AN ABSTRACT OF THE THESIS OF

Thomas P. Cavaiani for the degree of Doctor of Philosophy in Mathematics Education
presented May 12, 1988.

Title: An Investigation of the Relationship between Cognitive Style and the Diagnostic Skills of Novice COBOL Student Programmers

Abstract approved: Signature redacted for privacy.

Howard L. Wilson

For the past thirty years the majority of the research on the cognitive demands of learning computer programming has assumed that successful computer programmers display certain aptitudes. Some researchers indicate that studies linking different prerequisite profiles of abilities to different programming outcomes are needed. Such research might dispel many of these notions. There is evidence that personality variables are related to overall performance in computer programming. Only recently has this type of research been conducted. The purpose of this study was to investigate the influence of one personality variable, global-analytic cognitive style,
on one of the component skills of programming, debugging.

Thirty subjects were administered the Group Embedded Figures Test (GEFT) to determine their cognitive style. Instruments were developed to measure subjects' skills in diagnosing both syntactic and logic errors in COBOL programs. Weighting schemes, derived from results of previous research, were developed for scoring the diagnostic exams. Data were analyzed using Spearman's Rank Correlation Coefficient.

Results of statistical analysis failed to support hypotheses (at the .05 level of significance) stating that GEFT scores were related to subjects' ability to locate and correct either syntactic or logic errors in computer programs. This result appears to indicate that measures of individuals' cognitive style represent an unsatisfactory means for predicting success on one aspect of computer programming skill for novices. However at higher alpha levels the results indicate support of the hypotheses.
AN INVESTIGATION OF THE RELATIONSHIP
BETWEEN COGNITIVE STYLE AND THE DIAGNOSTIC SKILLS
OF NOVICE COBOL STUDENT PROGRAMMERS

by

Thomas P. Cavaiani

A Thesis
submitted to
Oregon State University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy

Completed May 12, 1988
Commencement June 1989
APPROVED:

[Signature redacted for privacy.]

Professor of Mathematics Education in charge of major

[Signature redacted for privacy.]

Head of Department of Science, Mathematics, and Computer Science Education

[Signature redacted for privacy.]

Dean of the School of Education

[Signature redacted for privacy.]

Dean of Graduate School

Date thesis is presented May 12, 1988
TABLE OF CONTENTS

Chapter

I. INTRODUCTION ............................................. 1
   Definition of Terms .................................. 3
   Identification of the Problem ...................... 6
   Rationale for the Study ............................. 7
   Need for the Study ................................. 10

II. COGNITIVE AND LEARNING STYLES ...................... 16
   Alternate Modes of Information Processing ........ 19
   Historical Development of Cognitive Styles ........ 24
   Visual Perception ................................... 25
   The Influence of Gestalt Psychology .............. 26
   Disembedding and Intellectual Functioning ....... 32
   The Influence of Linguistics on Cognitive Psychology .... 41
   Experimental Results on Linguistic Ambiguity ...... 42
   The Analytic/Global Cognitive Style Construct .... 43
   Developmental Factors ............................... 46
   Child Rearing Practices ............................ 48
   Cultural Variables .................................. 50

III. REVIEW OF THE LITERATURE ......................... 53

IV. METHOD .................................................. 65
   Operational Definitions ............................ 65
   Justification of Scoring Schemes .................. 68
   Hypotheses .......................................... 70
   Design .............................................. 71
   Instrumentation .................................... 73
   Experimental Procedure and Data Collection ...... 77

V. RESULTS .................................................. 79
   Preparation of Data ................................ 79
   Classification of Subjects ......................... 81
   Preliminary Analysis ............................... 83
   Correlation Analysis ................................ 94
   Supplemental Analysis ............................. 99

VI. DISCUSSION ............................................. 102
   Restructuring Tasks and Error Type ............... 102
   Educational Implications .......................... 104
   Discussion of Supplemental Analysis ............. 106
   Suggestions for Further Research .................. 107
BIBLIOGRAPHY .................................................. 110

APPENDIX A  COBOL Source Code - Syntactic
Error Diagnostic Tasks ......................... 142

APPENDIX B  COBOL Source Code - Logic
Error Diagnostic Tasks ......................... 152

