STORE SPACE(10) TO VSYG2000
STORE SPACE(10) TO VANT2003
STORE SPACE(10) TO VCDP3215
STORE SPACE(10) TO VEOG3210
STORE SPACE(10) TO VEOG3210C
STORE SPACE(10) TO VEOU2013
STORE SPACE(10) TO VCHS1440
STORE SPACE(10) TO VPHY3048
STORE SPACE(10) TO VPHY3049
STORE SPACE(10) TO VPHY3049L
STORE SPACE(10) TO VMAC3311
STORE SPACE(10) TO VMAC3312
STORE SPACE(10) TO VMAC3312
STORE SPACE(10) TO VMAC3313
STORE SPACE(10) TO VMAP3302
STORE SPACE(10) TO VEGN3311
STORE SPACE(10) TO VEGN3321
STORE SPACE(10) TO VEGN3331C
STORE SPACE(10) TO VEGN3373
STORE SPACE(10) TO VEGN3375C
STORE SPACE(10) TO VEGN3343
STORE SPACE(10) TO VEGN3353C
STORE SPACE(10) TO VEGN3613
STORE SPACE(10) TO VEGN3613
STORE SPACE(10) TO VEGN3704
STORE SPACE(10) TO VSTAS3032
STORE SPACE(10) TO VPHYS101
STORE SPACE(10) TO VBSC1020C
STORE SPACE(10) TO VBSC1030C
STORE SPACE(10) TO VGE01203
STORE SPACE(10) TO VGE01370
STORE SPACE(10) TO VGLY1000
STORE SPACE(10) TO VENC4634
STORE SPACE(10) TO VEGN4624
STORE SPACE(10) TO VEEL3342
STORE SPACE(10) TO VEEL4701
STORE SPACE(10) TO VECM4504
STORE SPACE(10) TO VECM4804
STORE SPACE(10) TO VEEL4702
STORE SPACE(10) TO VECM4343
STORE SPACE(10) TO VECM4124
STORE SPACE(10) TO VECM4411
STORE SPACE(20) TO VDATASTRUC
STORE SPACE(20) TO VIAPPFORELEC
STORE SPACE(20) TO VDESIGELE
STORE SPACE(20) TO VMICELL1
STORE SPACE(20) TO VMICELL2
STORE SPACE(20) TO VMICELL3
STORE SPACE(70) TO VNOTE1
STORE SPACE(70) TO VNOTE2
RETURN

