Programming Standards:

A Tool for Increased Programmer Productivity

Delores R. Payne
Programmer/Analyst
The ENI Companies

Wayne E. Holt
Director, Computer Services
Whitman College

The Orlando Meeting of the HP3000 Users Group

April 1981

Key words: Standards, COBOL, Programming, Productivity

I. Overview

II. The Whitman Approach

III. Conclusions

Supplement. COBOL Coding Standards

Appendix

I Overview

We are, according to the September special edition of ComputerWorld, entering the "Software Decade", a decade that will be marked by the passage of layered software technologies, and replaced by integrated methods that vastly increase productivity and simplify use of the computer.

An ambitious vision of the future to be sure. In the HP3000 world, precompilers, code-generators, ad-hoc report writers, procedural sub-languages, and the like are already appearing on the market. Certainly, this lends credence to the vision of the 1980's. But is that era truly upon us, and will programming as we know it quickly disappear from the face of the earth? We doubt it.

All too often, one of the decisions that upper management will dictate to the DP establishment is, "Thou shalt develop all programs in XXXXX", where XXXXX is frequently COBOL, RPG, or Fortran. They do this in a non-spiteful way, since they believe that they are looking out for the best interests of the company.

After all, COBOL programmers are, for instance, "easy" to obtain on the open market. If your software investment has been in COBOL, then you stand a good chance of keeping it running with people who are "off the street" if a real crisis erupts.

As for the "new software productivity tools", a frequent attitude seems to be, "too new", and, "not as easy or productive as they are advertised to be". Now, we won't debate the merit or spirit of these arguments in this paper - the situation is a fact, and should be coped with intelligently if possible. In fact, coping with this situation is one purpose of this paper.

The title of the paper would have you believe that a programmer productivity tool has been within our grasp since the beginning. That tool is the often maligned creature, the "programming standard". Almost everyone will agree that standards are important, but few can point to a real implementation that actually worked. In fact, many will argue that standards are impossible to create, maintain, or police, and are therefore counterproductive. We take umbrage with this line of reasoning. As will be explained in Section II, the situation at Whitman dictated a serious look at this issue. A comprehensive, 30-month experiment has led us to the conclusion that programming standards, regardless of style, will improve quality and quantity of code, and lower overall costs of development.

The Whitman College Computer Center has been in operation since July of 1977, when the College recognized the need for modern data processing in both the Academic and Administrative areas. The decision was not lightly considered. At that point, one office of the Administration had an NCR 101 that was meagerly shared with several other offices and one Academic department had an IBM 1130 that also served the sciences in a small way.

The HP3000 was purchased with the idea of consolidating services and eliminating old equipment. Being a small college, with a very conservative fiscal policy, a deliberate effort was made to control program development from the very beginning. Vendor-supplied application software was judged inadequate at that time and a decision was made to develop a major area of the Administration "in-house".

In order to deflate the total overall cost of such a decision, a variety of techniques were adopted to expedite coding and assure quality. The results, as evaluated after 30 months, were even better than expected. In summary, a time-honored industry standard of 4 lines of code per hour was decimated by comparison to the effective coding rate of 34 lines of code per hour that was achieved by our staff.

As noted, there were several techniques utilized to achieve this rate. However, the dominant factor was the adoption, at a very early stage, of a set of programming standards. In the four years of the Center's existance, these standards have grown and matured. In other words, they have changed. Yet, this change has not caused the development effort to speed up or slow down. Thus our statement to the effect that having standards is more important than what is in the standards. As a measure of the universality of such a set of standards, ours are distributed to a dozen other sites in the Pacific Northwest on a regular basis.

II The Whitman Approach

Before details of the project at Whitman can be explained, a bit of background on the environment should be related. The College is a four-year, liberal arts institution founded in a very conservative American tradition. This conservatism is pervasive from finances to curriculum. Thus, a project of the magnitude that will be presented was not undertaken lightly.

Some of the constraints on the project are unique to the academic environment. Salaries, for instance, run 30-40% below the average in most metropolitan business sectors. The programming staff at the Computer Center has been characterized as both young and without experience. In context, this is an accurate assessment; it would be erroneous to deduce that these two characteristics imply a lack of either skill or finesse in their chosen profession, however.

For the most part, the staff is, in fact, fairly untrained when hired by the College. They are picked for the job based upon potential rather than experience, since the College is not competitive with industry in the job market and can rarely attract experienced people. Criteria for selection include above-average intelligence, high self-motivation, and goal orientation. So, while the staff may have obvious drawbacks, it nonetheless is primed to produce an above-average quality of program code within minimum time constraints.

In a way, this has been an asset rather than a liability. Staff training has little wasted motion, and the integration into the shop is sped by having no pre-conceived notions and habits to be altered to fit within the College's operating philosophy. Strong standards make such training possible.

In 1973, Datamation published an article, "Chief-Programmer Team for the New York Times", in which it was indicated that about 29 lines of code per manday was an optimum rate for COBOL coding in a batch environment. This is slightly less than 4 lines of code per manhour. This rate has been reaffirmed since then, but always in the batch world.

There are no reputable studies for development on interactive machines, such as the HP3000. There are many issues hidden in the batch vs. interactive environment that make our study invalid as a scientific experiment. However, we will attempt to enumerate the various factors and let you decide which is the most significant.

Many a manager has chewed out the programmer caught keypunching his own cards, regardless of how persuasively the programmer argued that "it was faster to do it myself", "we aren't paying you to keypunch!" is the usual opening line. However, this attitude just doesn't pay when carried over to the interactive environment. At Whitman, this approach was rejected in favor of having a terminal for every person developing code.

In the batch environment, many sharp programmers will duplicate program cards in order to "cut and paste" new programs quickly. In the interactive environment, text editors carry this art to new heights. Well-written programs are ideal templates for building similar software. This not only increases productivity, it also helps train new programmers in proper programming techniques.

This leads of course to programming standards. If everyone follows basically the same rules in designing and building software, then communication among the programmers is greatly facilitated for both training and maintenance.

Communication is a vital link in the development process. A very interesting study in the March 1980 issue of Datamation, "Software Manpower Costs: A Model", states that one programmer working 60 hours a week can complete a project in the same calendar time as two others, but at three-quarters the cost. A bold claim, but one that has some basis in fact. Communication between humans slows down productivity! The more people on a project, the slower each increment of progress for each person. Good standards can bridge the gap in communications. If everyone talks the same lingo, then the need for redundant communication is eliminated.

Whitman has in fact also subscribed to the "work longer" theory in a production mode. This is a further complication in evaluating the reason for rate of code development.

In Figure 1, a breakout is presented for program development between July 1977 and December 1979. During that period, 289 programs were developed in 9595 manhours, consisting of 328,305 lines of program code. This is an average of 34 lines per manhour, or 274 lines per manday. The 30 months were equivalent to 4.6 manyears of effort. Each of the techniques outlined above were utilized and contributed to the rates achieved.

Whitman College Computer Services

Software Development July 1977-December 1979

	#PROG	#LINES	#HOURS
ADMISSIONS OFFICE			
AD Admissions System	43	33896	980.0
COMPUTER CENTER			
BG Beacon/Guardian	09	13125	338.5
JA Job Accounting	09	11210	292.5
UT Utilities	33	10693	233.5
FINANCIAL AID OFFICE			
FA Financial Aid	14	19542	904.5
FINANCIAL DEVELOPMENT			
AL Alumni Affairs	13	18217	428.5
FD Financial Dev			204.0
GR Gift Records	15	27936	810.5
ML Central Mailing	10	10258	229.0
HOUSING OFFICE			
SH Student Housing	28	25934	727.0
PROVOSTS OFFICE			
BD Budget Status	10	11065	609.0
ER Employee Records			1017.0
REGISTRAR			
CG Class Grading	22	29199	818.5
CR Class Records	45	59335	2029.1
SR Student Records	29	51013	994.0
TREASURERS OFFICE			
SC Securities	08	6885	400.0
< TOTAL >	289	328305	9594.6

Figure 1

III Conclusions

The project at Whitman took advantage of several techniques for increasing programmer productivity, including:

- o programmers have their own terminal for online text editing of source code,
- o cut and paste template techniques were perfected within certain COBOL standards,
- o programmers were encouraged to spend more contiguous time on projects, and
- o comprehensive programming standards were established early in the project.

Without a doubt, programming standards played a vital role in the success of the other three techniques. The standards defined the boundaries of programmer communication, and set the stage for role models and templates to be established.

The standards created for COBOL have been carried over to other languages, such as SPL and Fortran. In the early months of the project, cut and paste models were "whatever could be found" that worked out well previously. Now, formal standard skeletons have been created for all types of standard programs. These skeletons ensure that all code is developed uniformly, and cuts down on re-inventing programming methods.

These techniques can increase productivity in virtually any shop. They meet the needs of those managers who must intone the gospel of the "standard language" to data processing, and assure a more reasonable expenditure for code generated.

When coupled with the new productivity tools, projects developed in the coming decade will certainly live up to the bright promises written in the current journals. At Whitman, this marriage is occurring today. We continue to develop COBOL code for our primary needs, but make strong use of ad-hoc report writers (such as QUIZ) for unique user-requested reports. Small systems of limited life are quickly developed without programming by creating data bases, maintaining them with HPGSUG Contributed software such as DBENTRY, modifying them with ADAGER as needed, and generating reports with QUERY, QUIZ, or REX.

In short, the old adage "use the right tool for the job" will take on more meaning in the future, as more tools are developed to

service our needs. We would hope that one of the tools selected to enhance programmer productivity is the creation of shop standards, no matter what size the shop is. It is one of the most effective means available, when one realizes that the cost is, simply, better planning and more efficient usage of resources. That is a price that all shops can afford to pay.

SUPPLEMENT, COBOL Coding Standards

COBOL programming has always emphasized clarity and ease of maintenance, and these objectives form the basis for most standards manuals in the industry today. Careful attention should be paid in their development, however, to avoid standards that are too lax and inconsistent or too restrictive and inflexible. Such an attempt has been made at Whitman College as the standards presented here are designed to be flexible and yet insure some degree of continuity between programs.

The following discussion of Whitman COBOL Coding Standards is divided into two major sections, General Standards and Standards by Division. Those topics that do not fit within a particular COBOL division are included as general standards and they are as follows:

* Compilation Techniques
* Model "cut and paste" Skeletons
* COBOL Copy Library
* Subsystem Calls
* Common Code Table (CCT)
* Error Handling
* Abort Processing
* Console Posting

The four major COBOL divisions contain standards that are specific to a particular division, and they will be included where appropriate.

General COBOL standards will be discussed first, followed by a presentation of the standards by division. Appendix A includes ALL of the examples noted in this section, and will be referenced by the letter A and a number indicating the page of the appendix containing the figure (Ex - see page A-7). All COBOL examples used are from a production program, CG228, that was modified to test the current set of programming standards, and from the V/3000 skeleton program.

General Standards

* Compilation Techniques. In many shops, the days of the interactive compiles are over. The demands on system resources are too great to allow programmers the luxury of a "quick" compile to get a new, fresh listing. This is precisely the case at Whitman. To help insure that compiles are in fact streamed, a new utility program, BANNER (included in the Contributed Library), has been developed to process COBOL compiles. It works in most other languages as well.

  BANNER must be run in a batch mode in order to obtain the information required on the cover sheet about the source, and then insure that it be printed with the source listing. The file equations required for BANNER may be set up in a UDC (see page A-1) to facilitate its use. It creates a cover sheet with large block letters indicating the PROGRAM-ID, and many other pieces of information about the compile including the actual file name used and the User who streamed the job (see page A-1). Actually two identical headers are produced so that cover sheets need never be folded! The information on this cover sheet has also proved invaluable in helping operators deliver the source compile to the proper individual since in many cases involving maintenance the person working on the program is not the original author.

* Model "cut and paste" Skeletons. In order to assist the programmers at Whitman College, special model programs (known as skeletons) have been developed to serve as a base starting point for all new development (see page A-2). The first two pages of a skeleton program are included in the appendix. Note the use of special abbreviations and X's to indicate values that must be entered by the programmer. These skeletons are clean-compiled programs containing each of the four COBOL divisions with all necessary elements and copylib routines normally found in the particular type of program (report, prompt/answer, V/3000, etc). Consequently no one actually ever writes a program from "scratch." A solid foundation is laid for the programmer before any attempt at adding the new program logic is made. This helps keep existing employees on top of the current set of standards in the shop, and helps new employees become familiar with the acceptable coding standards at a much faster rate. These skeletons may also serve as a guide when older programs are modified and standardized, and as a guide to the acceptable level of programming expected in this shop.