APPENDIX C  Error Corrections for the
Diagnostic Tasks ................................. 195

APPENDIX D  Diagnostic Task Scores, GEFT Scores
and Experience Values ....................... 200
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Syntax Task - Scheme 1 (S1) by GEFT</td>
<td>84</td>
</tr>
<tr>
<td>2. Syntax Task - Scheme 2 (S2) by GEFT</td>
<td>85</td>
</tr>
<tr>
<td>3. Syntax Task - Scheme 3 (S3) by GEFT</td>
<td>86</td>
</tr>
<tr>
<td>4. Syntax Task - Scheme 4 (S4) by GEFT</td>
<td>87</td>
</tr>
<tr>
<td>5. Syntax Task - Scheme 5 (S5) by GEFT</td>
<td>88</td>
</tr>
<tr>
<td>6. Logic Task - Scheme 1 (L1) by GEFT</td>
<td>89</td>
</tr>
<tr>
<td>7. Logic Task - Scheme 2 (L2) by GEFT</td>
<td>90</td>
</tr>
<tr>
<td>8. Logic Task - Scheme 3 (L3) by GEFT</td>
<td>91</td>
</tr>
<tr>
<td>9. Logic Task - Scheme 4 (L4) by GEFT</td>
<td>92</td>
</tr>
<tr>
<td>10. Logic Task - Scheme 5 (L5) by GEFT</td>
<td>93</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1. A Comparison of Mean Performance of Diagnostic Tasks and GEFT Scores for those with Experience and those without Experience</td>
<td>82</td>
</tr>
<tr>
<td>2. Spearman Correlation Coefficients GEFT versus Diagnostic Task Scores</td>
<td>95</td>
</tr>
<tr>
<td>3. Spearman Correlation Coefficients for Several Variables</td>
<td>100</td>
</tr>
</tbody>
</table>
AN INVESTIGATION OF THE RELATIONSHIP BETWEEN COGNITIVESTYLE AND THE DIAGNOSTIC SKILLS OF NOVICE COBOLSTUDENT PROGRAMMERS

CHAPTER I

INTRODUCTION

According to Pea & Kurland (1983), the majority of research during the last thirty years concerning the cognitive demands of learning computer programming has assumed that potential programmers must exhibit a certain configuration of abilities to be successful. "These studies always ask whether particular aptitude variables have an effect on programming success rather than the more fundamental question of how they might have such effects, in terms of component skills or knowledge representations mediating specific programming activities" (Pea & Kurland, 1983, p. 4). Such research ignores the complexity of the cognitive demands of learning computer programming by treating programming as a unitary skill. Like reading, computer programming is comprised of different abilities that interrelate with the organization of the learner's knowledge base, memory, and processing capacities. Research showing that different prerequisite profiles of abilities are necessary for different programming outcomes would dispel many of the notions previously
associated with required aptitudes for student programmers.

Only recently has research on the cognitive requirements of learning to program been conducted. Although the findings are limited such research provides evidence that certain personality variables are related to performance in computer programming.

The purpose of this study is to investigate the relationship between one personality variable, analytic-global cognitive style, and one component skill of programming, diagnostic skill, for programmers involved in a first college course in COBOL. Specifically, this study will:

1. Review the relevant literature in the areas of the component skills and subtasks of programming and cognitive style, and develop testable hypotheses concerning the relationship between cognitive style and diagnostic skills.

2. Develop an experimental design, procedure, stimulus material, and evaluative techniques for such material to be used to study the influence of cognitive style on diagnostic skill.

3. Experimentally test the set of hypotheses on a group of student programmers enrolled in a first college-level course in COBOL.
Definition of Terms

1. **Analytic/Global Cognitive Style Construct** - the capacity to overcome embedding contexts in perception, that is, to separate a part from an organized whole. Individuals who overcome embedding contexts are designated as having an analytic cognitive style, while individuals who do not overcome embedding contexts are designated as having a global cognitive style. This construct has assumed many different labels during the period in which the theory of cognitive styles has been developed and refined. In addition to the label analytic/global, others such as field independence/dependence, field articulation, and psychological differentiation have been used (Witkin and Goodenough, 1981).

2. **Cognitive restructuring** - the ability to analyze the initially apprehended organization of a problem, and perhaps to change it in order to identify possible solutions for the problem.