*=====================================================================
PROCEDURE GETDATA
*----------------------------------------------------------------------
PUBLIC ANSWER
DO WHILE .T.
STORE .T. TO PAGE
STORE SPACE(1) TO VANSW
DO WHILE PAGE
CLEAR
* INITIALIZE OUTPUT FORM. ROW VARIABLE IS USED TO KEEP TRACK
* OF POSITION OF THE CURSOR. THIS ENABLES THE SYSTEM TO USE
* THIS PROCEDURE FOR PRINTER AND SCREEN OUTPUTS
STORE 1 TO ROW
@ ROW,1 SAY "STUDENT RECORD"
ROW=ROW+1
@ ROW,1 SAY "SOCIAL SECURITY NUMBER " + VSSNO
ROW=ROW+1
@ ROW,1 SAY "LAST NAME " GET VLNAME
ROW=ROW+1
@ ROW,1 SAY "FIRST AND MIDDLE NAME " GET VFNAME
ROW=ROW+1
\( \text{ROW, 1 SAY "STREET ADDRESS" GET VADDR} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "CITY" GET VCITY} \)
\( \text{ROW, 20 SAY "STATE" GET VSTATE} \)
\( \text{ROW, 40 SAY "ZIP CODE" GET VZIP} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "TELEPHON NUMBER" GET VPHON} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "ADVISOR NAME" GET VADV} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "MAJOR" GET VMajor} \)
\( \text{ROW, 40 SAY "CLASSIFICATION" GET VCLASS} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "ENC_1101" GET VENC1101} \)
\( \text{ROW, 25 SAY "ENC_1102" GET VENC1102} \)
\( \text{ROW, 50 SAY "SPC_1014" GET VSPC1014} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "EUH_2000" GET VEUH2000} \)
\( \text{ROW, 25 SAY "EUH_2001" GET VEUH2001} \)
\( \text{ROW, 50 SAY "AMH_2010" GET VAMH2010} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "AMH_2020" GET VAMH2020} \)
\( \text{ROW, 25 SAY "HUM_2211" GET VHUM2211} \)
\( \text{ROW, 50 SAY "HUM_2230" GET VHUM2230} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "POS_2041" GET VP052041} \)
\( \text{ROW, 25 SAY "PSY_2013" GET VP052013} \)
\( \text{ROW, 50 SAY "SYG_2000" GET VSYG2000} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "ANT_2003" GET VANT2003} \)
\( \text{ROW, 25 SAY "CFP_3215" GET VCFP3215} \)
\( \text{ROW, 50 SAY "EGN_3210" GET VEGN3210} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "EBN_1111C" GET VEBN1111C} \)
\( \text{ROW, 25 SAY "ECO_2013" GET VECO2013} \)
\( \text{ROW, 50 SAY "CHS_1440" GET VCHS1440} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "PHY_3048" GET VFHY3048} \)
\( \text{ROW, 25 SAY "PHY_3049" GET VFHY3049} \)
\( \text{ROW, 50 SAY "PHY_3049L" GET VFHY3049L} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "MAC_3311" GET VMAC3311} \)
\( \text{ROW, 25 SAY "MAC_3312" GET VMAC3312} \)
\( \text{ROW, 50 SAY "MAC_3313" GET VMAC3313} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "MAP_3302" GET VMAP3302} \)
\( \text{ROW, 25 SAY "EGN_3311" GET VEGN3311} \)
\( \text{ROW, 50 SAY "EGN_3321" GET VEGN3321} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "EGN_3331C" GET VEGN3331C} \)
\( \text{ROW, 25 SAY "EGN_3331C" GET VEGN3331C} \)
\( \text{ROW, 50 SAY "EGN_3375C" GET VEGN3375C} \)
\( \text{ROW}=\text{ROW}+1 \)
\( \text{ROW, 1 SAY "EGN_3343" GET VEGN3343} \)
\( \text{ROW, 25 SAY "EGN_3353C" GET VEGN3353C} \)
@ ROW,50 SAY "EGN_3613" GET VEGN3613
ROW=ROW+1
@ ROW,1 SAY "EGN_3703" GET VEGN3703
@ ROW,25 SAY "EGN_3363" GET VEGN3363
@ ROW,50 SAY "EGN_3704" GET VEGN3704
* CHECK FLAG FOR RECORD VIEWING
IF SCREFLAG =1
* DISABLE CHANGING OF THE RECORD
CLEAR GETS
WAIT
ENDIF
* CHECK FOR CORRECTNESS OF THE PAGE INPUTS,
* WHEN IN ADDING OR MODIFYING MODE
IF ADMDFLAG=1
@ ROW,1 SAY "ARE ALL CORRECT (Y/N) " GET VANSW
READ
IF VANSW="N"
STORE SPACE(1) TO VANSW
LOOP
ENDIF
STORE .F. TO PAGE
ENDDO WHILE PAGE
* SECOND PAGE
CLEAR
STORE .T. TO PAGE
STORE SPACE(1) TO VANSW
STORE 1 TO ROW
DO WHILE PAGE
@ ROW,1 SAY "++++++ CONTINUE STUDENT RECORD +++++++"
ROW=ROW+2
@ ROW,1 SAY "STA_3032" GET VSTA3032
@ ROW,25 SAY "PHY_3101" GET VPHY3101
@ ROW,50 SAY "BSC_1020C" GET VBSG1020C
ROW=ROW+1
@ ROW,1 SAY "BSC_1030C" GET VBSG1030C
@ ROW,25 SAY "GEO_1200" GET VGEO1200
@ ROW,50 SAY "GEO_3370" GET VGEO3370
ROW=ROW+1
@ ROW,1 SAY "GLY_1000" GET VGLY1000
@ ROW,25 SAY "ENC_4634" GET VENC4634
@ ROW,50 SAY "GEN_4624" GET VGEN4624
ROW=ROW+1
@ ROW,1 SAY "EEL_3342" GET VEEL3342
@ ROW,25 SAY "EEL_4701" GET VEEL4701
@ ROW,50 SAY "ECM_4504" GET VECM4504
ROW=ROW+1
@ ROW,1 SAY "ECM_4804" GET VECM4804
@ ROW,25 SAY "EEL_4702" GET VEEL4702
@ ROW,50 SAY "ECM_4343" GET VECM4343
ROW=ROW+1
@ ROW,1 SAY "ECM_4124" GET VECM4124
@ ROW,25 SAY "ECM_4411" GET VECM4411
ROW=ROW+1
& ROW,1 SAY "DATA_STRUCTURE" GET VDATASTRUCT
& ROW,35 SAY "AFFORDED_ELECTIVE" GET VAPFFORELEC
ROW=ROW+1
& ROW,1 SAY "DESIGN_ELECTIVE" GET VDESIGELEC
& ROW,35 SAY "OTHER_COURSES 1" GET VMICELL1
ROW=ROW+1
& ROW,1 SAY "OTHER_COURSES 2" GET VMICELL2
& ROW,35 SAY "OTHER_COURSES 3" GET VMICELL3
ROW=ROW+1
& ROW,1 SAY "NOTE " GET VNOTE1
ROW=ROW+1
& ROW,1 SAY "NOTE " GET VNOTE2
* CHECK FLAG FOR RECORD VIEWING
  IF SCREFLAG=1
    STORE 0 TO SCREFLAG
* DISABLE CHANGING OF THE RECORD
  CLEAR GETS
  WAIT
  RETURN
ENDIF
* CHECK FOR CORRECTNESS OF THE PAGE INPUTS,
  * WHEN IN ADDING OR MODIFYING MODE
  IF ADMAFLAG=1
    STORE 0 TO ADMAFLAG
    ROW=ROW+1
& ROW,1 SAY "ARE ALL CORRECT (Y/N)" GET VANSW
  READ
    IF VANSW="N"
      STORE SPACE(1) TO VANSW
      STORE 1 TO ADMAFLAG
    LOOP
  ENDIF
ENDIF
STORE .F. TO PAGE
ENDDO WHILE PAGE
* CHECK FOR NEED OF AN EMPTY RECORD
  IF ADDFLAG =1
  * CREATE AN EMPTY RECORD
    APPEND BLANK:
  ENDF
  * PUTTING DATA IN RECORD FIELDS
    REPLACE SSNO WITH VSSNO,LNAME WITH VLNAME,FNAME WITH VFNAME,;
    ADDR WITH VADDR,CITY WITH VCITY,STATE WITH VSTATE
    REPLACE ZIP WITH VZIP,PHON WITH VPHON,ADVI WITH VADVI,;
    MAJOR WITH VMAJOR,CLASS WITH VCLASS
    REPLACE ENC_1101 WITH VENC1101,ENC_1102 WITH VENC1102,;
    SPC_1014 WITH VSPC1014,EUH_2000 WITH VEUH2000
    REPLACE ENU_2001 WITH VENUH2001,AMH_2010 WITH VAMH2010,;
    AMH_2020 WITH VAMH2020,HUM_2211 WITH VHUM2211
    REPLACE UH_2230 WITH VHUM2230,FOS_2011 WITH VFOS2011,;
    PSY_2013 WITH VFPSY2013,SY6_2200 WITH VSY62200
    REPLACE ANT_2003 WITH VANT2003,COF_3215 WITH VCOP3215,;
    EBN_3210 WITH VEBN3210,EBN_111C WITH VEBN111C
    REPLACE ECD_2013 WITH VECO2013,CHS_1440 WITH VCHS1440,;
    PHY_3048 WITH VPHY3048,PHY_3049 WITH VPHY3049
REPLACE PHY_3049L WITH VPHY3049L, MAC_3311 WITH VMAC3311,;
MAC_3312 WITH VMAC3312, MAC_3313 WITH VMAC3313
REPLACE MAP_3302 WITH VMAP3302, EGN_3311 WITH VEGN3311,;
EGN_3312 WITH VEGN3312, EGN_3313C WITH VEGN3313C
REPLACE EGN_3373 WITH VEGN3373, EGN_3375C WITH VEGN3375C,;
EGN_3343 WITH VEGN3343, EGN_3355C WITH VEGN3355C
REPLACE EGN_3613 WITH VEGN3613, EGN_3702 WITH VEGN3702,;
EGN_3363 WITH VEGN3363, EGN_3704 WITH VEGN3704
REPLACE STA_3032 WITH VSTA3032, PHY_3101 WITH VPHY3101,;
BSC_1020C WITH VBSC1020C, BSC_1030C WITH VBSC1030C
REPLACE GEO_1200 WITH VGEO1200, GEO_3370 WITH VGEO3370,;
GLY_1000 WITH VGILY1000, ENC_4634 WITH VENC4634
REPLACE EGN_4624 WITH VEGN4624, EEL_3342 WITH VEEL3342,;
EEL_4701 WITH VEEL4701, ECM_4504 WITH VECM4504
REPLACE ECM_4804 WITH VECM4804, EEL_4702 WITH VEEL4702,;
ECM_4543 WITH VECM4543, ECM_4124 WITH VECM4124
REPLACE ECM_4411 WITH VECM4411, DATA_STRUC WITH VDATASTRUC,;
APPOR_ELEC WITH VAPPORELEC, DESIG_ELEC WITH VDESIGELEC
REPLACE MICELL_1 WITH VMICELL1, MICELL_2 WITH VMICELL2,;
MICELL_3 WITH VMICELL3
REPLACE NOTE1 WITH VNOTE1, NOTE2 WITH VNOTE2
* CHECK FOR ANOTHER ENTRY
IF ADDFLAG = 1
  CLEAR
  @ 10,2 SAY "DO YOU WISH TO ADD ANOTHER STUDENT ? (Y/N) " GET ANSWER
  READ
  RETURN
ELSE
  STORE "N" TO ANSWER
  RETURN
ENDIF
ENDDO WHILE .T.
RETURN
* =----------------------------------------------------------------------
PROCEDURE ADDSTU
* ADD A NEW STUDENT PROCEDURE
* ADDSTU
* BY F. BEHI 5/29/85
*----------------------------------------------------------------------
* CHECK FOR ANOTHER ENTRY
PUBLIC ANSWER
IF ANSWER ="N"
  STORE SPACE(1) TO ANSWER
  RETURN
ENDIF
* SELECT AND USE THE DATA FILE AND INDEX FILE
SELECT 1
USE B:MASTER INDEX B:SSNDINX
* DATA ENTRY SETUP
DO WHILE .T.
  SET DEVICE TO SCREEN
  * ENTERING SOCIAL SECURITY NUMBER AS UNIQUE IDENTIFIER
  STORE SPACE(11) TO VSSNO
  CLEAR
DO 2,1 SAY "INPUTING STUDENT RECORD"
DO 2.50 SAY DATE()
DO 5.1 SAY "ENTER SOCIAL SECURITY NUMBER" GET VSSNO PICTURE "999-99-9999"
DO 23.1 SAY "ENTER A BLANK TO EXIT"

* GET INPUT
READ
IF SUBSTR(VSSNO,1,1) = " "
   RETURN
ENDIF

* CHECK FOR DUPLICATE RECORD
SEEK VSSNO
IF .NOT. EOF()
   SAY "A RECORD FOR THIS STUDENT IS ALREADY ON FILE"
   SAY "PLEASE CHECK THE SOCIAL SECURITY NUMBER"
   WAIT
   LOOP
ELSE