* COBOL Copy Library. Whenever possible and practical, COPYLIB routines are created to standardize a set of common variables in working storage or establish a standard paragraph for the procedure division. Special naming conventions are used when naming the routine so that it may be easily identified (see page A-3). Certain standard headings are also required of the COPYLIB routines to insure consistency in the COBOL copy

library, and these fall into several different categories (see page A-4) depending on the type of working storage area. To help insure that COPYLIB standards are followed and understood, one individual in the shop is assigned the responsibility for the maintenance of this file. It is intended that these Copylib routines make the code more consistent and easier to maintain, and reduce as much duplicate effort as possible by providing standard procedures for every programmer to use.

* Subsystem Calls. Special attention is given to standardization of HP specific software subsystems, such as KSAM, V/3000, and IMAGE. The reason most obvious is that the shop cannot afford the time for a "learning curve" for new employees. No matter how well trained in COBOL, few new employees know HP subsystems! The copylib, combined with the skeleton "cut and paste" method, enforces standards while propping up productivity.

  - KSAM. All working storage areas for production KSAM files are standardized in COBOL copylib routines. A standard file format is created for each using proper descriptive prefixes. All production programs access at least one KSAM file, the Common Code Table (CCT), to obtain values for report and screen headings as well as a variety of other codes. Standard lookup procedures have been developed for this special KSAM file since it is so heavily used (see pages A-12, A-13). A sample table, 901 (see page A-11), is included as an example of the file contents. All other KSAM files are referenced in a similar manner, and are built using the same standard format in working storage.

  - V/3000. V/3000 standard screen techniques are heavily used in this shop. As time permits, DEL/3000 programs are being rewritten using the new set of V/3000 standards and all new screen handling programs are being built from a skeleton. A standard V/3000 communications area is included in the COBOL Copylib, and a special VIEW-DATA-BUFFER structure has been developed as well (see page A-5 for a partial picture). The VIEW-DATA-BUFFER has three distinct breakouts. The first contains the system and office names, and the third contains the form name. These three fields must be numbered 1, 2, and 3 on each V/3000 screen in order for the standard heading routines to work properly. That leaves VIEW-DATA-BUFFER-2 as the area for all other screen fields. Using this VIEW-DATA-BUFFER, standard procedures get, display, and read V/3000 screens. The initialization routine is included in the appendix (see page A-6) to show how the first screen is processed. From that point on, standard individual procedures are called to manipulate the V/3000 screens.

  - IMAGE. One of the most heavily used HP subsystems is IMAGE. Fairly specific standards have been established for working

storage areas as well as procedure division calls since IMAGE is used in many production programs. Examples of the standard IMAGE working storage buffer and the list and buffer areas for two Student data base data sets are included in appendix A (see pages A-7 and A-8) to give you an idea of the conventions used.

The IMAGE calls used in the procedure division have recently undergone a major change. In the past the practice has been to use ALL-ITEMS as the list item in an IMAGE call. When the most recent set of standards were modified, this practice came under scrutiny and to the most part has been abandoned in favor of the PREVIOUS-LIST option due to greater processing efficiency and ease of maintenance.

Initially the working storage areas of the program must be created containing lists of the desired data elements and the corresponding data buffers containing the same elements (see page A-8). Each IMAGE list is then ready for "initialization". This "initialization" involves a serial read of one record in each data set using the actual LIST as it is defined in working storage (see page A-9). Care must be taken to properly process the first record, however, should any data set actually be read serially. All necessary information is obtained from the IMAGE root file with the read of these initial records, and is stored away for future use. From this point on, all IMAGE calls involving a LIST item should use the IMAGE-PREVIOUS-LIST value (see page A-7) to avoid the overhead of going to the root file to obtain the LIST information with each call (see page A-10).

There are tradeoffs between the ALL-ITEMS and PREVIOUS-LIST options, and they must be considered when deciding how to best use IMAGE in a given program. The following criteria may be used as a guide when a new program is being developed or a program is being modified:

1. execution - by priming all data sets used with the proper IMAGE lists (manually selected lists containing only the data elements necessary) and using the PREVIOUS-LIST option mentioned above, time will be saved during program execution since the root file information is only obtained once,

2. maintenance - by using manually prepared lists instead of ALL-ITEMS, only programs containing affected data items will require recompilation when a data base is rebuilt,

3. additions - when records are being added to a data base, it takes less overhead to use the ALL-ITEMS option since most of the data items are usually included in an add mode anyway,

4. working storage requirements - only one Copylib routine for each data set would be required for the ALL-ITEMS option, but these buffers would be the largest possible and fixed in size. The PREVIOUS-LIST option would necessitate the creation of more Copylib areas, but the buffer sizes in the programs would be optimized.

* Common Code Table. A special KSAM file has been created at Whitman College to house certain "Common Codes". This file contains over fifty tables which house codes from several different systems. These codes are unique and have the same meaning no matter what system they are found in. Some examples are Admissions Office Test Codes, Registrar Student Status Codes, Financial Aid Adjustment Codes, and United States Postal Codes (see page A-11). This standardization of codes simplifies maintenance, and since these are external tables changes do not usually affect the programs. The standard working storage area for the Common Code Table (CCT) and the common routines required to obtain information from this "table" are included in the appendix to show how the CCT is normally accessed (see pages A-12 and A-13).

* Error Handling. Another Common Code Table is used by programmers in the development of error handling procedures. This is a table of standard Error Messages (a partial list is on page A-14). All programs requiring User error messages must access this table as no program-imbedded error messages are allowed. If a programmer does need a new error message, then the analyst responsible for the Common Code Table maintenance must approve the request and make sure that it is entered properly.

  To help describe how errors are handled in a screen program, a few sample V/3000 error handling paragraphs are located on page A-15 of the appendix. When an error occurs in a typical V/3000 program, the CCT-ERROR-KEY is primed with the proper error number, the VIEW-FIELD-NUM is primed with the number of the field in error, and error handling procedures are performed. Screen error handling is controlled basically from paragraph P625-VIEW-SECONDARY-ERROR (see page A-15). All routines required to lookup the error message, display it to the User, and flash the field in error are performed from this standard routine. Errors are processed in this manner until all have been corrected.

  One key reason for the standardizing of error messages is to facilitate User awareness and understanding of their particular

system. Consistent error messages make it easier for the User to become familiar with the types of errors and problems that may occur. Each error message is given a unique number to increase the ease of lookup, and to avoid unnecessary duplication. Unique error messages also insure program and programmer consistency with regard to error handling.

* Abort Processing. The same philosophy of consistency holds for abort messages as well. Standard IMAGE, KSAM, and V/3000 abort COPYLIB routines have been developed to standardize abort procedures and messages. Refer to pages A-16 and A-17 of the appendix to find samples of the general abort working storage area, a typical paragraph calling an abort procedure, and the abort procedure itself. In this case an IMAGE abort is used to illustrate the steps required to perform a standard abort. The abort messages come from the same Common Code Table mentioned above, and are written to the Users screen or job execution report when an abort condition occurs. The messages are designed to help the programmers pinpoint the actual line within a particular paragraph that the abort occurred, and give the User some indication of the problem so that they can notify the computer center with meaningful information. This speeds the identification of the problem, and hence the solution since finding out exactly where something went wrong is often over half the battle.

  In addition to notifying the User of an abort condition, each standard abort routine sends a very obvious message to the operator's console (an entire line of lozenges) so that they are made aware of a problem (see page A-18). This is especially helpful when a batch job aborts since the printer may be busy for hours, thus delaying the printout of a job execution report that tells the operator that the job "blew up". Messages of this type necessitate the existence of a hard copy console, however, to insure that the abort message is "heard" and not lost on a screen that "rolled up" and disappeared. This instant operator notification is another mechanism designed to insure that exception conditions are noticed as early as possible so that proper actions may be taken to recover.

* Console Posting. In another effort to assist the operator in day-to-day activities, a standard set of "TELL" procedures have been developed and are installed in all production programs. These special procedures identify three points in each job process: 1) TELLSTART, 2) TELLFORM and 3) TELLSTOP. Using a special subroutine, these messages are printed in columns 81-132 of the operators console to help distinguish them from MPE messages. Other information such as the User, the file Output number (#O), the form name, and the job/session number are also included in these operator messages to help them readily identify jobs and form requirements (see page A-19).

Standards by Division

* IDENTIFICATION DIVISION

The first three pages in every source program should look fundamentally the same. Special standards have been established for all three pages so that by the time these pages have been read specific pieces of information about the program will be known in some detail. It is too easy to specify a PROGRAM-ID and go onto the ENVIRONMENT DIVISION without the inclusion of any other pertinent information.

To avoid this all too common occurence, a special Title page is included (see page A-20). Using the $TITLE command, the title of the program is printed on the left side of each compile and the PROGRAM-ID and revision level are printed on the right. Every subsequent page will have the title and program number on it unless overridden by a $PAGE heading. Following this important piece of information are:

PROGRAM-ID	program name <revision level>
DATE-WRITTEN	month, year
INSTALLATION	Whitman College
SECURITY	Public, Restricted, or Confidential

To identify as many of the people involved with a particular program as possible, a special author/analyst section is included. It lists the programmer(s), systems analyst(s), and system designer(s). To complete the title page, a brief summary of the program's purpose is given, along with all necessary file descriptions and non-standard subroutine calls.

The second page of a COBOL program (see page A-21) contains detailed compilation instructions which are especially important to programmers unfamiliar with the program. The MAXDATA information is a very important piece of information, and it should be updated properly so that programs are compiled correctly. A maintenance log is also included to identify all of the modifications that have been made and by whom (initials). This can be helpful in understanding the extent of modifications that have been made to a particular program and learning some of its history as well.

One particular piece of information that is often left to the last minute and then left out due to time pressure is the synopsis of program logic (see page A-22). This in reality should be one of the first things written in a program as it can help organize the programmer's logic and possibly pinpoint special problem areas or give insight to better solutions. At any rate, this is a very important standard as it can really help a maintenance programmer get a better feel for what is going on in a program as well as refresh the memory of the author who may have written the program some time ago. Finally, this synopsis should be written in enough detail to

give a good logic flow of the program, but be simple enough so that someone unfamiliar with the program can derive a basic understanding from it.

* ENVIRONMENT DIVISION

This division is always the smallest, but nevertheless it contains some fundamental information that should not be overlooked (see page A-23). First of all a $PAGE is used to isolate the division and make it easier to find. Some standard special names are used to refer to the console and for top-of-page.

The Input-Output Section contains the select and assign clauses. A special format is assigned to printer files so that they are easily recognizable on the console during a :SHOWOUT. All other files described in this division will require MPE file equations.

* DATA DIVISION

The most common problem found in data divisions is the inability to find variables without searching the entire working storage section. One of the basic standards used is the $PAGE. It helps delineate particular variables or copylib routines from one another so as to enhance readability. The file and working storage sections are also separated by a $PAGE to further separate them for easier reference.

Another standard developed to increase the ability to find things is the requirement that each variable in working storage be assigned a meaningful prefix (suffixes are acceptable but not preferred) to associate it with other common variables. Each area containing common variables should then be preceded by a header statement (see page A-24) describing what kind of variables are included. Some common areas of differentiation could be FLAGS, HOLD AREAS, TABLES, or other such breakouts.

All working storage copylib routines use the prefix notion in that all variables used in these routines should contain proper three-character prefixes. The copylib routines themselves are then given a file name that also contains the prefix. All of these special naming conventions are used to help keep common variables together to enhance readability, increase variable location, and make the procedure division logic easier to follow with the use of meaningful data names.

In order to make the working storage section even easier to read, all copylib routines and other variable areas should fall in a similar sequence. The following guidelines are used in the order given to arrange all working storage elements:

* all general copylib routines and general variables (flags, counters, save areas),

* communications areas,

* all file descriptions (KSAM, V/3000, IMAGE),

* all copylib report routines should be followed by report format areas and be the last contents of working storage.

A fairly common COBOL standard found in this division is the practice of aligning PICTURE clauses on some predetermined column. The standard used at Whitman stresses that PICTURE clauses line up in column 40 whenever possible. In conjunction with this standard is the practice of indenting data hierarchies in units of 4, with even levels being skipped. Indentation beyond the 05 level is not necessary.