3. **Cognitive style** - characteristic, self-consistent modes of functioning which individuals show in their perceptual and intellectual activities (Witkin, Oltman, Raskin, & Karp, 1971); a hypothetical construct that has been developed to explain the process of mediation between stimuli and responses in terms of characteristic ways in which individuals
conceptually organize the environment (Goldstein & Blackman, 1976).

4. **Computer programming** - a set of activities involved in developing a reusable product consisting of a series of written instructions that make a computer accomplish some task (Pea & Kurland, 1983).

5. **Diagnostic skill** - the ability to analyze existing code that has errors, locate the errors, and correct them (Collins & White, 1984; Shneiderman, 1976). Diagnostic skill is sometimes referred to as "debugging skill" by different authors. Both terms will be assumed to be equivalent.

6. **Logic error** - an error that occurs when a computer program runs but not as planned (Wertz, 1982). These errors are the result of either incorrect applications of program statements with regard to the semantics of the statements or incorrect transformations (programmer misconceptions) from the program specifications to program statements (Shneiderman & Mayer, 1979).

7. For purposes of this study a **novice student programmer** will be defined as an individual who (1) has at most one semester of credit in not more than two different programming languages at the college level, (2) has less than one year of job related
computer programming experience, and (3) is studying a given computer programming language for the first time. The rationale for this definition is provided by studies (Atwood & Ramsey, 1978; Brooks, 1977; Jeffries, 1982) comparing expert and novice programmers indicating that expert programmers remember larger, more meaningful chunks of information that enable them to perceive and remember programs more effectively than do novice programmers. Further evidence (Wescourt & Hemphill, 1978) indicates that this expertise is derived from specific knowledge acquired through experience.

8. Restructuring ability - the tendency for people to deal with the visual field in either an active or passive manner by imposing varying degrees of organization upon the field during problem-solving of visual-perceptual tasks.

9. Restructuring task - a task that requires cognitive restructuring ability to solve. In other words, a type of problem that requires an individual to analyze the initially apprehended organization of the problem and perhaps to change it in order to identify possible solutions for the problem.

10. Style - a term that refers to regularities and consistencies of behavior over a wide variety of situations; this term is used by a number of
personality theorists in a manner synonymous with the term 'trait'. However, in most cases 'style' is used in conjunction with those traits that are concerned with perceptual, cognitive, or expressive behavior (Allport, 1961; Wallach, 1962).

11. **Syntactic error** - a deviation from the rules governing the allowable forms of computer program statements (Wertz, 1982). These types of errors are trivial by nature, can be recognized by a compiler, and resolved by reference to a programming manual. Thus, they are not considered as errors that should challenge the cognitive skills of programmers (Shneiderman and Mayer, 1979).

**Identification of the Problem**

Studies (Adejumo, 1983; Cheney, 1980; Goodman, 1971; Katz & Postal, 1964; Lefever & Ehri, 1976; MacKay & Bever, 1967; Tillman, 1979) have shown that subjects identified as having a global cognitive style have greater difficulty solving problems classified as restructuring tasks than subjects identified as having an analytic cognitive style. Examples of such research come from computer and non-computer fields of study (Cheney, 1980; Goodman, 1971; Katz & Postal, 1964; Lefever & Ehri, 1976; Snow, 1980; Testa, 1973).
The task of locating and correcting logic errors in computer programs qualifies as a restructuring task since it requires considerable restructuring skill to reformulate an initial program plan to eliminate faulty program performance (Collins & White, 1984; Shneiderman & Mayer, 1979; Witkin & Goodenough, 1981). On the other hand, the task of locating and correcting syntactic errors does not qualify as a restructuring task since use of restructuring skills is minimal due to the trivial nature of such errors. Thus, one would expect that cognitive style differences would be related to locating and correcting logic errors but not syntactic errors (Goodman, 1971; Shneiderman & Mayer, 1979).

The research question for this study is: Are the cognitive styles of novice student programmers related to their skills in diagnosing non-trivial program errors?