* SET FLAG FOR ADDING STUDENT
STORE 0 TO SCREFLAG

* SET ADD MODE FLAG
STORE 1 TO ADMDFLAG
STORE 1 TO ADDFLAG
DO GETDATA
   IF ANSWER ="N"
      STORE SPACE(1) TO ANSWER
      RETURN
   ENDIF

* INITIALIZE VARIABLES FOR NEXT ENTRY
   DO INITVAR
   LOOP
ENDIF

ENDDO
RETURN

*==================================================================
PROCEDURE FINDSTU
* PROEDURE TO FIND A STUDENT'S RECORD
* FINDSTU
* F. BEHI 6/7/85
*==================================================================
* MAKE THE VARIABLE GLOBAL
PUBLIC FLAGI
* GET THE STUDENT IDENTIFIER
SET DEVICE TO SCREEN
DO WHILE .T.
   CLEAR
   DO 2,1 SAY "WHICH IDENTIFIER DO YOU WISH TO USE "
   DO 3.1 SAY "TO FIND THE STUDENT RECORD ?"
   DO 5.1 SAY "A.---- NAME"
   DO 6.1 SAY "B.---- SOCIAL SECURITY NUMBER"
   DO 7.1 SAY "C.---- EXIT TO PREVIOUS MENU"
* PRINT 3 BLANK LINES
   ?
   ?
* WAIT FOR OPTION
  WAIT "SELECT OPTION : " TO OPT
  DO CASE
* CHECK VALID INPUT
  CASE OPT ="A"
* FIND THE STUDENT BY NAME
* USE THE DATA BASE FILE AND THE NAME INDEX FILE
  USE B:MASTER
  CLEAR
* INITIALIZE VARIABLES
  STORE SPACE(15) TO VLNAME
  STORE SPACE(20) TO VFNAME
* GET STUDENT NAME
  @ 4,1 SAY "PLEASE ENTER THE LAST NAME" GET VLNAME
  @ 6,1 SAY "PLEASE ENTER THE FIRST AND MIDDLE NAME" GET VFNAME
  @ 9,1 SAY "NOTE THAT THE NAME MUST BE ENTERED AS IT WAS WHEN"
  @ 10,1 SAY "THE RECORD WAS CREATED FOR THIS STUDENT."
  READ
* FIND THE RECORD
  LOCATE FOR LNAME=VLNAME .AND. FNAME=VFNAME
  IF EOF()
    ? "NO RECORD FOR THIS STUDENT EXIST"
    ? "PLEASE CHECK THE NAME"
    WAIT
    LOOP
  ELSE
    RETURN
  ENDIF
  CASE OPT ="B"
* SELECT AND USE THE DATA BASE AND THE INDEX FILE
  USE B:MASTER INDEX B:SSNOINDEX
  CLEAR
* INITIALIZE VARIABLES
  STORE SPACE(11) TO VSSNO
  STORE .T. TO OK
  STORE SPACE(1) TO ANSW
* GET IDENTIFIER
  DO WHILE OK
    @ 2,1 SAY "PLEASE ENTER THE SOCIAL SECURITY NUMBER" GET VSSNO PI "999-99-9999"
    @ 7,1 SAY "IS ALL CORRECT ? (Y/N) " GET ANSW
    READ
    IF ANSW="N"
      STORE SPACE(1) TO ANSW
      LOOP
    ELSE
      STORE .F. TO OK
    ENDIF
  ENDDO WHILE OK
* FIND THE RECORD USING INDEX FILE
  SEEK VSSNO
  IF EOF()
    ? "NO RECORD FOR THIS STUDENT EXIST"
    ? "PLEASE CHECK THE S.S.#"
    LOOP
  ELSE
PROCEDURE DELSTU
* PROCEDURE TO ERASE A STUDENT RECORD
* DELSTU
* -----------------------------------------------
* CLEAR THE SCREEN
CLEAR
* INITIALIZE VARIABLE
STORE SPACE(I) TO ANSW
* PRINT MESSAGE ON THE SCREEN
♫ 4.1 SAY "****** BE CAREFUL ******"
♫ 5.1 SAY "THE FOLLOWING STUDENT'S FILE WILL BE ERASED"
♫ 7.1 SAY "LAST NAME : " +LNAME
♫ 8.1 SAY "FIRST-MIDDLE NAME : " +FNAME
♫ 9.1 SAY "SOCIAL SECURITY # : " +SSNO
♫ 11.1 SAY "IS THIS WHAT YOU WISH ? " GET ANSW
* WAIT FOR CONFIRMATION
READ
IF ANSW="Y"
* MARK THE RECORD FOR DELITION
    DELETE NEXT 1
* DELETE THE RECORD
    PACK
ENDIF
* GIVE MESSAGE IF CONFIRMATION IS NEGATIVE
CLEAR
♫ 10.1 SAY "*** THE FILE WAS NOT ERASED ***"
WAIT
RETURN
*=================================================================
PROCEDURE OUTCHOIC
* PROCEDURE TO DISPLAY OR PRINT A STUDENT RECORD
* INITIALIZE VARIABLE
STORE .T. TO PRORSC
* CLEAR THE SCREEN
CLEAR
DO WHILE PRORSC
    STORE SPACE(1) TO ANSW
* CHECK FOR THE PRINTER OR SCREEN OUTPUT
CLEAR
DO 2,1 SAY "PLEASE ENTER : "
\* 4,16 SAY " P ---- FOR PRINTER OUTPUT 
\* 5,16 SAY " S ---- FOR SCREEN OUTPUT "
?
?
* WAIT FOR RESPONSE
WAIT "YOUR CHOICE : " TO ANSW
DO CASE
   CASE ANSW="P"
      CLEAR
   * CHECK PRINTER CONDITION
      \* 5.1 SAY "PLEASE CHECK THE FOLLOWINGS BEFORE PROCEEDING : "
      \* 7.5 SAY "1. TURN PRINTER ON"
      \* 8.5 SAY "2. PUT PAPER IN THE PRINTER"
      \* 9.5 SAY "3. PUT PRINTER ON LINE"
      WAIT
   * SEND OUTPUT TO PRINTER
      SET DEVICE TO PRINT
      STORE 1 TO PRFLAG
      RETURN
   ENDF
   CASE ANSW="S"
   * SEND OUTPUT TO SCREEN
      SET DEVICE TO SCREEN
      STORE 1 TO SCRFLAG
      RETURN
   ENDF
   OTHERWISE
   \* 15.1 SAY "PLEASE RESPOND WITH P OR S "
   WAIT
   LOOP
ENDCASE
ENDDO WHILE PRORSC
RETURN

*==================================================================
PROCEDURE F6ADV1
* PROCEDURE TO PRINT OR DISPLAY THE NAME
* OF ALL THE STUDENTS THAT HAVE A COMMON ADVISOR
*==================================================================
* INITIALIZE VARIABLE
STORE .T. TO ADVIS
STORE 6 TO SCLINE
STORE 6 TO PRLINE
STORE 1 TO CNT
* CLEAR THE SCREEN
CLEAR
* CHECK FOR OUTPUT CHOICE
DO OUTCHOIC
SET DEVICE TO SCREEN
* OPEN DATABASE FILE
USE B:MASTER
* GET ADVISOR NAME
CLEAR
STORE SPACE(20) TO VADVI
54