* PROCEDURE DIVISION

At Whitman College, a modular approach to logic coding is used, with heavy emphasis on the PERFORM verb. One major mainline is included at the beginning of the Procedure Division, and the logic here controls the rest of the program. In fact, the program starts, stops, and/or aborts from this single mainline (see page A-25). This approach is used to add an extra measure of control, and to help organize the rest of the program around one common point.

The logic sections within the Procedure Division are meaningful paragraph names that are composed of both sequential numbers and titles. The range is:

001-099	Mainline Logic
100-199	Data File Input
200-299	Data File Output
300-399	Screen Procedures
400-499	Edits and Field Processing
500-599	(Program Specific)
600-699	Error Procedures
700-799	(Program Specific)
800-899	Report Procedures
900-999	Housekeeping.

Each of these logic sections should be preceded by a $PAGE and begin with the three-line standard header which indicates what routines are being performed (see page A-26).

Major program segments should be kept to a minimum, and the logic should remain in a segment as long as possible. This is done to reduce the amount of swapping that occurs each time a different segment has to be brought into memory. Traditionally a COBOL sort program contains three major segments or sections: Mainline Logic Section, Sort Section (input procedure), and Report Section (output procedure). A data entry program often can perform all of its functions out of one section, the Mainline.

As in the Data Division, readability is of key importance. In order to improve readability of this division, only one logic statement should be coded per line. Also standard indentations should be observed in statements with multiple verbs. Arguments in a USING clause are listed on separate lines underneath the first argument to further enhance the code.

Each logic module should be organized into a paragraph which has only one exit and one entrance. Each paragraph should have an enclosed comment preceding it that summarizes the logic found in that logic module. A $PAGE is used to separate the paragraph for better readability.

The use of GO TO's in programming has created much controversy among programmers and proponents of different styles of COBOL coding. The standards developed at Whitman College allow GO TO statements to be used, but only with certain restrictions:

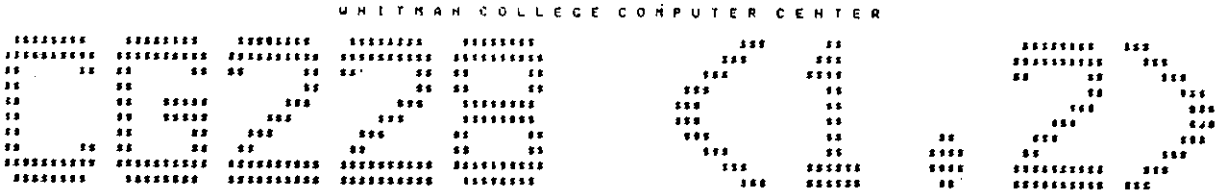
1. One Exit,
2. One Entrance,
3. Intermediate logic leading to 1 or 2,
4. Abort.

No other logic transfers should take place. "Fall through" should always transfer back to the Mainline. Rampant GO TO's that go to different paragraphs and even different sections are not permitted. Instead a controlled use of GO TO statements is proposed to preclude the use of unnecessary nested IF statements, and to make the code simpler and easier to follow.

APPENDIX A

```
:JOB COMPILE,USER/PASS,ACCT
:FILE COPYLIB=COBOLIB,PUB
:COBBANNER CG228C
:PURGE CG228
:PREP $OLDPASS,CG228
:SAVE CG228
:EOJ
```

```
COBBANNER TEXT,USL=$NEWPASS,LIST=$STDLIST,MAST=$NULL,NEW=$NULL
FILE COBTEXT=!TEXT
FILE COBUSL=!USL
FILE COBLIST=!LIST
FILE COBMAST=!MAST
FILE COBNEW=!NEW
RUN BANNER,UTIL,IRIS;PARM=0
RESET COBTEXT
RESET COBUSL
RESET COBLIST
RESET COBMAST
RESET COBNEW
```



COBOL Compile for: MANAGER.WCCS,PUB

MON, MAR 2, 1981, 10:39 AM

Source File Name: MODELC.PUB.WCCS

Compiler File Statistics

Formal Name	Actual Name
COBTEXT	MODELC.PUB.WCCS
COBUSL	#NEWPASS
COBLIST	$STDLIST (#STDLIST)
COBMAST	#NULL
COBNEW	#NULL

Source File Statistics

Name	: MODELC.PUB.WCCS
Creator	: MANAGER
LDEV	: 3
Filecode	: 1052
File Limit	: 1113
EOF Pointer	: 1113
Max Extents	: 15
Ext Size	: 25 Sectors
Block Size	: 1280 Bytes
Record Size	: 80 Bytes