Rationale for the Study

Research findings indicate that the cognitive demands of programming are quite similar to those for reading (Pea & Kurland, 1983). Typically, in psychological research in computer programming, "learning to program" has been equated with learning program language syntax and commands, just as
"learning to read" has been equated with decoding skills. However, once past an initial level of skill acquisition, what is meant by "reading" is actually reading comprehension, which entails an elaborate corpus of cultural knowledge, comprehension, monitoring, inferencing, hypothesis generation, and other cognitive and metacognitive strategies that take years to master. Skilled computer programming is similarly complex, implying that an array of subtasks for "learning to program" exists.

As is presently the case for programming, early research in developmental psycholinguistics was concerned with a cognitive prerequisite controversy. Rather than asking if language development requires prior concept development for the ideas expressed in language, current language development research recognizes that such a question must be asked for specific language constructions or for subparts of language, and that the answer will depend on the linguistic forms chosen (Johnston, 1982).

It is the consensus of cognitive psychologists (Bonar, 1982; Ehrlich & Soloway, 1983; Johnson, Draper & Soloway, 1983; Kahney & Eisenstadt, 1982; Shneiderman & Mayer, 1979; Soloway & Ehrlich, 1982; Soloway, Ehrlich, Bonar & Greenspan, 1982) who have developed theories of expert programming skill that
computer programming is highly complex since "it requires subtasks that draw on different knowledge domains and a variety of cognitive processes" (Pennington, 1982, p. 11). Just as in the case of theories of problem solving in general, theories that have been developed for expert programming articulate a set of distinctive cognitive activities that occur in computer program development. These activities are required for programming, whether the programmer is a novice or an expert, since they constitute phases of the problem-solving process in general (Heller & Greeno, 1979; Newell & Simon, 1972; Polya, 1957). Such cognitive subtasks may be summarized as follows: 1) understanding the programming problem; 2) designing or planning a programming solution; 3) writing programming code that implements the plan; and 4) comprehension of the written program and program debugging (Pea & Kurland, 1983; Shneiderman & Mayer, 1979).

Studies in linguistics (Goodman, 1971; Lefever & Ehri, 1976) have indicated a significant relationship between subjects' cognitive style and ability to comprehend and identify multiple meanings in ambiguous sentences, a task that tests comprehension and diagnostic skills. Research in the psychology of
computer programming indicating that learning computer programming skills involves similar subtasks as does learning linguistic skills provides support for a study in computer programming to determine if differences in cognitive style are related to performance on subtasks of comprehension and debugging in computer programming.

Need for the Study

In addition to the many cognitive psychological studies conducted, a number of studies (Bariff & Lusk, 1977; Benbasat & Taylor, 1978; Dickson, Senn, & Chervany, 1977; Driver & Mock, 1975; Keen, Bronsema, & Morton, 1978; Mason & Mitroff; 1973; Sage, 1981) have been conducted concerning the effect of cognitive style on decision making behavior. These studies were motivated by the belief that the user's cognitive style should be considered in the design of Management Information Systems (MIS) and Decision Support Systems (DSS). According to Huber (1983) an examination of the literature leads to two rather discomforting conclusions: (1) The literature on cognitive style is an unsatisfactory basis for deriving operational guidelines for MIS and DSS designs, and (2) further cognitive style research is unlikely to lead to operational guidelines for MIS and DSS designs.
Huber admits that these conclusions are at odds with some widely held beliefs, and qualifies these conclusions by stating that "they do not state that all cognitive style research is without merit. It may be that cognitive style research has been or will be useful for career counseling, selection and placement, coaching and training, choosing procedures for designing DSS, or other related purposes" (1983, p. 567-568).

Huber indicates a need for further research in the areas mentioned above. Concerning selection and placement and the predictive validity of cognitive style assessment instruments, Huber states that "to date the proportion of empirical studies that have found cognitive style to be highly predictive is very small" (p. 568). Huber notes that (1) there are a variety of instruments available for assessing cognitive style, (2) the intercorrelations of scores for these instruments tend to be low, and (3) some of the instruments assess multiple constructs. Rather than considering these factors as a problem, Huber views them as an opportunity. These differences "indicate that cognitive style is a generic construct possessing a number of subconstructs (p. 572)," and "what is needed is the development of a contingency theory of cognitive style subconstructs, a theory or
at least a body of empirical research that relates subconstructs and instruments to decision settings and tasks (p. 572-573)".