5.1 SAY "PLEASE ENTER THE ADVISOR NAME"
6.1 SAY "THE NAME MUST BE ENTERED AS IT WAS DURING DATABASE CONSTRUCTION"
8.1 SAY "ADVISOR NAME : " GET VADV
READ
* WRITE HEADING
CLEAR
3.1 SAY "ADVISOR : " + VADV
6.50 SAY DATE()
2.3 SAY "LIST OF STUDENTS UNDER THE SAME ADVISOR"
4.5 SAY "LAST NAME"
4.27 SAY "FIRST NAME"
4.54 SAY "S.S. #"
* SET PRINTER/SCREEN OUTPUT FORM
IF SCREFLAG=1
STORE 22 TO MAXSL
* SET THE POINTER ON THE FIRST RECORD
GOTO TOP
LOCATE FOR ADVI = VADV
IF EOF()
   CLEAR
   10.1 SAY "THE NAME "+ VADV +"IS NOT ON FILE"
   WAIT
   RETURN
ENDIF
DO WHILE .NOT. EOF()
   SCLINE,S SAY LNAME + " " + FNAME + " " + SSNO
   SCLINE=SCLINE+1
   CNT=CNT+1
   IF SCLINE > MAXSL
      STORE 1 TO SCLINE
      WAIT
      CLEAR
   ENDIF
   CONTINUE
ENDDO
CNT=CNT-1
?
" TOTAL NUMBER OF STUDENTS=",CNT
?
STORE 0 TO SCREFLAG
WAIT
RETURN
ENDIF
* THE OUTPUT TO PRINTER
SET DEVICE TO PRINT
SET PRINT ON
STORE 50 TO MAXPLINE
* SET THE POINTER ON THE FIRST RECORD
GOTO TOP
LOCATE FOR ADVI = VADV
IF EOF()
   SET DEVICE TO SCREEN
   CLEAR
   10.1 SAY "THE NAME "+ VADV +"IS NOT ON FILE"
   WAIT
RETURN
ENDIF
* WRITE HEADING
CLEAR
@ 1,3 SAY "ADVISOR : " + ADVI
@ 1,50 SAY DATE()
@ 2,3 SAY "LIST OF STUDENTS UNDER THE SAME ADVISOR"
@ 4,5 SAY "LAST NAME"
@ 4,27 SAY "FIRST NAME"
@ 4,54 SAY "S.S. #"
DO WHILE .NOT. EOF()
   @ PRLINE,5 SAY LNAME+ " " +FNAME+ " " + SSNO
   PRLINE=PRLINE+1
   CNT=CNT+1
   IF PRLINE > MAXPLINE
      STORE 2 TO PRLINE
      EJECT
   ENDIF
ENDDO
CNT=CNT-1
?
? " TOTAL NUMBER OF STUDENTS=",CNT
?
SET DEVICE TO SCREEN
SET PRINT OFF
STORE 0 TO SCREFLAG
RETURN
PROCEDURE FGCLASS
* = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
* PROCEDURE TO PRINT OR DISPLAY
* THE NAME OF ALL THE STUDENTS THAT ARE IN THE SAME YEAR OF SCHOOL
* = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
* INITIALIZE VARIABLE
STORE 6 TO SCLINE
STORE 6 TO PRLINE
STORE 1 TO CNT
* CLEAR THE SCREEN
CLEAR
* CHECK FOR OUTPUT CHOICE
DO OUTCHOIC
SET DEVICE TO SCREEN
* OPEN DATABASE FILE
USE B:MASTER
DO WHILE .T.
* GET CLASSIFICATION
CLEAR
STORE SPACE(10) TO VCLASS
@ 5,1 SAY "PLEASE ENTER THE CLASSIFICATION (FRESHMAN ,etc.)"
@ 6,1 SAY "CLASSIFICATION MUST BE ENTERED AS DURING DATABASE CONSTRUCTION"
@ 8,1 SAY "CLASSIFICATION : " GET VCLASS
READ
* WRITE HEADING
CLEAR
@ 1,50 SAY DATE()
2.1 SAY "LIST OF ALL " + VCLASS + " STUDENTS . "
3.4,5 SAY "LAST NAME"
3.4,27 SAY "FIRST NAME"
3.4,54 SAY "S.S. #"
* SET PRINTER/SCREEN OUTPUT FORM
IF SCREFLAG=1
STORE 22 TO MAXSL
* SET THE POINTER ON THE FIRST RECORD
GOTO TOP
Locate FOR CLASS= VCLASS
IF EOF()
   CLEAR
   @ 10,1 SAY "PLEASE CHECK SPELLING AND CORRECTNESS OF INPUT"
   WAIT
   LOOP
ENDIF
DO WHILE .NOT. EOF()
   @ SCLINE,5 SAY LNAME + " " + FNAME + " " + SSNO
   SCLINE= SCLINE+1
   CNT=CNT+1
   IF SCLINE >MAXSL
      STORE 1 TO SCLINE
      WAIT
      CLEAR
      ENDIF
   CONTINUE
ENDDO
CNT=CNT-1
? " TOTAL NUMBER OF STUDENTS=",CNT
?
WAIT
RETURN
ENDIF
* THE OUTPUT TO PRINTER
SET DEVICE TO PRINT
SET PRINT ON
STORE 50 TO MAXPLINE
* SET THE POINTER ON THE FIRST RECORD
GOTO TOP
Locate FOR CLASS=VCLASS
IF EOF()
   SET DEVICE TO SCREEN
   CLEAR
   @ 10,1 SAY " PLEASE CHECK SPELLING AND CORRECTNESS OF INPUT"
   WAIT
   LOOP
ENDIF
* WRITE HEADING
CLEAR
2.1,50 SAY DATE()
3.2,1 SAY "LIST OF ALL " + VCLASS + " STUDENTS . "
4.5 SAY "LAST NAME"
4.27 SAY "FIRST NAME"
4.54 SAY "S.S. #"
DO WHILE .NOT. EOF()
  @ PRLINE,5 SAY LNAME+" "+FNAME+" "+SSNO
  PRLINE=PRLINE+1
  CNT=_CNT+1
  IF PRLINE > MAXPLINE
    STORE 2 TO PRLINE
    EJECT
    CONTINUE
  ENDDO
  ? "TOTAL NUMBER OF STUDENTS=" ,CNT
? SET DEVICE TO SCREEN
SET PRINT OFF
RETURN
ENDDO
RETURN

PROCEDURE SORTALL
* PROCEDURE TO PRINT A LIST OF ALL STUDENTS IN THE DATABASE
* SORTED BY LAST NAME.
* OPEN THE DATABASE FILE
USE B:MASTER
CLEAR
@ 20,1 SAY "PLEASE WAIT ,NAMES ARE BEING SORTED"
* SORT
SORT TO B:SORTLIST ON LNAME
CLOSE DATABASES
USE B:SORTLIST
CLEAR
@ 20,1 SAY "NAMES ARE BEING PRINTED"
* TURN PRINTER ON/INITIALIZE
SET DEVICE TO PRINT
SET PRINT ON
* PRINT HEADING
@ 1,50 SAY DATE()
@ 2,11 SAY "++++ STUDENT LIST -- COMPUTER ENGINEERING ++++
@ 4,5 SAY "LAST NAME"
@ 4,27 SAY "FIRST NAME"
@ 4,54 SAY "S.S. #"
STORE 6 TO PRLINE
STORE 1 TO CNT
* START FROM THE FIRST RECORD
GOTO TOP
DO WHILE .NOT. EOF()
* PRINT NAMES
  @ PRLINE,5 SAY LNAME+" "+FNAME+" "+SSNO
  SKIP 1
  CNT=_CNT+1
  PRLINE=PRLINE+1
  IF PRLINE>50
    EJECT
    STORE 3 TO PRLINE
  ENDIF
ENDIF
Loop
ENDDD
CNT=CNT-1
? " TOTAL NUMBER OF STUDENTS =",CNT
? SET PRINT OFF
SET DEVICE TO SCREEN
RETURN