- A M.A.R.C.U.S. PRODUCTION -

```
001000$CONTROL USLINIT,SOURCE
001100$TITLE " >> .........1.........2.........3.........4 << ",N
001200$"                                  PPPPP <RRR>"
001300 IDENTIFICATION DIVISION.
001400 PROGRAM-ID.                        "PPPPP <RRR>".
001500 DATE-WRITTEN.                       XXXXXXXX, XXXX.
001600 INSTALLATION.                      WHITMAN COLLEGE.
001700 SECURITY.                          SSSSSSSSSSS.
001800*
001900*     *-------------------------------------------------------*
002000*     |                >> NOTICE OF COPYRIGHT <<              |
002100*     *-------------------------------------------------------*
002200*     |    PPPPP is Whitman College Proprietary software      |
002300*     *=======================================================*
002400*     |   Programming  | Systems Analysis  | Systems Design   |
002500*     *-------------------------------------------------------*
002600*     |      AAAAA     |       BBBBBB      |      CCCCCCC     |
002700*     *-------------------------------------------------------*
002800*
002900*     *-------------------------------------------------------*
003000*      SUMMARY:  PPPPP XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
003100*                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
003200*
003300*                (1)  File Descriptions:
003400*                     UT145K01   Common Code Table
003500*                     FFFFWIDE   Report Print File
003600*
003700*                (2)  Subroutine calls:
003800*                     UT817      Obtains JOB/USER Name
003900*                     UT818      Console Tell Form Routine
004000*
004100*     *-------------------------------------------------------*
```



```
004300*     *-------------------------------------------------------*
004400*      Compilation Instructions:
004500*
004600*           :FILE P;DEV=COMLP
004700*           :FILE COPYLIB=COBOLIB,PUB,DEV
004800*           :COBOL PPPPPC,SOURCE,DEV,,*P
004900*           :PREP SOLDPASS,PPPPP,XXXXXXXX,ADMIN;MAXDATA=XXXX
005000*
005100*     *-------------------------------------------------------*
005200*
005300*     *-------------------------------------------------------*
005400*       DATE    REV   BY    MAINTENANCE LOG
005500*     XX/XX/XX <1.0> XXX Primary Installation of software.
005600*
005700*     *-------------------------------------------------------*
```

```
        034600 01  RPT-DUMMY                          COPY RPT021WS.
RPT021WS 001000     .
RPT021WS 001100     03  COPY-CONTROL-BYTE             PIC X(01).
RPT021WS 001200*    *-------------------------------------------------*
RPT021WS 001300*          Standard Report Print Routine Common Formats
RPT021WS 001400*    *-------------------------------------------------*
RPT021WS 001410 01  RPT-STD-LOGON.
RPT021WS 001420     03  PPT-STD-SESSION-NBR           PIC X(06).
RPT021WS 001430     03  FILLER                        PIC X(01)  VALUE "/".
RPT021WS 001440     03  RPT-STD-JOB-NBR               PIC X(06).
RPT021WS 001500 01  RPT-SUB-HEAD-1                    PIC X(132) VALUE SPACES.
RPT021WS 001600 01  RPT-SUB-HEAD-2                    PIC X(132) VALUE SPACES.
RPT021WS 001700 01  RPT-SUB-HEAD-3                    PIC X(132) VALUE SPACES.
RPT021WS 001800 01  RPT-COL-HEAD-1                    PIC X(132) VALUE SPACES.
RPT021WS 001900 01  RPT-COL-HEAD-2                    PIC X(132) VALUE SPACES.
RPT021WS 002000 01  RPT-COL-HEAD-3                    PIC X(132) VALUE SPACES.
RPT021WS 002100 01  RPT-PRINT-LINE-OUT.
RPT021WS 002200     03  RPT-NARROW-PRINT-LINE         PIC X(96)  VALUE SPACES.
RPT021WS 002300     03  FILLER                        PIC X(36)  VALUE SPACES.
RPT021WS 002400 01  RPT-HOLD-PRINT-LINE.
RPT021WS 002500     03  RPT-NARROW-HOLD-LINE          PIC X(96)  VALUE SPACES.
RPT021WS 002600     03  FILLER                        PIC X(36)  VALUE SPACES.
RPT021WS 002700 01  RPT-PAGE-COUNT                    PIC 9(05)  VALUE 0.
RPT021WS 002800 01  RPT-LINE-COUNT                    PIC 9(02)  VALUE 99.
RPT021WS 002900 01  RPT-LINE-SLEW                     PIC 9(02)  VALUE 1.
RPT021WS 003000 01  RPT-HOLD-SLEW                     PIC 9(02)  VALUE 0.
RPT021WS 003100 01  RPT-TOP-CODE                      PIC X(01)  VALUE SPACE.
RPT021WS 003200     88  RPT-TOP-OF-PAGE-NEEDED                  VALUE "T".
RPT021WS 003300 01  RPT-SWITCHES.
RPT021WS 003400     03  RPT-SW-SUB-1                 PIC 9(01)  VALUE 0.
RPT021WS 003500         88  RPT-SUB-HEAD-1-NEEDED               VALUE 1.
RPT021WS 003600     03  RPT-SW-SUB-2                 PIC 9(01)  VALUE 0.
RPT021WS 003700         88  RPT-SUB-HEAD-2-NEEDED               VALUE 1.
RPT021WS 003800     03  RPT-SW-SUB-3                 PIC 9(01)  VALUE 0.
RPT021WS 003900         88  RPT-SUB-HEAD-3-NEEDED               VALUE 1.
RPT021WS 004000     03  RPT-SW-COL-1                 PIC 9(01)  VALUE 0.
RPT021WS 004100         88  RPT-COLUMN-LINE-1-NEEDED            VALUE 1.
RPT021WS 004200     03  RPT-SW-COL-2                 PIC 9(01)  VALUE 0.
RPT021WS 004300         88  RPT-COLUMN-LINE-2-NEEDED            VALUE 1.
RPT021WS 004400     03  RPT-SW-COL-3                 PIC 9(01)  VALUE 0.
RPT021WS 004500         88  RPT-COLUMN-LINE-3-NEEDED            VALUE 1.
```

```
       027200 01  CCT-DUMMY                        COPY CCT001WS.
CCT001WS 001000     .
CCT001WS 001100     03  COPY-CONTROL-BYTE           PIC X(01).
CCT001WS 001200*    *-------------------------------------------------*
CCT001WS 001300*     FILE FORMAT: UT145K01--Common Codes Table file
CCT001WS 001400*    *-------------------------------------------------*
CCT001WS 001500 01  CCT-BASIC-RECORD.
CCT001WS 001600         03  CCT-KEY                PIC X(15).
CCT001WS 001700         03  CCT-VALUE              PIC X(50).


       028400 01  CDB-COM-DUMMY                   COPY CDB001WS.
CDB001WS 001000     .
CDB001WS 001100     03  COPY-CONTROL-BYTE   PIC X(01).
CDB001WS 001200*    *-------------------------------------------------*
CDB001WS 001300*              Data Base COMMON -- Communications Area
CDB001WS 001400*    *-------------------------------------------------*
CDB001WS 001500 01  CDB-COMMON-BASE-INFO.
CDB001WS 001600     03  CDB-BASE-NAME          PIC X(16) VALUE "  COMMON,COMMON;".
CDB001WS 001700     03  CDB-PASSWORD           PIC X(08) VALUE ";       ".
CDB001WS 001800 01  CDB-COMMON-DATA-SETS.
CDB001WS 001900     03  CDB-UIC-MST            PIC X(16) VALUE "UIC-MASTER;     ".
CDB001WS 002000     03  CDB-WID-MST            PIC X(16) VALUE "WID-MASTER;     ".
CDB001WS 002100     03  CDB-ATTRIBUTE-MST      PIC X(16) VALUE "ATTRIBUTE-MASTER".
CDB001WS 002200     03  CDB-ZIP-CODE-MST       PIC X(16) VALUE "ZIP-CODE-MASTER;".
CDB001WS 002300     03  CDB-ADDRESS-DTL        PIC X(16) VALUE "ADDRESS-DETAIL; ".
CDB001WS 002400     03  CDB-ATTRIBUTE-DTL      PIC X(16) VALUE "ATTRIBUTE-POINTR

       028500*
       028600*     *-------------------------------------------------*
       028700*                    CDB001L2 --- LIST #2
       028800*      WID-MASTER Master Data Set          Data Base COMMON
       028900*     *-------------------------------------------------*
       029000 01  CDB001L2-LIST.
       029100     03  FILLER                 PIC X(50) VALUE
       029200         "ID-PREFIX;                                      "
       029300 01  CDB001L2-AREA.
       029400     03  CDB-ID-PREFIX          PIC 9(03) COMP.


       038000*     *-------------------------------------------------*
       038100*                  Report CG228/F482 Format Area
       038200*     *-------------------------------------------------*
       038300 01  F482-SYSTEM                    PIC X(50).
       038400 01  F482-REPORT                    PIC X(10) VALUE "CG228/F482".
       038500 01  F482-PRIVACY                   PIC X(14) VALUE "RESTRICTED".
       038600 01  F482-TITLE-3.
       038700     03  F482-TITLE-LITERAL         PIC X(30).
       038800     03  FILLER                     PIC X(05) VALUE " for ".
       038900     03  F482-LITERAL               PIC X(15) VALUE SPACES.
       039000 01  F482-OFFICE                    PIC X(50).
```

```
013900*     *-------------------------------------------------*
014000*                VIEW/3000 Screen Buffers
014100*     *-------------------------------------------------*
014200 01  VIEW-DATA-BUFFER.
014300     02  VIEW-DATA-BUFFER-1.
014400         03  DATA-SYSTEM-NAME      PIC X(50).
014500         03  DATA-OFFICE-NAME      PIC X(50).
014600     02  VIEW-DATA-BUFFER-2.
014700         03  DATA-ACTION           PIC X(01).
014800             88  END-OF-JOB                VALUE "!".
014900             88  VALID-ACTION              VALUE "A" "C" "D".
015000             88  ADD-ACTION                VALUE "A".
015100             88  CHANGE-ACTION             VALUE "C".
015200             88  DELETE-ACTION             VALUE "D".
015300         03  DATA-ID-CODE          PIC X(06).
015400     02  VIEW-DATA-BUFFER-3.
015500         03  DATA-FORM-TITLE       PIC X(50).
015600         03  DATA-VALIDATE         PIC X(01).
015700             88  DATA-IS-VALID             VALUE "/".

VEW021WS 001200*     *-------------------------------------------------*
VEW021WS 001300*          VIEW/3000 Communications and Parameter Area
VEW021WS 001400*     *-------------------------------------------------*
VEW021WS 001500 01  VIEW-ERROR-MSG.
VEW021WS 001600     03  VIEW-ERROR-CODE          PIC X(15) VALUE SPACES.
VEW021WS 001700     03  FILLER                   PIC X(57) VALUE SPACES.
VEW021WS 001800 01  VIEW-FIELD-NUM               PIC S9(04) COMP VALUE 0.
VEW021WS 001900 01  VIEW-NEXT-FLD-NUM            PIC S9(04) COMP VALUE 0.
VEW021WS 002000 01  VIEW-ERROR-FIELD-NUM         PIC S9(04) COMP VALUE 0.
VEW021WS 002100 01  VIEW-LENWINDOWMSG            PIC S9(04) COMP VALUE 0.
VEW021WS 002200 01  VIEW-ACTUAL-LEN              PIC S9(04) COMP VALUE 0.
VEW021WS 002300 01  VIEW-LENFLDBUFF              PIC S9(04) COMP VALUE 0.
VEW021WS 002400 01  VIEW-LENDATABUFF             PIC S9(04) COMP VALUE 0.
VEW021WS 002500 01  VIEW-LENERRMSG               PIC S9(04) COMP VALUE 0.
VEW021WS 002600 01  VIEW-LENERRBUFF              PIC S9(04) COMP VALUE 0.
VEW021WS 002700 01  VIEW-MESSAGE-BUFFER.
VEW021WS 002800     03  FILLER                   PIC X(13) VALUE
VEW021WS 002900         " X c &a23r20C".
VEW021WS 003000     03  VIEW-MSG-ENHANCEMENT     PIC X(04) VALUE SPACES.
VEW021WS 003100     03  VIEW-MSG-BUFF            PIC X(50) VALUE SPACES.
VEW021WS 003200     03  FILLER                   PIC X(04) VALUE " b W".
VEW021WS 003300 01  VIEW-MESSAGE-BUFFER-2.
VEW021WS 003400     03  VIEW-MSG-ENHANCEMENT-2   PIC X(04) VALUE SPACES.
VEW021WS 003500     03  VIEW-MSG-BUFF-2          PIC X(72) VALUE SPACES.
VEW021WS 003600 01  VIEW-STANDARD-ENHANCEMENT    PIC X(04) VALUE " &dJ".
VEW021WS 003700 01  VIEW-ERROR-ENHANCEMENT       PIC X(04) VALUE " &dF".
VEW021WS 003800 01  VIEW-CLEAR-ENHANCEMENT       PIC X(04) VALUE " &d@".
VEW021WS 003900 01  VIEW-DUMMY-PARAM             PIC X(02).
VEW021WS 004000 01  VIEW-TERM-NAME               PIC X(09) VALUE "TERMINAL".
VEW021WS 004100*
VEW021WS 004200 01  VIEW-COM-AREA.
VEW021WS 004300     03  VIEW-STATUS              PIC S9(04) COMP VALUE 0.
VEW021WS 004400         88  VIEW-OP-VALID        VALUE 0.
VEW021WS 004500     03  VIEW-LANGUAGE            PIC S9(04) COMP VALUE 0.
VEW021WS 004600     03  VIEW-COM-AREA-LENGTH     PIC S9(04) COMP VALUE 60.
VEW021WS 004700     03  FILLER                   PIC S9(04) COMP VALUE 0.