PROCEDURE GETVARI
* RESTORE VARIABLES WITH THE CONTENT OF THE
* CORRESPONDING FIELDS
STORE SSNO TO VSSNO
STORE LNAME TO VLNAME
STORE FNAME TO VFNAME
STORE ADDR TO VADDR
STORE CITY TO VCITY
STORE STATE TO VSTATE
STORE ZIP TO VZIP
STORE PHON TO VPJN
STORE ADVI TO VADVI
STORE MAJOR TO VMAJOR
STORE CLASS TO VCLASS
STORE ENC_1101 TO VENC_1101
STORE ENC_1102 TO VENC_1102
STORE SPC_1014 TO VSPC_1014
STORE EUH_2000 TO VEUH_2000
STORE EUH_2001 TO VEUH_2001
STORE AMH_2010 TO VAMH_2010
STORE AMH_2020 TO VAMH_2020
STORE HUM_2211 TO VHUM_2211
STORE HUM_2230 TO VHUM_2230
STORE POS_2041 TO VPPOS_2041
STORE PSY_2013 TO VPSY_2013
STORE SYG_2000 TO VSYG_2000
STORE ANT_2003 TO VANT_2003
STORE COP_3215 TO VCOP_3215
STORE EGN_3310 TO VEGN_3310
STORE EGN_3311C TO VEGN_3311C
STORE EGN_3313 TO VEGN_3313
STORE EGN_3313C TO VEGN_3313C
STORE EGN_3315 TO VEGN_3315
STORE EGN_3315C TO VEGN_3315C
STORE EGN_3317 TO VEGN_3317
STORE EGN_3317C TO VEGN_3317C
STORE EGN_3319 TO VEGN_3319
STORE EGN_3319C TO VEGN_3319C
STORE EGN_3321 TO VEGN_3321
STORE EGN_3321C TO VEGN_3321C
STORE EGN_3323 TO VEGN_3323
STORE EGN_3323C TO VEGN_3323C
STORE EGN_3325 TO VEGN_3325
STORE EGN_3325C TO VEGN_3325C
PROCEDURE F'REP
* THIS PROCEDURE PRINTS A STUDENT RECORD (HARD COPY)
*--------------------------------------------------------
SET DEVICE TO PRINT
SET PRINT ON
STORE DATA_STRUC TO VDATASTRUCT
STORE APPORT_ELEC TO VAPPORELEC
STORE DESIG_ELEC TO VDESIGELEC
STORE MICELL_1 TO VMICELL1
STORE MICELL_2 TO VMICELL2
STORE MICELL_3 TO VMICELL3
STORE NOTE1 TO VNOTE1
STORE NOTE2 TO VNOTE2
RETURN

* ----------------- ----------------------------------------
STORE EGN_3343 TO VEGN3343
STORE EGN_3353C TO VEGN3353C
STORE EGN_33613 TO VEGN33613
STORE EGN_33703 TO VEGN33703
STORE EGN_33863 TO VEGN33863
STORE EGN_34704 TO VEGN34704
STORE STA_3032 TO VSTA3032
STORE PHY_3101 TO VPHY3101
STORE BSC_1020C TO VBSC1020C
STORE BSC_1030C TO VBSC1030C
STORE GEO_1200 TO VGED1200
STORE GEO_3370 TO VGED3370
STORE GLY_1000 TO VGLY1000
STORE ENC_4634 TO VENC4634
STORE EGN_4624 TO VEGN4624
STORE EEL_3342 TO VEEL3342
STORE EEL_4701 TO VEEL4701
STORE ECM_4504 TO VECM4504
STORE ECM_4804 TO VECM4804
STORE EEL_4702 TO VEEL4702
STORE ECM_4343 TO VECM4343
STORE ECM_4124 TO VECM4124
STORE ECM_4411 TO VECM4411
STORE DATA_STRUC TO VDATASTRUCT
STORE APPORT_ELEC TO VAPPORELEC
STORE DESIG_ELEC TO VDESIGELEC
STORE MICELL_1 TO VMICELL1
STORE MICELL_2 TO VMICELL2
STORE MICELL_3 TO VMICELL3
STORE NOTE1 TO VNOTE1
STORE NOTE2 TO VNOTE2
RETURN
© ROW, 1 SAY "ADVISOR NAME : " + VADVI
© ROW=ROW+1
© ROW, 1 SAY "MAJOR : " + VMAJOR
© ROW, 40 SAY "CLASSIFICATION : " + VCLASS
© ROW=ROW+1
© ROW, 1 SAY "ENC_1101 : " + VENC1101
© ROW, 25 SAY "ENC_1102 : " + VENC1102
© ROW, 50 SAY "SPC_1014 : " + VSPC1014
© ROW=ROW+1
© ROW, 1 SAY "EUM_2000 : " + VEAUM2000
© ROW, 25 SAY "EUM_2001 : " + VEAUM2001
© ROW, 50 SAY "AMH_2010 : " + VAMH2010
© ROW=ROW+1
© ROW, 1 SAY "AMM_2020 : " + VAMM2020
© ROW, 25 SAY "HUM_2211 : " + VHUM2211
© ROW, 50 SAY "HUM_2230 : " + VHUM2230
© ROW=ROW+1
© ROW, 1 SAY "POS_2041 : " + VPOS2041
© ROW, 25 SAY "PSY_2013 : " + VPSY2013
© ROW, 50 SAY "SYG_2000 : " + VSYG2000
© ROW=ROW+1
© ROW, 1 SAY "ANT_2003 : " + VANT2003
© ROW, 25 SAY "COF_3215 : " + VCOF3215
© ROW, 50 SAY "EGN_3210 : " + VEGN3210
© ROW=ROW+1
© ROW, 1 SAY "EGN_1111C : " + VEGN1111C
© ROW, 25 SAY "ECO_2013 : " + VECO2013
© ROW, 50 SAY "CHS_1440 : " + VCHS1440
© ROW=ROW+1
© ROW, 1 SAY "PHY_3048 : " + VPHY3048
© ROW, 25 SAY "PHY_3049 : " + VPHY3049
© ROW, 50 SAY "PHY_3049L : " + VPHY3049L
© ROW=ROW+1
© ROW, 1 SAY "MAC_3311 : " + VMAC3311
© ROW, 25 SAY "MAC_3312 : " + VMAC3312
© ROW, 50 SAY "MAC_3313 : " + VMAC3313
© ROW=ROW+1
© ROW, 1 SAY "MAP_3302 : " + VMAP3302
© ROW, 25 SAY "EGN_3311 : " + VEGN3311
© ROW, 50 SAY "EGN_3321 : " + VEGN3321
© ROW=ROW+1
© ROW, 1 SAY "EGN_3331C : " + VEGN3331C
© ROW, 25 SAY "EGN_3373 : " + VEGN3373
© ROW, 50 SAY "EGN_3375C : " + VEGN3375C
© ROW=ROW+1
© ROW, 1 SAY "EGN_3343 : " + VEGN3343
© ROW, 25 SAY "EGN_3353C : " + VEGN3353C
© ROW, 50 SAY "EGN_3613 : " + VEGN3613
© ROW=ROW+1
© ROW, 1 SAY "EGN_3703 : " + VEGN3703
© ROW, 25 SAY "EGN_3763 : " + VEGN3763
© ROW, 50 SAY "EGN_3704 : " + VEGN3704
© ROW=ROW+1
© ROW, 1 SAY "STA_3032 : " + VSTA3032
© ROW, 25 SAY "PHY_3101 : " + VPHY3101
PROCEDURE FGCORS

* PROCEDURE TO PRINT OR DISPLAY THE NAME OR THE TOTAL *
* NUMBER OF ALL STUDENTS THAT HAVE TAKEN ATHE SAME COURSE. *
* THIS PROCEDURE WILL CHECK EVERY COURSE FIELD NAME IN THE *
* MASTER DATABASE TO FIND THE COURSE FIELD REQUESTED BY THE *
* USER, THAN IT CHECKS EVERY RECORD UNDER THAT COURSE *
* FIELD FOR EXISTANCE OF AN ENTRY(DATA). *

CLEAR
* INITIALIZE COUNTERS AND VARIABLES
STORE .T. TO P1
STORE .T. TO P2
STORE 22 TO MAXSL
STORE 50 TO MAXLINE
STORE 6 TO SCLINE