VEW021WS 004800     03  VIEW-CURRENT-MODE        PIC S9(04) COMP VALUE 0.
VEW021WS 004900     03  VIEW-LAST-KEY            PIC S9(04) COMP VALUE 0.
```

```
           036700*----------------*-------------------------------------*----------------
           036800*----------------*      Screen Procedures              *----------------
           036900*----------------*-------------------------------------*----------------
           037000*
           037100 P300-DUMMY. COPY VEW300P2.
VEW300P2   001000*
VEW300P2   001100*     *-------------------------------------------------------*
VEW300P2   001200*       This routine displays the View/3000 form to the screen
VEW300P2   001300*       and performs the initial read of the screen.
VEW300P2   001400*     *-------------------------------------------------------*
VEW300P2   001600 P300-VIEW-DISPLAY-FORM.
VEW300P2   001700     CALL "VGETNEXTFORM" USING VIEW-COM-AREA.
VEW300P2   001800     IF  NOT VIEW-OP-VALID
VEW300P2   001900         MOVE "925??P300A" TO GEN-ABORT-BREAKOUT.
VEW300P2   002000         PERFORM P658-VIEW-STANDARD-ERROR THRU P658-EXIT.
VEW300P2   002100         GO TO P099-ABORT.
VEW300P2   002200 P300-INIT-FORM.
VEW300P2   002300     CALL "VINITFORM" USING VIEW-COM-AREA.
VEW300P2   002400     IF  NOT VIEW-OP-VALID
VEW300P2   002500         MOVE "925??P300B" TO GEN-ABORT-BREAKOUT.
VEW300P2   002600         PERFORM P658-VIEW-STANDARD-ERROR THRU P658-EXIT.
VEW300P2   002700         GO TO P099-ABORT.
VEW300P2   002800 P300-SHOWFORM.
VEW300P2   002900     PERFORM P375-VIEW-SCREEN-TITLES THRU P375-EXIT.
VEW300P2   003000     CALL "VSHOWFORM" USING VIEW-COM-AREA.
VEW300P2   003100     IF  NOT VIEW-OP-VALID
VEW300P2   003200         MOVE "925??P300C" TO GEN-ABORT-BREAKOUT.
VEW300P2   003300         PERFORM P658-VIEW-STANDARD-ERROR THRU P658-EXIT.
VEW300P2   003400         GO TO P099-ABORT.
VEW300P2   003500 P300-READ-FORM.
VEW300P2   003600     CALL "VREADFIELDS" USING VIEW-COM-AREA.
VEW300P2   003700     IF  NOT VIEW-OP-VALID
VEW300P2   003800         MOVE "925??P300D" TO GEN-ABORT-BREAKOUT.
VEW300P2   003900         PERFORM P658-VIEW-STANDARD-ERROR THRU P658-EXIT.
VEW300P2   004000         GO TO P099-ABORT.
VEW300P2   004100 P300-GET-BUFFER.
VEW300P2   004200     CALL "VGETBUFFER" USING VIEW-COM-AREA
VEW300P2   004300                             VIEW-DATA-BUFFER
VEW300P2   004400                             VIEW-LENDATABUFF.
VEW300P2   004500     IF  NOT VIEW-OP-VALID
VEW300P2   004600         MOVE "925??P300E" TO GEN-ABORT-BREAKOUT.
VEW300P2   004700         PERFORM P658-VIEW-STANDARD-ERROR THRU P658-EXIT.
VEW300P2   004800         GO TO P099-ABORT.
VEW300P2   004900 P300-EXIT.
VEW300P2   005000     EXIT.
```

```
           028200 01   IMAGE-COM-DUMMY                COPY IMG021WS.
IMG021WS   001000        .
IMG021WS   001100     03  COPY-CONTROL-BYTE           PIC X(01).
IMG021WS   001200*      *-------------------------------------------------*
IMG021WS   001300*            IMAGE/3000 Communications and Parameter Area
IMG021WS   001400*      *-------------------------------------------------*
IMG021WS   001500 01   IMAGE-LENERRBUF                PIC S9(04) USAGE COMP.
IMG021WS   001600 01   IMAGE-STATUS.
IMG021WS   001700           03  IMAGE-CONDITION       PIC S9(04) USAGE COMP.
IMG021WS   001800               88  IMAGE-OP-VALID                VALUE  0.
IMG021WS   001900               88  IMAGE-BAD-PASSWORD            VALUE -21.
IMG021WS   002000               88  IMAGE-EOF                     VALUE 11.
IMG021WS   002100               88  IMAGE-RECORD-NOT-FOUND        VALUE 17.
IMG021WS   002200               88  IMAGE-END-OF-CHAIN            VALUE 15.
IMG021WS   002300               88  IMAGE-BEGINNING-OF-CHAIN      VALUE 14.
IMG021WS   002400           03  IMAGE-WORD2           PIC S9(04) USAGE COMP.
IMG021WS   002500           03  IMAGE-WORD3-4         PIC S9(09) USAGE COMP.
IMG021WS   002600           03  IMAGE-WORD5-6         PIC S9(09) USAGE COMP.
IMG021WS   002610               88  IMAGE-NO-DETAILS-FOUND        VALUE 0.
IMG021WS   002700           03  IMAGE-WORD7-8         PIC S9(09) USAGE COMP.
IMG021WS   002800           03  IMAGE-WORD9-10        PIC S9(09) USAGE COMP.
IMG021WS   002900 01   IMAGE-MODES.
IMG021WS   003000           03  IMAGE-MODE1           PIC S9(04) USAGE COMP VALUE 1.
IMG021WS   003100           03  IMAGE-MODE2           PIC S9(04) USAGE COMP VALUE 2.
IMG021WS   003200           03  IMAGE-MODE3           PIC S9(04) USAGE COMP VALUE 3.
IMG021WS   003300           03  IMAGE-MODE4           PIC S9(04) USAGE COMP VALUE 4.
IMG021WS   003400           03  IMAGE-MODE5           PIC S9(04) USAGE COMP VALUE 5.
IMG021WS   003500           03  IMAGE-MODE6           PIC S9(04) USAGE COMP VALUE 6.
IMG021WS   003600           03  IMAGE-MODE7           PIC S9(04) USAGE COMP VALUE 7.
IMG021WS   003700           03  IMAGE-MODE8           PIC S9(04) USAGE COMP VALUE 8.
IMG021WS   003800           03  IMAGE-MODE9           PIC S9(04) USAGE COMP VALUE 9.
IMG021WS   003900 01   IMAGE-ERROR-MSG                 PIC  X(72).
IMG021WS   004000 01   IMAGE-DUMMY-PARAM              PIC S9(04) USAGE COMP.
IMG021WS   004100 01   IMAGE-ALL-ITEMS                PIC  X(02) VALUE "@;".
IMG021WS   004200 01   IMAGE-NULL-ITEMS               PIC  X(02) VALUE "0;".
IMG021WS   004300 01   IMAGE-PREVIOUS-LIST            PIC  X(02) VALUE "*;".
IMG021WS   004400 01   IMAGE-DSET-NAME                PIC  X(16) VALUE SPACES.
IMG021WS   004500 01   IMAGE-KEY                      PIC  X(16) VALUE SPACES.
IMG021WS   004600 01   IMAGE-ARGUMENT                 PIC  X(10) VALUE SPACES.
```

```
029800*     *-------------------------------------------------*
029900*                      STU008L2 --- LIST #2
030000*      SEM-CREDIT-DTL Detail Data Set          Data Base STU
030100*     *-------------------------------------------------*
030200 01  STU008L2-LIST.
030300     03  FILLER                  PIC X(50) VALUE
030400         "SEMESTER-KEY,CUM-F-CREDITS,CUM-GPA,TOTL-CREDIT-EAR".
030500     03  FILLER                  PIC X(50) VALUE
030600         "N;                                              ".
030700 01  STU008L2-AREA.
030800     03  STU-SEMESTER-KEY        PIC 9(03) COMP.
030900     03  STU-CUM-F-CREDITS       PIC 9(02) COMP.
031000     03  STU-CUM-GPA             PIC 9V999 COMP.
031100     03  STU-TOTL-CREDIT-EARN    PIC 9(04) COMP.
031200*     *-------------------------------------------------*
031300*                      STU012L2 --- LIST #2
031400*      REG-STUDENT-DET Detail Data Set         Data Base STU
031500*     *-------------------------------------------------*
031600 01  STU012L2-LIST.
031700     03  FILLER                  PIC X(50) VALUE
031800         "WHITMAN-ID,STUDENT-LEVEL,NAME,STUDENT-STATUS,PROJ-".
031900     03  FILLER                  PIC X(50) VALUE
032000         "GRAD-MO,PROJ-GRAD-YR,CLASS-RANK,CLASS-SIZE,UPDATE-".
032100     03  FILLER                  PIC X(50) VALUE
032200         "DATE,UPDATE-TIME;                               ".
032300 01  STU012L2-AREA.
032400     03  STU-WHITMAN-ID          PIC X(06).
032500     03  STU-STUDENT-LEVEL       PIC X(01).
032600     03  FILLER                  PIC X(01).
032700     03  STU-NAME                PIC X(30).
032800     03  STU-STUDENT-STATUS      PIC X(02).
032900     03  STU-PROJ-GRAD-MO        PIC X(02).
033000     03  STU-PROJ-GRAD-YR        PIC X(02).
033100     03  STU-CLASS-RANK          PIC 9(03) COMP.
033200     03  STU-CLASS-SIZE          PIC 9(03) COMP.
033300     03  STU-UPDATE-DATE         PIC X(08).
033400     03  STU-UPDATE-TIME         PIC X(06).
033500 01  STU-REG-LOCK-DESCRIPTOR.
033600     03  STU-REG-LOCK-DESC-NBR   PIC S9(04) COMP VALUE 1.
033700     03  STU-REG-LOCK-DESC-1.
033800         05  STU-REG-LOCK-LENGTH PIC S9(04) COMP VALUE 21.
033900         05  STU-REG-LOCK-DSET   PIC X(16)  VALUE
034000             "REG-STUDENT-DET;".
034100         05  STU-REG-LOCK-ITEM   PIC X(16)  VALUE
034200             "WHITMAN-ID;     ".
034300         05  STU-REG-LOCK-RELOP  PIC X(02)  VALUE "= ".
034400         05  STU-REG-LOCK-VALUE  PIC X(06)  VALUE SPACES.
```

```
058500*     *-------------------------------------------------*
058600*      This routine establishes the IMAGE lists for the
058700*      following data sets:  WID-MASTER, REG-STUDENT-DET, and
058800*      SEM-CREDIT-DTL.  The previous list option will then
058900*      be used in all subsequent IMAGE calls.
059000*     *-------------------------------------------------*
059100 P951-ESTABLISH-IMAGE-LISTS.
059200     CALL "DBGET"   USING CDB-BASE-NAME
059300                          CDB-WID-MST
059400                          IMAGE-MODE2
059500                          IMAGE-STATUS
059600                          CDB001L2-LIST
059700                          CDB001L2-AREA
059800                          IMAGE-DUMMY-PARAM.
059900     IF  NOT IMAGE-OP-VALID
060000         MOVE "92598P951A" TO GEN-ABORT-BREAKOUT.
060100         GO TO P951-ERROR-OUT.
060200     CALL "DBGET"   USING STU-BASE-NAME
060300                          STU-SEM-CRED-DTL
060400                          IMAGE-MODE2
060500                          IMAGE-STATUS
060600                          STU008L2-LIST
060700                          STU008L2-AREA
060800                          IMAGE-DUMMY-PARAM.
060900     IF  NOT IMAGE-OP-VALID
061000         MOVE "92598P951B" TO GEN-ABORT-BREAKOUT.
061100         GO TO P951-ERROR-OUT.
061200     CALL "DBGET"   USING STU-BASE-NAME
061300                          STU-REG-STUD-DTL
061400                          IMAGE-MODE2
061500                          IMAGE-STATUS
061600                          STU012L2-LIST
061700                          STU012L2-AREA
061800                          IMAGE-DUMMY-PARAM.
061900     IF  NOT IMAGE-OP-VALID
062000         MOVE "92598P951C" TO GEN-ABORT-BREAKOUT.
062100         GO TO P951-ERROR-OUT.
062200     GO TO P951-EXIT.
062300 P951-ERROR-OUT.
062400     PERFORM P645-IMAGE-STANDARD-ERROR THROUGH P645-EXIT.
062500     GO TO P099-ABORT.
062600 P951-EXIT.
062700     EXIT.
```

```
080500*     *-------------------------------------------------*
080600*        This routine locks the STUDENT RECORDS Data Base.
080700*     *-------------------------------------------------*
080800 P250-LOCK-STU.
080900     CALL "DBLOCK" USING STU-BASE-NAME
081000                         STU-REG-LOCK-DESCRIPTOR
081100                         IMAGE-MODE5
081200                         IMAGE-STATUS.
081300 P250-EXIT.
081400     EXIT.
081500*
081600*     *-------------------------------------------------*
081700*        This routine unlocks the STUDENT RECORDS Data Base.
081800*     *-------------------------------------------------*
081900 P255-UNLOCK-STU.
082000     CALL "DBUNLOCK" USING STU-BASE-NAME
082100                           IMAGE-DUMMY-PARAM
082200                           IMAGE-MODE1
082300                           IMAGE-STATUS.
082400 P255-EXIT.
082500     EXIT.
082600*
082700*     *-------------------------------------------------*
082800*        This routine updates the CLASS-RANK and the CLASS-
082900*        SIZE on the REG-STUD-DTL.
083000*     *-------------------------------------------------*
083100 P270-UPDATE-STU-DTL.
083200     CALL "DBUPDATE" USING STU-BASE-NAME
083300                           STU-REG-STUD-DTL
083400                           IMAGE-MODE1
083500                           IMAGE-STATUS
083600                           IMAGE-PREVIOUS-LIST
083700                           STU012L2-AREA.
083800 P270-EXIT.
083900     EXIT.
```