@ ROW,50 SAY "BSC_1020C : " + VBSC1020C
ROW=ROW+1
@ ROW,1 SAY "BSC_1030C : " + VBSC1030C
@ ROW,50 SAY "GEO_1200 : " + VGED1200
@ ROW,50 SAY "GEO_3370 : " + VGED3370
@ ROW,1 SAY "GLY_1000 : " + VGLY1000
@ ROW,25 SAY "ENC_4634 : " + VENC4634
@ ROW,50 SAY "ENC_4624 : " + VENC4624
@ ROW,1 SAY "EEL_3342 : " + VEEL3342
@ ROW,25 SAY "EEL_4701 : " + VEEL4701
@ ROW,50 SAY "ECM_4504 : " + VECM4504
@ ROW,25 SAY "EEL_4702 : " + VEEL4702
@ ROW,50 SAY "ECM_4411 : " + VECM4411
@ ROW,1 SAY "DATA_STRUCTURE : " + VDATASTRUC
@ ROW,25 SAY "APPFORED_ELECTIVE : " + VAPFORELEC
@ ROW,35 SAY "OTHER_COURSES 1 : " + VMICELL1
@ ROW,35 SAY "OTHER_COURSES 2 : " + VMICELL2
@ ROW,35 SAY "OTHER_COURSES 3 : " + VMICELL3
@ ROW,1 SAY "NOTE : " + VNOTE1
@ ROW,1 SAY "NOTE : " + VNOTE2

SET DEVICE TO SCREEN
SET PRINT OFF
STORE 0 TO FRFLAG
RETURN

* =================================================================================*
STORE 6 TO PRLINE
STORE 1 TO CNT
STORE SPACE(1) TO ANS
* OPEN DATABASE FILE
USE B:MASTER
* GET COURSE NAME
DO WHILE P1
CLEAR
STORE SPACE(9) TO VCORS
@ 2,1 SAY "PLEASE ENTER THE PREFIX AND THE NUMBER OF THE COURSE."
@ 3,1 SAY "THIS COURSE MUST BE ONE OF THE COURSES IN THE DATA BASE."
@ 4,1 SAY "IF NOT, AN ERROR MESSAGE WILL BE DISPLAYED."
@ 5,1 SAY "PLEASE RESPOND WITH Y TO THE ERROR MESSAGE, THEN TYPE "
@ 6,1 SAY "DO B:MENU. THE MAIN MENU WILL BE DISPLAYED"
@ 8,1 SAY "COURSE : " GET VCORS PICTURE "AAA_9999A"
@ 9,1 SAY "ALL CORRECT ? (Y/N) " GET ANS
READ IF ANS<>"Y"
    STORE SPACE(1) TO ANS
    LOOP
ENDIF
STORE .F. TO P1
ENDDO
DO WHILE P2
CLEAR
@ 9,5 SAY "L ---- FOR LIST AND TOTAL NUMBER"
@ 10,5 SAY "N ---- FOR TOTAL NUMBER ONLY"
?
?
WAIT "SELECT OPTION : " TO OP
* CHECK VALID OPTION
DO CASE
CASE OF="L"
STORE 1 TO LISTFLAG
STORE 0 TO NUMBFLAG
STORE .F. TO P2
CASE OF="N"
STORE 1 TO NUMBFLAG
STORE 0 TO LISTFLAG
STORE .F. TO P2
OTHERWISE
@ 15,3 SAY "PLEASE RESPOND WITH L OR N"
WAIT
LOOP
ENDDO
STORE .F. TO P2
* DETERMINE OUTPUT DEVICE
DO OUTCHOIC
* SET POINTER TO FIRST RECORD
GOTO TOP
* FOLLOWING IF STATEMENTS DETERMINE THE FIELD OF SEARCH
IF VCORS ="ENC_1101"
    LOCATE FOR ENC_1101 <> ""
ENDIF
IF VCorS = "ENC_1102"
   LOCATE FOR ENC_1102 <> "
ENDIF
IF VCorS = "SPC_1014"
   LOCATE FOR SPC_1014 <> "
ENDIF
IF VCorS = "EUH_2000"
   LOCATE FOR EUH_2000 <> "
ENDIF
IF VCorS = "EUH_2001"
   LOCATE FOR EUH_2001 <> "
ENDIF
IF VCorS = "AMH_2010"
   LOCATE FOR AMH_2010 <> "
ENDIF
IF VCorS = "AMH_2020"
   LOCATE FOR AMH_2020 <> "
ENDIF
IF VCorS = "HUM_2211"
   LOCATE FOR HUM_2211 <> "
ENDIF
IF VCorS = "HUM_2230"
   LOCATE FOR HUM_2230 <> "
ENDIF
IF VCorS = "POS_2041"
   LOCATE FOR POS_2041 <> "
ENDIF
IF VCorS = "FSY_2013"
   LOCATE FOR FSY_2013 <> "
ENDIF
IF VCorS = "SYG_2000"
   LOCATE FOR SYG_2000 <> "
ENDIF
IF VCorS = "ANT_2003"
   LOCATE FOR ANT_2003 <> "
ENDIF
IF VCorS = "COP_3215"
   LOCATE FOR COP_3215 <> "
ENDIF
IF VCorS = "EGN_3210"
   LOCATE FOR EGN_3210 <> "
ENDIF
IF VCorS = "EGN_1111C"
   LOCATE FOR EGN_1111C <> "
ENDIF
IF VCorS = "ECO_2013"
   LOCATE FOR ECO_2013 <> "
ENDIF
IF VCorS = "CHS_1440"
   LOCATE FOR CHS_1440 <> "
ENDIF
IF VCorS = "PHY_3048"
   LOCATE FOR PHY_3048 <> "
ENDIF
IF VCorS = "PHY_3049"
ENDIF
IF VCORS = "BSC_1020C"
   LOCATE FOR BSC_1020C <> " "
ENDIF
IF VCORS = "BSC_1030C"
   LOCATE FOR BSC_1030C <> " "
ENDIF
IF VCORS = "GEO_1200"
   LOCATE FOR GEO_1200 <> " "
ENDIF
IF VCORS = "GEO_3370"
   LOCATE FOR GEO_3370 <> " "
ENDIF
IF VCORS = "GLY_1000"
   LOCATE FOR GLY_1000 <> " "
ENDIF
IF VCORS = "ENC_4634"
   LOCATE FOR ENC_4634 <> " "
ENDIF
IF VCORS = "EGN_4624"
   LOCATE FOR EGN_4624 <> " "
ENDIF
IF VCORS = "EEL_3342"
   LOCATE FOR EEL_3342 <> " "
ENDIF
IF VCORS = "EEL_4701"
   LOCATE FOR EEL_4701 <> " "
ENDIF
IF VCORS = "ECM_4504"
   LOCATE FOR ECM_4504 <> " "
ENDIF
IF VCORS = "ECM_4804"
   LOCATE FOR ECM_4804 <> " "
ENDIF
IF VCORS = "EEL_4702"
   LOCATE FOR EEL_4702 <> " "
ENDIF
IF VCORS = "ECM_4343"
   LOCATE FOR ECM_4343 <> " "
ENDIF
IF VCORS = "ECM_4124"
   LOCATE FOR ECM_4124 <> " "
ENDIF
IF VCORS = "ECM_4411"
   LOCATE FOR ECM_4411 <> " "
ENDIF
* WRITE HEADING
CLEAR
3 1,50 SAY DATE()
3 2,1 SAY "ALL STUDENTS THAT HAVE TAKEN " + VCORS + " ."
3 4,5 SAY "LAST NAME"
3 4,27 SAY "FIRST NAME"
3 4,54 SAY "S.S. #"
DO WHILE .NOT. EOF()
* CHECK FOR LIST OR TOTAL NUMBER
4.54 SAY "S.S. #"
DO WHILE .NOT. EOF()
  + CHECK FOR LIST OR TOTAL NUMBER
  IF LISTFLAG=1
    IF SCLFLAG=1
      ° SCLINE,5 SAY LNAME + " " + FNAME + " " + SSNO
      SCLINE=SCLINE+1
      CNT=CNT+1
      IF SCLINE>MAXSL
        STORE 1 TO SCLINE
        WAIT
        CLEAR
      ENDIF
    ENDIF
  IF PRFLAG=1
    SET PRINT ON
    SET DEVICE TO PRINT
    ° PRLINE,5 SAY LNAME + " " + FNAME + " " + SSNO
    PRLINE=PRLINE+1
    CNT=CNT+1
    IF PRLINE>MAXPLINE
      STORE 2 TO PRLINE
      EJECT
    ENDIF
  ENDIF
  ENDIF
  IF NUMBFLAG=1
    CNT=CNT+1
  ENDIF
  CONTINUE
ENDDO
CNT=CNT-1
?
? " TOTAL NUMBER OF STUDENTS=",CNT
?
SET PRINT OFF
SET DEVICE TO SCREEN
WAIT
RETURN
67

IF LISTFLG=1
  IF SCREFLAG=1
    & SCLINE,5 SAY LNAME + " " + FName +" " + SSNO
    SCLINE=SCLINE+1
    CNT=CNT+1
  IF SCLINE>MAXSL
    STORE 1 TO SCLINE
    WAIT
    CLEAR
  ENDF
ENDIF
ENDIF
IF PRFLAG=1
  SET PRINT ON
  SET DEVICE TO PRINT
  & PRLINE,5 SAY LNAME +" " + FName +" " + SSNO
  PRLINE=PRLINE+1
  CNT=CNT+1
  IF PRLINE>MAXPLINE
    STORE 2 TO PRLINE
    EJECT
  ENDF
ENDIF
ENDIF
IF NUMBFLAG=1
  CNT=CNT+1
ENDIF
CONTINUE
ENDDO
CNT=CNT-1
?
" TOTAL NUMBER OF STUDENTS=" ,CNT
?
SET PRINT OFF
SET DEVICE TO SCREEN
WAIT
RETURN

*=======================================================================*

PROCEDURE FGCLCO
* PROCEDURE TO PRINT OR DISPLAY THE NAME OR THE TOTAL *
* NUMBER OF ALL STUDENTS THAT HAVE TAKEN THE SAME COURSE, *
* AND ARE IN THE SAME YEAR OF SCHOOL. *
* THIS PROCEDURE WILL CHECK THE CLASSIFICATION OF EACH *
* STUDENT IN THE DATA BASE FILE FOR THE GIVEN STATUS *
* (FRESHMAN, SOPHOMORE, etc.) OF THE STUDENT, IF A *
* STUDENT IS FOUND WITH THAT STATUS, THEN IT CHECKS FOR *
* THE ENTRY UNDER THE GIVEN COURSE NUMBER. *
*=======================================================================*

CLEAR
* INITIALIZE COUNTERS AND VARIABLES
STORE .T. TO P1
STORE .T. TO P2
STORE 22 TO MAXSL
STORE 50 TO MAXPLINE
STORE 6 TO SCLINE
STORE 6 TO PRLINE
STORE 1 TO CNT
STORE SPACE(1) TO ANS
* OPEN DATABASE FILE
USE B:MASTER
* GET COURSE NAME AND CLASSIFICATION
DO WHILE P1
CLEAR
STORE SPACE(10) TO VCLASS
STORE SPACE(9) TO VCORS
@ 2,1 SAY "PLEASE ENTER THE PREFIX AND THE NUMBER OF THE COURSE."
@ 3,1 SAY "THIS COURSE MUST BE ONE OF THE COURSES IN THE DATA BASE."
@ 4,1 SAY "IF NOT, AN ERROR MASSAGE WILL BE DISPLAYED. "
@ 5,1 SAY "PLEASE RESPOND WITH Y TO THE ERROR MASSAGE, THEN TYPE "
@ 6,1 SAY "DO B:MENU. THE MAIN MENU WILL BE DISPLAYED "
@ 8,1 SAY "COURSE : " GET VCORS PICTURE "AAA_9999A"
@ 10,1 SAY "PLEASE ENTER ONE OF THE FOUR CLASSIFICATIONS : "
@ 12,1 SAY "FRESHMAN   SOPHOMORE   JUNIOR   SENIOR"
@ 14,1 GET VCLASS
@ 20,1 SAY "ALL CORRECT ? (Y/N) " GET ANS
READ
IF ANS<>"Y"
STORE SPACE(1) TO ANS
LOOP
ENDIF
STORE .F. TO P1
ENDDO
DO WHILE P2
CLEAR
@ 9,5 SAY "L ---- FOR LIST AND TOTAL NUMBER"
@ 10,5 SAY "N ---- FOR TOTAL NUMBER ONLY"
?
WAIT "SELECT OPTION : " TO OP
* CHECK VALID OPTION
DO CASE
CASE OP="L"
STORE 1 TO LISTFLAG
STORE .F. TO P2
STORE 0 TO NUMBFLAG
CASE OP="N"
STORE 1 TO NUMBFLAG
STORE .F. TO P2
STORE 0 TO LISTFLAG
OTHERWISE
@ 15,3 SAY "PLEASE RESPOND WITH L OR N "
WAIT
LOOP
ENDCASE
ENDDO
* DETERMINE OUTPUT DEVICE
DO OUTCHOIC
* SET POINTER TO FIRST RECORD
GOTO TOP
* FOLLOWING IF STATEMENTS DETERMINE THE FIELD OF SEARCH
IF V CORS = "ENC_1101"
   LOCATE FOR CLASS=VCLASS .AND. ENC_1101 <> " "
ENDIF
IF V CORS = "ENC_1102"
   LOCATE FOR CLASS=VCLASS .AND. ENC_1102 <> " "
ENDIF
IF V CORS = "SPC_1014"
   LOCATE FOR CLASS=VCLASS .AND. SPC_1014 <> " "
ENDIF
IF V CORS = "EUH_2000"
   LOCATE FOR CLASS=VCLASS .AND. EUH_2000 <> " "
ENDIF
IF V CORS = "EUH_2001"
   LOCATE FOR CLASS=VCLASS .AND. EUH_2001 <> " "
ENDIF
IF V CORS = "AMH_2010"
   LOCATE FOR CLASS=VCLASS .AND. AMH_2010 <> " "
ENDIF
IF V CORS = "AMH_2020"
   LOCATE FOR CLASS=VCLASS .AND. AMH_2020 <> " "
ENDIF
IF V CORS = "HUM_2211"
   LOCATE FOR CLASS=VCLASS .AND. HUM_2211 <> " "
ENDIF
IF V CORS = "HUM_2230"
   LOCATE FOR CLASS=VCLASS .AND. HUM_2230 <> " 
ENDIF
IF V CORS = "POS_2041"
   LOCATE FOR CLASS=VCLASS .AND. POS_2041 <> " "
ENDIF
IF V CORS = "PSY_2013"
   LOCATE FOR CLASS=VCLASS .AND. PSY_2013 <> " "
ENDIF
IF V CORS = "SYG_2000"
   LOCATE FOR CLASS=VCLASS .AND. SYG_2000 <> " "
ENDIF
IF V CORS = "ANT_2003"
   LOCATE FOR CLASS=VCLASS .AND. ANT_2003 <> " "
ENDIF
IF V CORS = "COP_3215"
   LOCATE FOR CLASS=VCLASS .AND. COP_3215 <> " "
ENDIF
IF V CORS = "EGN_3210"
   LOCATE FOR CLASS=VCLASS .AND. EGN_3210 <> " "
ENDIF
IF V CORS = "EGN_1111C"
   LOCATE FOR CLASS=VCLASS .AND. EGN_1111C <> " "
ENDIF
IF V CORS = "ECO_2013"
   LOCATE FOR CLASS=VCLASS .AND. ECO_2013 <> " "
ENDIF
IF V CORS = "CHS_1440"
   LOCATE FOR CLASS=VCLASS .AND. CHS_1440 <> " "
ENDIF
IF V CORS = "PHY_3048"
ENDIF
IF VCORS = "PHY_3101"
   LOCATE FOR CLASS=VCLASS .AND. PHY_3101 <> " "
ENDIF
IF VCORS = "BSC_1020C"
   LOCATE FOR CLASS=VCLASS .AND. BSC_1020C <> " 
ENDIF
IF VCORS = "BSC_1030C"
   LOCATE FOR CLASS=VCLASS .AND. BSC_1030C <> " 
ENDIF
IF VCORS = "GEO_1200"
   LOCATE FOR CLASS=VCLASS .AND. GEO_1200 <> " 
ENDIF
IF VCORS = "GEO_3370"
   LOCATE FOR CLASS=VCLASS .AND. GEO_3370 <> " 
ENDIF
IF VCORS = "GLY_1000"
   LOCATE FOR CLASS=VCLASS .AND. GLY_1000 <> " 
ENDIF
IF VCORS = "ENC_4634"
   LOCATE FOR CLASS=VCLASS .AND. ENC_4634 <> " 
ENDIF
IF VCORS = "EGN_4624"
   LOCATE FOR CLASS=VCLASS .AND. EGN_4624 <> " 
ENDIF
IF VCORS = "EEL_3342"
   LOCATE FOR CLASS=VCLASS .AND. EEL_3342 <> " 
ENDIF
IF VCORS = "EEL_4701"
   LOCATE FOR CLASS=VCLASS .AND. EEL_4701 <> " 
ENDIF
IF VCORS = "ECM_4504"
   LOCATE FOR CLASS=VCLASS .AND. ECM_4504 <> " 
ENDIF
IF VCORS = "ECM_4804"
   LOCATE FOR CLASS=VCLASS .AND. ECM_4804 <> " 
ENDIF
IF VCORS = "EEL_4702"
   LOCATE FOR CLASS=VCLASS .AND. EEL_4702 <> " 
ENDIF
IF VCORS = "ECM_4343"
   LOCATE FOR CLASS=VCLASS .AND. ECM_4343 <> " 
ENDIF
IF VCORS = "ECM_4124"
   LOCATE FOR CLASS=VCLASS .AND. ECM_4124 <> " 
ENDIF
IF VCORS = "ECM_4411"
   LOCATE FOR CLASS=VCLASS .AND. ECM_4411 <> " 
ENDIF
* WRITE HEADING
CLEAR
@ 1.50 SAY DATE()
@ 2.1 SAY VCLASS + "STUDENTS THAT HAVE TAKEN " + VCORS + ","
@ 4.5 SAY "LAST NAME"
@ 4.27 SAY "FIRST NAME"
APPENDIX C