COMMON CODE TABLE 901
United States Postal Codes

KEY	BASIC VALUE
901AK	ALASKA
901AL	ALABAMA
901AR	ARKANSAS
901AZ	ARIZONA
901CA	CALIFORNIA
901CO	COLORADO
901CT	CONNECTICUT
901DC	DISTRICT OF COLUMBIA
901DE	DELAWARE
901FL	FLORIDA
901GA	GEORGIA
901HI	HAWAII
901IA	IOWA
901ID	IDAHO
901IL	ILLINOIS
901IN	INDIANA
901KS	KANSAS
901KY	KENTUCKY
901LA	LOUISIANA
901MA	MASSACHUSETTS
901MD	MARYLAND
901ME	MAINE
901MI	MICHIGAN
901MN	MINNESOTA
901MO	MISSOURI
901MS	MISSISSIPPI
901MT	MONTANA
901NC	NORTH CAROLINA
901ND	NORTH DAKOTA
901NE	NEBRASKA
901NH	NEW HAMPSHIRE
901NJ	NEW JERSEY
901NM	NEW MEXICO
901NV	NEVADA
901NY	NEW YORK
901OH	OHIO
901OK	OKLAHOMA
901OR	OREGON
901PA	PENNSYLVANIA
901RI	RHODE ISLAND
901SC	SOUTH CAROLINA
901SD	SOUTH DAKOTA
901TN	TENNESSEE
901TX	TEXAS
901UT	UTAH
901VA	VIRGINIA
901VT	VERMONT
901WA	WASHINGTON
901WI	WISCONSIN
901WV	WEST VIRGINIA
901WY	WYOMING

```
       027200 01  CCT-DUMMY                          COPY CCT001WS.
CCT001WS 001000     .
CCT001WS 001100     03  COPY-CONTROL-BYTE              PIC X(01).
CCT001WS 001200*     *-------------------------------------------------*
CCT001WS 001300*      FILE FORMAT: UT145K01--Common Codes Table file
CCT001WS 001400*     *-------------------------------------------------*
CCT001WS 001500 01  CCT-BASIC-RECORD.
CCT001WS 001600         03  CCT-KEY                   PIC X(15).
CCT001WS 001700         03  CCT-VALUE                 PIC X(50).
CCT001WS 001800 01  CCT-FILE-TABLE.
CCT001WS 001900         03  CCT-FILE-NUMBR        PIC S9(04) VALUE 0 USAGE COMP.
CCT001WS 002000         03  CCT-FILE-NAME         PIC  X(08) VALUE "UT145K01".
CCT001WS 002100         03  CCT-FILE-I-O          PIC S9(04) VALUE 0 USAGE COMP.
CCT001WS 002200         03  CCT-FILE-ACCESS       PIC S9(04) VALUE 2 USAGE COMP.
CCT001WS 002300         03  CCT-FILE-PREV-OP      PIC S9(04) VALUE 0 USAGE COMP.
CCT001WS 002400 01  CCT-KEY-LENGTH          PIC S9(04) VALUE 15 USAGE COMP.
CCT001WS 002500 01  CCT-REC-LENGTH          PIC S9(04) VALUE 65 USAGE COMP.
CCT001WS 002600 01  CCT-KEY-LOCATION        PIC S9(04) VALUE  1 USAGE COMP.
CCT001WS 002700 01  HARDWIRED-LOOKUP-ERRORS.
CCT001WS 002800     02  CCT-KSAM-GENERAL-ERROR.
CCT001WS 002900         03  FILLER                    PIC X(34) VALUE
CCT001WS 003000             "ERROR 801 - KSAM FILE SYSTEM ERROR".
CCT001WS 003100         03  CCT-KSAM-ERROR-NBR        PIC 9(04).
CCT001WS 003200         03  FILLER                    PIC X(12) VALUE SPACES.
CCT001WS 003300     02  TERM-ACCESS-ERROR.
CCT001WS 003400         03  FILLER                    PIC X(34) VALUE
CCT001WS 003500             "Error 707 - Terminal Access Error ".
CCT001WS 003600         03  ERROR-CODE-1              PIC -9(04).
CCT001WS 003700         03  FILLER                    PIC X(11) VALUE SPACES.
CCT001WS 003800     02  FORM-FILE-ERROR.
CCT001WS 003900         03  FILLER                    PIC X(12) VALUE
CCT001WS 004000             "Error 701 - ".
CCT001WS 004100         03  FORM-FILE-MSG             PIC X(14).
CCT001WS 004200         03  FILLER                    PIC X(24) VALUE
CCT001WS 004300             " is not a form file     ".
CCT001WS 004400     02  FILE-FORM-ERROR.
CCT001WS 004500         03  FILLER                    PIC X(28) VALUE
CCT001WS 004600             "Error 712 - Form File Error ".
CCT001WS 004700         03  ERROR-CODE-2              PIC -9(04).
CCT001WS 004800         03  FILLER                    PIC X(17) VALUE SPACES.
CCT001WS 004900     02  CCT-EMERGENCY.
CCT001WS 005000         03  FILLER                    PIC X(32) VALUE
CCT001WS 005100             "Error 924 - Error Msg not in CCT".
CCT001WS 005200         03  FILLER                    PIC X(18) VALUE SPACES.
CCT001WS 005300 01  CCT-ERROR-MSG                     PIC X(50).
CCT001WS 005400 01  CCT-MATCH-KEY                     PIC X(15)       VALUE SPACES
CCT001WS 005500 01  CCT-REDEFINE-KEY-1 REDEFINES CCT-MATCH-KEY.
CCT001WS 005600     03   CCT-ERROR-KEY                PIC X(08).
CCT001WS 005700     03   FILLER                       PIC X(07).
       027300 01  CCT-REDEFINE-KEY-2 REDEFINES CCT-MATCH-KEY.
       027400     03  CCT-TABLE-KEY                  PIC 9(03).
       027500     03  CCT-2-CHAR-KEY                 PIC X(02).
       027600     03  FILLER                         PIC X(10).
       027700 01  CCT-REDEFINE-KEY-4 REDEFINES CCT-MATCH-KEY.
       027800     03  FILLER                         PIC X(03).
       027900     03  CCT-4-CHAR-KEY                 PIC X(04).
       028000     03  FILLER                         PIC X(08).
```

```
046800*-----------------*-----------------------------------------*----------------
046900*-----------------*       INPUT ROUTINES                    *----------------
047000*-----------------*-----------------------------------------*----------------
047100*
047200*      *-------------------------------------------------------*
047300*       This routine will look up the necessary titles from
047400*       the following Common Code Table: 983, 985, and 987.
047500*       These titles will be used to prime the report headings.
047600*      *-------------------------------------------------------*
047700 P100-READ-CCT-TITLES.
047800      MOVE 983                          TO CCT-TABLE-KEY.
047900      MOVE "CG"                         TO CCT-2-CHAR-KEY.
048000      PERFORM P105-CCT-INPUT THROUGH P105-EXIT.
048100      IF  GEN-ERROR-FOUND
048200          MOVE 0                        TO GEN-ERROR-FLAG,
048300          MOVE SPACES                   TO CCT-VALUE.
048400      MOVE CCT-VALUE                    TO F482-SYSTEM.
048500      MOVE 985                          TO CCT-TABLE-KEY.
048600      MOVE "64"                         TO CCT-2-CHAR-KEY.
048700      PERFORM P105-CCT-INPUT THROUGH P105-EXIT.
048800      IF  GEN-ERROR-FOUND
048900          MOVE 0                        TO GEN-ERROR-FLAG,
049000          MOVE SPACES                   TO CCT-VALUE.
049100      MOVE CCT-VALUE                    TO F482-OFFICE.
049200      MOVE 987                          TO CCT-TABLE-KEY.
049300      MOVE "F482"                       TO CCT-4-CHAR-KEY.
049400      PERFORM P105-CCT-INPUT THROUGH P105-EXIT.
049500      IF  GEN-ERROR-FOUND
049600          MOVE 0                        TO GEN-ERROR-FLAG,
049700          MOVE SPACES                   TO CCT-VALUE.
049800      MOVE CCT-VALUE                    TO F482-TITLE-LITERAL.
049900 P100-EXIT.
050000      EXIT.
050100 P105-DUMMY. COPY CCT105P2.
CCT105P2 001000*      *-------------------------------------------------------*
CCT105P2 001100*        This routine will read the Common Code Table based
CCT105P2 001200*        on the key fields defined during data field edits.
CCT105P2 001300*      *-------------------------------------------------------*
CCT105P2 001400 P105-CCT-INPUT.
CCT105P2 001500      MOVE ZERO TO GEN-ERROR-FLAG.
CCT105P2 001600      CALL "CKREADBYKEY" USING CCT-FILE-TABLE
CCT105P2 001700                               KSAM-FILE-STATUS
CCT105P2 001800                               CCT-BASIC-RECORD
CCT105P2 001900                               CCT-MATCH-KEY
CCT105P2 002000                               CCT-KEY-LOCATION
CCT105P2 002100                               CCT-REC-LENGTH.
CCT105P2 002200      MOVE SPACES TO CCT-MATCH-KEY.
CCT105P2 002300      IF  NOT KSAM-SUCCESSFUL-OPERATION
CCT105P2 002400          MOVE "???" TO CCT-VALUE,
CCT105P2 002500          MOVE 1 TO GEN-ERROR-FLAG,
CCT105P2 002600          GO TO P105-EXIT.
CCT105P2 002700 P105-EXIT.
CCT105P2 002800      EXIT.
```