OUTPUT SAMPLES
OUTPUT OF PROCEDURE "FGADVI"

ADVISOR: DR. BAUER

LIST OF STUDENTS UNDER THE SAME ADVISOR

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>S.S. #</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMSTRONG</td>
<td>JAMES C</td>
<td>111-11-1111</td>
</tr>
<tr>
<td>BOSCH</td>
<td>THOMAS M</td>
<td>533-54-3533</td>
</tr>
<tr>
<td>COBBO</td>
<td>CHARLIE N</td>
<td>444-44-4444</td>
</tr>
</tbody>
</table>

TOTAL NUMBER OF STUDENTS: 3

07/15/85
OUTPUT OF PROCEDURE "FGCLCOR"

FRESHMAN STUDENTS THAT HAVE TAKEN EUH_2000  .  07/15/85

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>S.S. #</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMSTRONG</td>
<td>JAMES C</td>
<td>111-11-1111</td>
</tr>
<tr>
<td>HARISS</td>
<td>DAVID J</td>
<td>555-55-5555</td>
</tr>
</tbody>
</table>

TOTAL NUMBER OF STUDENTS= 2
OUTPUT OF PROCEDURE "FGCLAS"

LIST OF ALL SOPHOMORE STUDENTS.

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>S.S. #</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATWELL</td>
<td>ROBERT A</td>
<td>222-22-2222</td>
</tr>
<tr>
<td>COBB</td>
<td>CHARLIE N</td>
<td>444-44-4444</td>
</tr>
</tbody>
</table>

TOTAL NUMBER OF STUDENTS = 3

07/15/85
OUTPUT OF PROCEDURE "PREREP"

STUDENT RECORD
SOCIAL SECURITY NUMBER : 333-33-3333
LAST NAME : BOSCH
FIRST AND MIDDLE NAME : THOMAS M
STREET ADDRESS : 290 E LAKE MERY BLVD
CITY : LAKE MERY   STATE : FL
TELEPHONE NUMBER : 333-3333
ADVISOR NAME : DR. DAUER
MAJOR : COMPUTER ENG
CLASSIFICATION : JUNIOR

<table>
<thead>
<tr>
<th>COURSE</th>
<th>Credits</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENC_1101</td>
<td>3</td>
<td>U R CO A</td>
</tr>
<tr>
<td>ENC_1102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIH_2000</td>
<td>3</td>
<td>U R CO B</td>
</tr>
<tr>
<td>AMH_2020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POS_2041</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANT_2003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGN_1111C</td>
<td>3</td>
<td>U R CO B</td>
</tr>
<tr>
<td>ECO_2013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHY_3048</td>
<td>3</td>
<td>U R IN</td>
</tr>
<tr>
<td>MAC_3311</td>
<td>3</td>
<td>U R CO A</td>
</tr>
<tr>
<td>MAP_3302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGN_3331C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGN_3333C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGN_3703</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STA_3032</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSC_1030C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GYL_1090</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EEL_3342</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECM_4804</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECM_4124</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECG_4411</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DATA_STRUCTURE :

APPORROVED ELECTIVE :

DESIGN_ELECTIVE :

NOTE :

NOTE :
APPENDIX D

BACKUP OF THE DATA BASE/
RECOVERY OF THE DATA BASE
Backup of the Data Base

Backing up the data base is a precautionary measure that is taken to guard the information in the data base against a partial or total loss. The user must backup the student tracking system's data base (master) each time after a successful modification of the data base by performing the following steps:

1. Exit from the student tracking system through the menu option
2. Insert the backup disk in Drive B (assuming the disk containing the data base is in Drive A)
3. Type COPY B:MASTER.DBF A:

Recovery of the Data Base

In case of the loss of the information in the original data base, the following steps must be taken to recover the information from the backup copy:

1. When DOS system prompt (A>) is displayed, insert the backup disk in Drive A and the original disk in Drive B.
2. Type COPY A:MASTER.DBF B:

Once the copy process is completed, follow the start-up process using the original disk. Any operation which was in-progress when the loss of data occurred must be started over.

NOTE: It is recommended that the student tracking system disk be backed up before using.
APPENDIX E

GENERAL USER INSTRUCTIONS
System Start-Up

In order to use the student tracking system, dBASE III data base manager software package is required. This software package must be installed according to the dBASE III manual instructions for the IBM personal computer. To begin the student tracking system, the following steps must be followed:

1. Use DOS 2.0 (or higher) to boot up the machine
2. Place the dBASE III Disk #1 in Drive A (left) and type DBASE
3. Follow the instructions until dBASE III prompt (.) appears on the screen
4. Insert the disk containing "MASTER" data base and programs of student tracking system in Drive B (right)
5. Type DO B:MENU

NOTE: B: indicates a machine with two 360K floppy disk drives.

The student tracking system will take control and the user has to only select options from the menus.

Entering Data

The system provides the user with the width of each data filed on the screen by displaying them in different densities of screen color. The entry process for all of the data elements must start from the first character of the field. Until the user becomes familiar with the structure of the course's data element, he/she must refer to the Student Record: Structure and Contents chapter of this research report. The last two fields of the data base "MASTER" (NOTE 1 and NOTE 2) can contain any necessary note that is wished by the user.
Exit from the System

The exit from the system should be through a menu exit option to prevent any unfinished operation which might cause any loss of data. However, the ESC key can be used to return to dBASE III dot prompt. In case the ESC key is used during reading or writing of the disk operation, that operation must be repeated.

Generally, the student tracking system is a user-oriented system and efforts have been made to make it user-friendly and easy to use.
REFERENCES