COMMON CODE TABLE 980
Error Codes

KEY	BASIC VALUE (Error Message)
98000101	Error 001 - Must be Alphabetic
98000102	Only alphabetic characters are allowed.
98000201	Error 002 - Must be alphabetic or space fill
98000202	Only alphabetic characters are allowed, as well as
98000203	spaces (to the right of the alpha data ONLY)
98000301	Error 003 - Must be alphanumeric or space-fill
98000302	Alphabetic, numeric, or spaces (including those
98000303	embedded in data)are allowed-NO special characters
98000401	Error 004 - Must be numeric
98000402	Only numeric characters are allowed.
98000501	Error 005 - Must be numeric or zero-fill
98000502	Numeric characters only, with the exception of
98000503	spaces before and/or after the data only, NOT
98000504	embedded. These spaces are converted to zeros.
98000601	Error 006 - Must be numeric within range
98000602	Only a range of numeric values are valid. See the
98000603	User Guide for this format for further information
98001001	Error 010 - No data has been entered.
98001002	ENTER has been pressed, but no data has been input
98001101	Error 011 - Must be entered as space or -
98002501	Error 025 - City must not be left blank
98002701	Error 027 - Zip Prefix invalid, see Table 910
98002801	Error 028 - Mail Flag Invalid
98002901	Error 029 - State Code invalid, see Tables 901/903
98003201	Error 032 - Telephone must be blank or numeric
98003202	The only valid combinations of blanks and numerics
98003203	are as follows: bbb-bbb-bbbb
98003204	bbb-nnn-nnnn
98003205	nnn-nnn-nnnn.
98006001	Error 060 - Name must not be left blank
98006101	Error 061 - Surname must be followed by a comma
98006201	Error 062 - Name must not be changed
98006202	The name is used as an IMAGE/3000 search item
98006203	You MUST NOT change its value while correcting
98006204	other data on the screen.
98006501	Error 065 - Data item must NOT be changed
98006901	Error 069 - Action must NOT be changed
98007001	Error 070 - Invalid ID
98007801	Error 078 - Major Connector must NOT be blank
98007901	Error 079 - Major Connector must be -, ;, or space
98008001	Error 080 - Social Security Number must be numeric

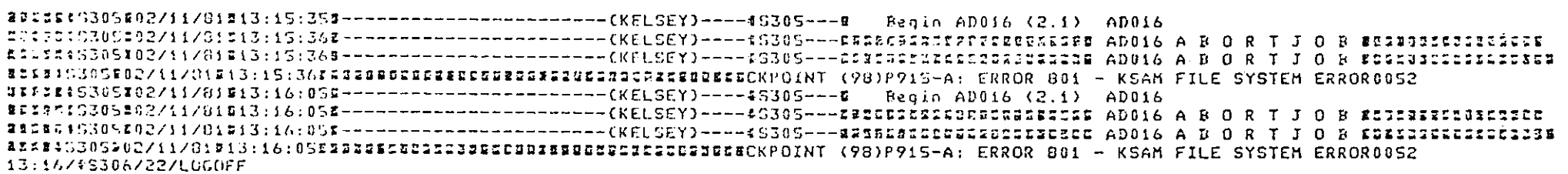
```
           060400 P615-DUMMY. COPY VEW615P2.
 :W615P2   001000*
VEW615P2   001100*     *-------------------------------------------------*
VEW615P2   001200*        This routine displays the CCT error message in the
VEW615P2   001300*        form name box on the Whitman Standard View screen.
VEW615P2   001400*     *-------------------------------------------------*
VEW615P2   001600 P615-DISPLAY-ERROR.
VEW615P2   001700      MOVE CCT-VALUE TO VIEW-MSG-BUFF.
VEW615P2   001800      MOVE VIEW-ERROR-ENHANCEMENT TO VIEW-MSG-ENHANCEMENT.
VEW615P2   002000      DISPLAY VIEW-MESSAGE-BUFFER.
VEW615P2   003000      PERFORM P322-VIEW-SHOWFORM THRU P322-EXIT.
VEW615P2   003100 P615-EXIT.
VEW615P2   003200      EXIT.
           060500 P617-DUMMY. COPY VEW617P2.
VEW617P2   001000*
VEW617P2   001100*     *-------------------------------------------------*
VEW617P2   001200*        This routine displays the form title in the form
VEW617P2   001300*        name box on the Whitman Standard View screen.
VEW617P2   001400*     *-------------------------------------------------*
VEW617P2   001600 P617-RESET-ERROR.
VEW617P2   001700      MOVE FORM-TITLE TO VIEW-MSG-BUFF.
VEW617P2   001800      MOVE VIEW-STANDARD-ENHANCEMENT TO VIEW-MSG-ENHANCEMENT.
VEW617P2   002000      DISPLAY VIEW-MESSAGE-BUFFER.
VEW617P2   003100      PERFORM P322-VIEW-SHOWFORM THRU P322-EXIT.
VEW617P2   003200 P617-EXIT.
VEW617P2   003300      EXIT.
VEW625P2   001100*     *-------------------------------------------------*
VEW625P2   001200*      This routine flags a particular field on the View/3000
VEW625P2   001300*      screen with an error condition.  If the CCT-ERROR-KEY
VEW625P2   001400*      is not spaces, a lookup is done to find the proper CCT
VEW625P2   001500*      error message and display it to the User's screen.
VEW625P2   001600*     *-------------------------------------------------*
VEW625P2   001800 P625-VIEW-SECONDARY-ERROR.
VEW625P2   001900      MOVE -1 TO VIEW-LENERRMSG.
VEW625P2   002000      CALL "VSETERROR" USING  VIEW-COM-AREA
VEW625P2   002100                              VIEW-FIELD-NUM
VEW625P2   002200                              VIEW-DUMMY-PARAM
VEW625P2   002300                              VIEW-LENERRMSG.
VEW625P2   002400      IF  NOT VIEW-OP-VALID
VEW625P2   002500          MOVE "925??P625A" TO GEN-ABORT-BREAKOUT.
VEW625P2   002600          PERFORM P658-VIEW-STANDARD-ERROR THRU P658-EXIT.
VEW625P2   002700          GO TO P099-ABORT.
VEW625P2   002800      IF  CCT-ERROR-KEY EQUAL TO SPACES
VEW625P2   002900          GO TO P625-EXIT.
VEW625P2   003000      PERFORM P680-ERROR-LOOKUP THRU P680-EXIT.
VEW625P2   003100      PERFORM P615-DISPLAY-ERROR THRU P615-EXIT.
VEW625P2   003200 P625-EXIT.
VEW625P2   003300      EXIT.
KSM680P2   001000*     *-------------------------------------------------*
KSM680P2   001100*      This routine will look up the appropriate error
KSM680P2   001200*      message on the CCT file (UT145K01) on request.
KSM680P2   001300*     *-------------------------------------------------*
KSM680P2   001400 P680-ERROR-LOOKUP.
KSM680P2   001500      MOVE 0 TO GEN-ERROR-FLAG.
KSM680P2   001600      PERFORM P105-CCT-INPUT THROUGH P105-EXIT.
KSM680P2   001700      IF  GEN-ERROR-FOUND
KSM680P2   001800          MOVE CCT-EMERGENCY TO CCT-ERROR-MSG.
KSM680P2   001900          GO TO P680-EXIT.
KSM680P2   002000      MOVE CCT-VALUE TO CCT-ERROR-MSG.
KSM680P2   002100 P680-EXIT.
KSM680P2   002200      EXIT.
```

```
       026800 01  CONSOLE-COPY-DUMMY          COPY GEN022WS.
GEN022WS 001000     .
GEN022WS 001100     03  COPY-CONTROL-BYTE              PIC X(01).
GEN022WS 001200*     *-------------------------------------------------*
GEN022WS 001300*                    Standard ABORT Format Area
GEN022WS 001500*     *-------------------------------------------------*
GEN022WS 001600 01  GEN-ABORT-BREAKOUT.
GEN022WS 001700         03  GEN-ABORT-ERROR           PIC X(03).
GEN022WS 001800         03  GEN-ABORT-SECTION         PIC X(02).
GEN022WS 001900         03  GEN-ABORT-PARAGRAPH       PIC X(04).
GEN022WS 002000         03  GEN-ABORT-POINT           PIC X(01).
GEN022WS 002100 01  GEN-ABORT-MESSAGE.
GEN022WS 002200         03  FILLER                    PIC X(09) VALUE "CKPOINT ("
GEN022WS 002300         03  GEN-ABORT-SECTION         PIC X(02).
GEN022WS 002400         03  FILLER                    PIC X(01) VALUE ")".
GEN022WS 002500         03  GEN-ABORT-PARAGRAPH       PIC X(04).
GEN022WS 002600         03  FILLER                    PIC X(01) VALUE "-".
GEN022WS 002700         03  GEN-ABORT-POINT           PIC X(01).
GEN022WS 002800         03  FILLER                    PIC X(02) VALUE ": ".
GEN022WS 002900         03  GEN-ABORT-LITERAL         PIC X(50) VALUE SPACES.
GEN022WS 003000 01  GEN-ABORT-KEY.
GEN022WS 003100         03  FILLER                    PIC X(03) VALUE "980".
GEN022WS 003200         03  GEN-ABORT-ERROR-KEY       PIC X(03).
GEN022WS 003300         03  FILLER                    PIC X(09) VALUE "01      "


       095900*     *-------------------------------------------------*
       096000*       This routine reads the WID-MST based on the WID-VALUE
       096100*       from the STUDENT-DTL.
       096200*     *-------------------------------------------------*
       096300 P130-READ-WID-MST.
       096400     MOVE STU-WHITMAN-ID OF STU012L2-AREA
       096500          TO CDB-WID-VALUE, STU-WID-VALUE.
       096600     PERFORM P165-GET-WID-MST THROUGH P165-EXIT.
       096700     IF  IMAGE-RECORD-NOT-FOUND
       096800         MOVE 0 TO CDB-ID-PREFIX,
       096900         GO TO P130-EXIT.
       097000     IF  NOT IMAGE-OP-VALID
       097100         MOVE "92548P130A" TO GEN-ABORT-BREAKOUT,
       097200         GO TO P130-ERROR-OUT.
       097300     GO TO P130-EXIT.
       097400 P130-ERROR-OUT.
       097500     PERFORM P645-IMAGE-STANDARD-ERROR THROUGH P645-EXIT.
       097600     GO TO P099-ABORT.
       097700 P130-EXIT.
       097800     EXIT.
```

```
          121400 P645-DUMMY. COPY IMG645P2.
IMG645P2  001000*     *-------------------------------------------------*
IMG645P2  001100*                  Standard IMAGE/3000 ABORT routine
IMG645P2  001200*     *-------------------------------------------------*
IMG645P2  001300 P645-IMAGE-STANDARD-ERROR.
IMG645P2  001400      MOVE CORR GEN-ABORT-BREAKOUT TO GEN-ABORT-MESSAGE.
IMG645P2  001500      MOVE GEN-ABORT-ERROR TO GEN-ABORT-ERROR-KEY.
IMG645P2  001600      MOVE GEN-ABORT-KEY TO CCT-ERROR-KEY.
IMG645P2  001700      PERFORM P680-ERROR-LOOKUP THROUGH P680-EXIT.
IMG645P2  001800      MOVE CCT-VALUE TO GEN-ABORT-LITERAL.
IMG645P2  001900      DISPLAY " X H J".
IMG645P2  002000      DISPLAY GEN-ABORT-MESSAGE.
IMG645P2  002100      MOVE SPACES TO IMAGE-ERROR-MSG.
IMG645P2  002200      CALL "DBERROR"   USING IMAGE-STATUS,
IMG645P2  002300                             IMAGE-ERROR-MSG,
IMG645P2  002400                             IMAGE-LENERRBUF.
IMG645P2  002500      CALL "DBEXPLAIN" USING IMAGE-STATUS.
IMG645P2  002600      MOVE IMAGE-ERROR-MSG TO GEN-ABORT-LITERAL.
IMG645P2  002700      MOVE GEN-ABORT-MESSAGE TO GEN-TELL-ABORT-MSG.
IMG645P2  002800      CALL "TELLABORT" USING GEN-TELL-JOB-NAME
IMG645P2  002900                             GEN-TELL-USER,
IMG645P2  003000                             GEN-TELL-DUMMY,
IMG645P2  003100                             GEN-TELL-DUMMY,
IMG645P2  003200                             GEN-TELL-ABORT-MSG.
IMG645P2  003300      GO TO P645-EXIT.
IMG645P2  003400 P645-EXIT.
IMG645P2  003500      EXIT.
          121500 P680-DUMMY. COPY KSM680P2.
KSM680P2  001000*     *-------------------------------------------------*
KSM680P2  001100*      This routine will look up the appropriate error
KSM680P2  001200*      message on the CCT file (UT145K01) on request.
KSM680P2  001300*     *-------------------------------------------------*
KSM680P2  001400 P680-ERROR-LOOKUP.
KSM680P2  001500      MOVE 0 TO GEN-ERROR-FLAG.
KSM680P2  001600      PERFORM P105-CCT-INPUT THROUGH P105-EXIT.
KSM680P2  001700      IF  GEN-ERROR-FOUND
KSM680P2  001800          MOVE CCT-EMERGENCY TO CCT-ERROR-MSG.
KSM680P2  001900          GO TO P680-EXIT.
KSM680P2  002000      MOVE CCT-VALUE TO CCT-ERROR-MSG.
KSM680P2  002100 P680-EXIT.
KSM680P2  002200      EXIT.
          121600 P999-END-SECTION-05.
          121700      EXIT.

    DATA AREA IS %006303 WORDS.
    CPU TIME = 0:01:06.  WALL TIME = 0:03:35.
END COBOL/3000 COMPILATION.  NO ERRORS.  NO WARNINGS.
```



13:14/#S306/22/LOGOFF
13:17/#S308/29/LOGON FOR: GALPIN,MANAGER.WCCS,PUB ON LDEV #23
13:20/15/STANDARD FORMS ON LDEV#6
13:20/#S280/45/LOGOFF
?13:20/15/SP#6/IS #S300; F969WIDE ON LDEV#6 (Y/N)?
13:20:47#02/11/81##----------#S300-----#
13:20/#S309/45/LOGON FOR: OH,PAINE.WCCS,P4 ON LDEV #46
OK
REPLY 15,Y
:
13:21/#S308/29/LOGOFF

```
#S100#03/02/81#13:17:20#----------(POLZIN)----#S100---#   Begin ML705 (1.0) .ML705
13:17/#S102/33/LOGOFF
SC

DEV/CL    DFID    JOBNUM  FNAME    STATE FRM SPACE RANK PRI #C
COMLP     #0206   #J32    $STDLIST OPENED      1024       8   1
COMLP     #0241   #S94    F969WIDE OPENED      1024       8   1
COMLP     #0125   #S38    F260SPCL READY  F       8     D 1   1
COMLP     #0242   #S94    F222SPCL OPENED F    1024     D 1   1

4 FILES (DISPLAYED):
    0 ACTIVE
    1 READY; INCLUDING 1 SPOOFLES, 1 DEFERRED
    3 OPENED; INCLUDING 3 SPOOFLES
    0 LOCKED; INCLUDING 0 SPOOFLES
    4 SPOOFLES:  3000 SECTORS
OUTFENCE = 6
OUTFENCE = 10 FOR LDEV 24
.
#S94#03/02/81#13:23:24#----------(COMFORAB)--#S94----#                          #0241          STD WIDE (---F969
#S94#03/02/81#13:23:24#----------(COMFORAB)--#S94----#                          #0242   8.5 X 11 WHITE (---F222
13:23/15/STANDARD FORMS ON LDEV#6
#S94#03/02/81#13:23:25#----------(COMFORAB)--#S94----#    End AD110 (2.2)
?13:23/15/SP#6/IS #S94; F969WIDE ON LDEV#6 (Y/N)?
#S94#03/02/81#13:23:40#----------(COMFORAB)--#S94----#   Begin AD105 (2.0)  AD105
13:24/#S103/33/LOGON FOR: DIETZ,MGR.CONTACT,G3K ON LDEV #45
13:25/12/MISSING USER FOR "MATH.CLASS,        " ON LDEV "50"
OK
REPLY 15,Y
:
#S100#03/02/81#13:26:09#----------(POLZIN)----#S100---#    End ML705 (1.0)  ML705
13:26/#J36/40/LOGON FOR: ML107CPL,KELSEY.WCCS,K1 ON LDEV #10
13:26/#S100/30/LOGOFF
13:27/#S104/31/LOGON FOR: CL408,NOEL.WCCS,N2 ON LDEV #23
13:20/#J36/40/LOGOFF
13:32/#S105/43/LOGON FOR: RICHMOND,ECON39.CLASS,ECON39 ON LDEV #27
13:32/#J37/29/LOGON FOR: CL408CPL,NOEL.WCCS,N2 ON LDEV #10
```

```
001000$CONTROL USLINIT,SOURCE
001100$TITLE ">>    Senior Rank in Class Update      <<            ",&
0012008"                                  CG228 <1.2>"
001300 IDENTIFICATION DIVISION.
001400 PROGRAM-ID.                        "CG228 <1.2>".
001500 DATE-WRITTEN.                       JUNE, 1979.
001600 INSTALLATION.                       WHITMAN COLLEGE.
001700 SECURITY.                           CONFIDENTIAL.
001800*
001900*     *-------------------------------------------------*
002000*     |              >> NOTICE OF COPYRIGHT <<          |
002100*     *-------------------------------------------------*
002200*     |    CG228 is Whitman College Proprietary software |
002300*     *=================================================*
002400*     | Programming    | Systems Analysis  | Systems Design |
002500*     *-------------------------------------------------*
002600*     | Delores Payne  | Delores Payne     | Wayne Holt     |
002700*     *-------------------------------------------------*
002800*
002900*     *-------------------------------------------------*
003000*      SUMMARY:  CG228 ranks graduating seniors by cumulative
003100*                GPA, and updates the Registrar Student Detail
003200*                with their rank and class size when an update
003300*                run is requested.  Only seniors that satisfy
003400*                the following criteria will be ranked:
003500*
003600*                     * STU-STUDENT-LEVEL of 9,
003700*                     * STU-STUDENT-STATUS of B3, B4, B5, B6,
003800*                                             B8, C2, or C3,
003900*                     * PROJ-GRAD-DATE of PARAM-WINTER-DATE,
004000*                                         PARAM-SPRING-DATE,
004100*                                         PARAM-FALL-DATE.
004200*
004300*                A standard narrow report, F482, is produced
004400*                of those seniors meeting the above criteria
004500*                and is sorted by descending cumulative GPA.
004600*
004700*                (1)  File Descriptions:
004800*                     CG228S16   Parameters for CG228
004900*                     UT145K01   Common Code Table
005000*                     COMMON     Data Base Common
005100*                     STU        Student Records Data Base
005200*                     F482NARO   Report Print File
005300*
005400*                (2)  Subroutine calls:
005500*                     UT817      Obtains JOB/USER Name
005600*
005700*     *-------------------------------------------------*
```

```
005900*     *-------------------------------------------------*
006000*      Compilation Instructions:
006100*
006200*           :FILE P;DEV=COMLP
006300*           :FILE COPYLIB=COBOLIB.PUB.WCCS
006400*           :COBOL CG228C.SOURCE.IRIS,,*P
006500*           :PREP $OLDPASS,CG228.REGSTRAR.ADMIN;MAXDATA=8000
006600*
006700*     *-------------------------------------------------*
006800*
006900*     *-------------------------------------------------*
007000*      DATE      REV   BY    MAINTENANCE LOG
007100*      6/29/79 <1.0> DRP Primary Installation of software.
007200*
007300*      6/09/80 <1.1> DRP Added statuses B8 and C3 to the list
007400*                        of valid senior student status codes.
007500*
007600*      6/12/80 <1.2> DRP Brought up to programming standards
007700*                        established May 30, 1980.
007800*     *-------------------------------------------------*
```

```
008000*-------------------*-------------------------------*-----------------*
008100*------------------* SYNOPSIS OF PROGRAM LOGIC *-----------------*
008200*-------------------*-------------------------------*-----------------*
008300*
008400*   LOGIC DESCRIPTION;
008500*
008600*  I.  MAINLINE SECTION.
008700*
008800*      A. Displays the program number and revision level.
008900*      B. Performs all file opens and initial reads:
009000*         1. Parameter file,
009100*         2. CCT KSAM file,
009200*         3. IMAGE files,
009300*         4. Report files,
009400*      C. Performs the TELL-START routine.
009500*      D. Performs the LEGEND SECTION which produces a front
009600*         page legend on the report identifying the parameters
009700*         chosen by the User.
009800*      E. Performs the SORT-LOGIC SECTION.
009900*      F. Sorts the students by descending CUM-GPA.
010000*      G. Performs the REPORT-LOGIC SECTION.
010100*      H. Performs file closes:
010200*         1. CCT KSAM files,
010300*         2. Parameter file,
010400*         3. Report files,
010500*         4. IMAGE files.
010600*      I. Print a last page message on the report, CG228/F482.
010700*      J. Perform the TELL-FORM routine which posts a forms
010800*         message to the system console for a STD NARROW RPT.
010900*      K. Performs the TELL-STOP routine.
011000*      L. Stops the program.
011100*
011200* II.  SORT-LOGIC SECTION.
011300*
011400*      A. Serially reads the Registrar Student Detail on the
011500*         STU Data Base (REG-STUDENT-DET).
011600*         1. Checks for valid senior STU-STUDENT-LEVEL (9 only)
011700*            and STU-STUDENT-STATUS (B3-B6,B8,C2, and C3).
011800*         2. Checks for acceptable Projected Graduation Dates
011900*            input by the User in the paramater interface:
012000*            a. WINTER-DATE - December graduates,
012100*            b. SPRING-DATE - Spring graduates,
012200*            c. FALL-DATE   - Fall graduates.
012300*         3. If the student does NOT satisfy the above criteria,
012400*            another REG-STUDENT-DET will be read (return to A).
012500*      B. Perform a calculated read on the WID-MST on the COMMON
012600*         Data Base to obtain the ID-PREFIX for the WHITMAN-ID.
012700*      C. Perform a backwards chained read on the SEM-CREDIT-DTL
012800*         on the STU Data Base to obtain the record matching the
012900*         input academic term (contains the lastest grade data).
013000*      D. If the student does not have the correct SEM-CREDIT-DTL,
013100*         an error is flagged and another student record is read
013200*         (return to A).
013300*      E. The student has passed all edits so the necessary data
013400*         is now moved to the SORT-REC, and the record is released
013500*         to the COBOL SORT.
013600*      F. Repeat steps A through E until all students are read.
```

```
018700 ENVIRONMENT DIVISION.
018800 CONFIGURATION SECTION.
018900 SOURCE-COMPUTER.                    HP-3000.
019000 OBJECT-COMPUTER.                    HP-3000.
019100 SPECIAL-NAMES.
019200     TOP IS TOP-OF-PAGE,
019300     CONSOLE IS MASTER-CONSOLE.
019400 INPUT-OUTPUT SECTION.
019500 FILE-CONTROL.
019600     SELECT PRINT-FILE   ASSIGN TO "F482NARO,UR,,CONLP(CCTL)".
019700     SELECT PARAM-FILE   ASSIGN TO "CG228S16".
019800     SELECT SORT-FILE    ASSIGN TO "SORTFILE,DA".
```

```
           024300 WORKING-STORAGE SECTION.
           024400 01  GEN-COPY-DUMMY                  COPY GEN021WS.
GEN021WS   001000      .
GEN021WS   001100      03  COPY-CONTROL-BYTE                PIC X(01).
GEN021WS   001200*      *-------------------------------------------------*
GEN021WS   001300*                  General Constants and Variables
GEN021WS   001400*              Console Communication and Message Parameters
GEN021WS   001500*      *-------------------------------------------------*
GEN021WS   001600 01  GEN-ERROR-FLAG                   PIC 9(01) VALUE 0.
GEN021WS   001700          88  GEN-ERROR-FOUND                    VALUE 1.
GEN021WS   001800 01  GEN-EDIT-FLAG                    PIC 9(01) VALUE 0.
GEN021WS   001900          88  GEN-NO-DATA-EDITED                 VALUE 0.
GEN021WS   002000 01  GEN-LENGTH                       PIC S9(04) COMP.
GEN021WS   002100 01  GEN-RUN-TIME.
GEN021WS   002200          03  GEN-RUN-HOUR             PIC 9(02).
GEN021WS   002300          03  GEN-RUN-MIN              PIC 9(02).
GEN021WS   002400          03  GEN-RUN-SEC              PIC 9(02).
GEN021WS   002500 01  GEN-JOB-SESSION-INFORMATION.
GEN021WS   002600          03  GEN-JOB-SESSION-NBR      PIC X(06).
GEN021WS   002700          03  GEN-JOB-NUMBER-HYPH      PIC X(06).
GEN021WS   002800          03  GEN-JOB-NUMBER-BLANK     PIC X(06).
GEN021WS   002900          03  GEN-JOB-NAME             PIC X(08).
GEN021WS   003000          03  GEN-JOB-USER-NAME        PIC X(08).
GEN021WS   003100          03  GEN-JOB-COMBINED-NAME    PIC X(15).
GEN021WS   003200          03  GEN-JOB-LDEV             PIC S9(04) COMP SYNC.
GEN021WS   003300 01  GEN-CONSOLE-TELL-INFORMATION.
GEN021WS   003400          03  GEN-TELL-JOB-NAME        PIC X(06).
GEN021WS   003500          03  GEN-TELL-USER            PIC X(08).
GEN021WS   003600          03  GEN-TELL-PROGRAM         PIC X(12).
GEN021WS   003700          03  GEN-TELL-OUTPUT-NBR      PIC X(08).
GEN021WS   003800          03  GEN-TELL-FORM-NAME       PIC X(04).
GEN021WS   003900          03  GEN-TELL-FORM-MESSAGE    PIC X(14).
GEN021WS   004000          03  GEN-TELL-DUMMY           PIC X(02).
GEN021WS   004100          03  GEN-TELL-ABORT-MSG       PIC X(70).
           024500 01  GEN-SORT-COUNT                   PIC 9(03) VALUE 0.
           024600 01  GEN-WORK-WID.
           024700      03  GEN-WORK-WID-PREFIX          PIC 9(03).
           024800      03  GEN-WORK-WID-SUFFIX          PIC X(06).
           024900 01  GEN-FIRST-TIME-FLAG              PIC 9(02) VALUE 0.
           025000          88  GEN-FIRST-TIME                     VALUE 0.
           025100*      *-------------------------------------------------*
           025200*                          Hold Area
           025300*      *-------------------------------------------------*
           025400 01  HOLD-NEW-RANK                    PIC 9(03) VALUE 0.
           025500 01  HOLD-SAME-RANK                   PIC 9(03) VALUE 0.
           025600 01  HOLD-STUDENT-LEVEL               PIC X(01).
           025700          88  HOLD-VALID-SENIOR-LEVEL            VALUE "9".
           025800 01  HOLD-CUM-GPA                     PIC 9V999 COMP.
           025900 01  HOLD-STUDENT-STATUS              PIC X(02).
           026000          88  HOLD-VALID-SENIOR-STATUS           VALUE "B3" "B4"
           026100                                                  "B5" "B6"
           026200                                                  "B8" "C2"
           026300                                                  "C3".
           026400 01  HOLD-PROJ-DATE.
           026500      03  HOLD-PROJ-MO                 PIC X(02).
           026600      03  HOLD-PROJ-YR                 PIC X(02).
```

```
042900 PROCEDURE DIVISION.
043000*         >>>>>>>>>>-------------------------------<<<<<<<<<<
043100                          MAINLINE SECTION 98.
043200*         >>>>>>>>>>-------------------------------<<<<<<<<<<
043300*
043400*      *---------------------------------------------------*
043500*        SUMMARY: This Section of code is designed to be
043600*        the Main Driver for all program logic modules, It
043700*        contains all major file I/O overhead and general
043800*        housekeeping routines, and is the primary module.
043900*      *---------------------------------------------------*
044000*
044100 P015-INITIALIZE.
044200      DISPLAY "CG228 <1,2>".
044300      PERFORM P912-OPEN-PARAM-FILE THROUGH P912-EXIT.
044400      PERFORM P131-READ-PARAM-FILE THROUGH P131-EXIT.
044500      PERFORM P915-OPEN-KSAM-FILES THROUGH P915-EXIT.
044600      PERFORM P100-READ-CCT-TITLES THROUGH P100-EXIT.
044700      PERFORM P975-OPEN-IMAGE-BASES THROUGH P975-EXIT.
044800      PERFORM P951-ESTABLISH-IMAGE-LISTS THROUGH P951-EXIT.
044900      PERFORM P917-OPEN-REPORT THROUGH P917-EXIT.
045000      PERFORM P995-TELL-START THROUGH P995-EXIT.
045100 P015-MAINLINE.
045200      PERFORM P030-LEGEND-DRIVER THROUGH P999-END-SECTION-25.
045300      SORT SORT-FILE
045400           ON DESCENDING KEY SORT-CUM-GPA,
045500              INPUT PROCEDURE IS SORT-LOGIC
045600              OUTPUT PROCEDURE IS REPORT-LOGIC.
045700 P090-STOP.
045800      PERFORM P925-CLOSE-KSAM-FILES THROUGH P925-EXIT.
045900      PERFORM P965-CLOSE-PARAM-FILE THROUGH P965-EXIT.
046000      PERFORM P969-CLOSE-REPORT-FILE THROUGH P969-EXIT.
046100      PERFORM P985-CLOSE-CDB THROUGH P985-EXIT.
046200      PERFORM P996-TELL-STOP THROUGH P996-EXIT.
046300      STOP RUN.
046400 P099-ABORT.
046500      CALL "ABORTRUN".
```

```
075300*----------------*-------------------------------------*----------------*
075400*----------------*       OUTPUT ROUTINES               *----------------*
075500*----------------*-------------------------------------*----------------*
075600*
075700*     *-----------------------------------------------------*
075800*      This routine will update the STU Data Base with the
075900*      rank in class for seniors and the size for that class.
076000*      The data will be put on the REG-STUD-DTL only for a
076100*      live run.  Otherwise this paragraph will be omitted.
076200*     *-----------------------------------------------------*
076300 P220-UPDATE-RECORD.
076400     MOVE SORT-WID                 TO GEN-WORK-WID.
076500     MOVE GEN-WORK-WID-SUFFIX      TO STU-WID-VALUE.
076600     MOVE SORT-WID                 TO STU-WID-VALUE.
076700 P220-DBLOCK.
076800     PERFORM P250-LOCK-STU THROUGH P250-EXIT.
076900     IF  NOT IMAGE-OP-VALID
077000         MOVE "92556P220A" TO GEN-ABORT-BREAKOUT,
077100         GO TO P220-ERROR-OUT.
077200 P220-DBFIND.
077300     PERFORM P160-FIND-STU-DTL THROUGH P160-EXIT.
077400     IF  IMAGE-WORD5-6 EQUAL TO 0
077500         GO TO P220-EXIT.
077600     IF  NOT IMAGE-OP-VALID
077700         MOVE "92556P220B" TO GEN-ABORT-BREAKOUT,
077800         GO TO P220-ERROR-OUT.
077900 P220-DBGET.
078000     PERFORM P165-READ-STU-DTL THROUGH P165-EXIT.
078100     IF  IMAGE-END-OF-CHAIN
078200         GO TO P220-EXIT.
078300     IF  NOT IMAGE-OP-VALID
078400         MOVE "92556P220C" TO GEN-ABORT-BREAKOUT,
078500         GO TO P220-ERROR-OUT.
078600 P220-DBUPDATE.
078700     MOVE HOLD-NEW-RANK  TO STU-CLASS-RANK OF STU012L2-AREA.
078800     MOVE GEN-SORT-COUNT TO STU-CLASS-SIZE OF STU012L2-AREA.
078900     PERFORM P270-UPDATE-STU-DTL THROUGH P270-EXIT.
079000     IF  NOT IMAGE-OP-VALID
079100         MOVE "92556P220D" TO GEN-ABORT-BREAKOUT,
079200         GO TO P220-ERROR-OUT.
079300 P220-DBUNLOCK.
079400     PERFORM P255-UNLOCK-STU THROUGH P255-EXIT.
079500     IF  NOT IMAGE-OP-VALID
079600         MOVE "92556P220E" TO GEN-ABORT-BREAKOUT,
079700         GO TO P220-ERROR-OUT.
079800     GO TO P220-EXIT.
079900 P220-ERROR-OUT.
080000     PERFORM P645-IMAGE-STANDARD-ERROR THROUGH P645-EXIT.
080100     GO TO P099-ABORT.
080200 P220-EXIT.
080300     EXIT.
```