A number of studies (Payne, 1976; Sage, 1981; Simon & Newell, 1971; Taggart & Robey, 1981; Vroom & Jago, 1974; Vroom, Jago, & Yetton, 1973) indicate that task considerations (such as type and complexity of the task) tend to dominate cognitive style considerations as predictors of decision making behavior. In fact, for those tasks in which the range of functional decision making approaches is quite narrow, this dominance is appropriate since it may force decision makers to use approaches that are not congruent with their natural cognitive styles (Benbasat, Taylor, & Dexter, 1979; Benbasat, Taylor, & Dexter, 1981; Dickson, Senn, & Chervany, 1977).

According to Huber (1983) the implications of the above research as it pertains to further research in the areas of coaching and training are as follows:

(1) There is a need for more and better research concerning the assessment of cognitive style, as suggested by Taylor and Benbasat (1980) and Zmud (1979), so that coaches, trainers, and decision makers will have an accurate assessment of "natural" propensities;

(2) There is a need for contingency-focused research, research relating particular cognitive style subconstructs to different decision settings and tasks, as suggested earlier and by Payne (1982), so that decision makers will have
available a taxonomy of decision situations that is useful for matching styles to situations; and
(3) There is a need for research that will show how decision makers can be trained to employ cognitive styles other than that to which they are naturally predisposed. It may be useful, before initiating such research, to examine two sets of studies: (1) studies where training was effective in enhancing cognitive complexity (Gardiner, 1972; Gardiner, 1974; Sieber & Lanzetta, 1966) and (2) studies examining cultural differences in cognitive style (Gruenfeld & MacEachron, 1975; Laosa & DeAvila, 1979; Ramirez & Herold, 1979), especially studies relating acculturation to changes in cognitive style (Buriel, 1975; Knight, 1978; Ramirez, Castaneda, & Herold, 1979; Witkin & Berry, 1975) (1983, p. 573-574).

To summarize the above discussion, the existing body of literature tends to indicate that individuals possessing global cognitive styles make decisions differently than those possessing analytic cognitive styles. Individuals must employ qualitative as well as quantitative skills during decision making, thus training is necessary to develop those skills that are not "natural". Since the decision task plays a major role in decision making, it is necessary to consider which subconstructs relate to which particular tasks during decision making. There is a need for this type of research as indicated by the inconsistent findings of previous studies.

Regardless of the terminology or the discipline, the theme of this discussion is that before conclusions can be drawn about the effects of
cognitive style on problem-solving (or decision making) further research must be conducted on the relationship between the subconstructs of cognitive style and/or the subtasks of problem-solving (or decision making). This study is an attempt at this type of research. Two more studies are reviewed below to provide additional support for this study which examines the relationship of cognitive style and a subtask of computer programming. Both studies relate cognitive style to general programming ability.

Testa (1973) used Witkin's measure of "perceptual style", the embedded figures task (EFT), to study programmer aptitude. The EFT is a series of problems requiring subjects to locate a specific shape in the context of a larger set of different shapes. The dependent measure is the time taken to locate the embedded figure. Higher scores provide an index of "field independent" cognitive style, while lower scores serve as an index of "field dependent" cognitive style. Testa found a significant correlation between field independence and success in a college COBOL programming course as indicated by test grades. The explanation for this finding was that "programming requires an ability to perceive the whole and a concomitant ability to proceed from the general to the particular" (p. 50), and "clearly, the EFT must have
tapped some of the characteristics of the programming task" (p. 52).

Cheney (1980) took an approach similar to Testa's in hypothesizing a significant relationship between scores on a programming examination for a college course in BASIC and cognitive style, as indexed by a questionnaire (Barkin, 1974) assumed to relate to the EFT's style categories of field independence/dependence. Cheney found scores on these two measures to be significantly correlated (r=.82), and suggested that those scoring highly on the style measure (field independent) may learn programming best by progressing at their own rate on programming projects, but that the field dependent thinker may require more structured and formal teaching to understand how to program. Pea & Kurland, (1983) indicate that these results signify a need for further research:

In each of these studies, the aim was to develop aptitude measures that would predict success in a programming course distinct from "mathematically oriented tests." Although they were successful in this limited goal, we do not yet know how cognitive style may be related to programming skill, and the studies do not bear on the more important question of how cognitive style may contribute to performance on specific programming subtasks. They do offer the suggestion, however, that cognitive style interacts with how one is taught programming, not with whether one can learn to program (p. 48).
CHAPTER II

COGNITIVE AND LEARNING STYLES

Cognitive style is one category in a broader area of psychology that is called "learning styles". The concept of learning styles traces its origins to two books published in the early 70's. These books introduced the concept of learning styles as a model of how certain environmental, emotional, sociological, and physical stimuli affected information processing in individuals (Dunn & Dunn, 1972, Kolb, 1971). The model established by the Dunns evolved in 1979 to include elements of cognitive style (Dunn, Dunn, & Price, 1979), and was revised again in 1980 to include hemispheric preference (Dunn, Cavanaugh, Eberle, & Zenhausern, 1982). Most recently the Dunns incorporated into the model the concepts of left and right cognitive styles as part of simultaneous and successive processing (Dunn, 1984).

Many other researchers (Canfield & Lafferty, 1970; Gregorc, 1979; Hunt, 1979; Kolb, 1971; Ramirez & Castaneda, 1974; and Schmeck, Ribich, & Ramanaiah, 1977) developed learning style constructs, all with varied definitions and models. Although many of the models established have considerable differences there is some agreement that they all have some essential
similarities (Dunn, DeBello, Brennan, & Murrain, 1981) as explained below.

According to Keefe (1979, p.9) "Learning styles are characteristic cognitive, affective, and physiological behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment". Affective styles are habits representing attention spans, emotional qualities, and value systems that influence the ways that individuals process information, develop within a given cultural environment, and are affected by family influence and personality differences. Physiological styles depend on human physical needs and how individuals react to their physical environment.

Cognitive styles are "information processing habits representing the learners' typical mode of perceiving, thinking, problem solving, and remembering" (Messick, 1969, p. 8). Cognitive styles are consistent stable habitual strategies that influence all human activities including social and interpersonal functioning (Messick, 1970). Cognitive styles are different from cognitive abilities. Cognitive styles represent the manners of cognition, while cognitive abilities are the capacities and the content of cognition. In other words, cognitive
styles are the "how" of learning, while cognitive abilities are the "what" of learning. In particular it should be noted that cognitive style emphasizes an approach rather than an ability. Many researchers (Coop & Brown, 1970; Garder et al., 1959; Kagan, Moss, & Sigel, 1963; Messick, 1969; Hill, 1971; Witkin, 1975) have experimented with cognitive style; even though their definitions differ all were essentially concerned with how information processing was affected by perception (Dunn, 1984).

Of the many different bipolar dimensions of cognitive style reported reflective/implusive (cognitive tempo) and tolerant/intolerant (tolerance for unrealistic experiences) are but two examples (Letteri, 1979). The definitions differed according to what each investigator regarded as individual style differences. For example, Tyler (1974) defined one dimension as the way of differentiating between parts or aspects of a confusing situation, while Doktar (1976) proposed that cognitive styles reflect two alternate modes of processing information; one defined as an analytic-sequential, linear, verbal-symbolic, and left-brain oriented mode, while the other was defined as being heuristic-intuitive, global, pictorial, and right-brain oriented.
Alternate Modes of Information Processing

The concept of alternate modes of information processing, especially the issue of how processes and information are represented in the brain, is one that has been discussed for quite some time by the scientific community (Gardner, 1987). Descartes recognized that different parts of the brain control different bodily functions. Juan Huarte, a contemporary of Descartes, rejected the doctrine of localization, suggesting that the brain instead functions as a unit (Diamond, 1974).

Understanding of the nervous system developed rapidly in the eighteenth century, when scientists such as Galvani, Bell and Magendie, and Muller showed that the sensory and motor functions of an organism were separate anatomical functions (Gardner, 1975; Herrnstein and Boring, 1965). In the early nineteenth century, Gall proposed a theory of localization (Robinson, 1976) that gained widespread public acceptance, but was criticized by other scientists. Gall's chief critic, Pierre Jean Marie Flourens, carried out experiments that demonstrated that the brain functioned in a holistic manner (Herrnstein and Boring, 1965).

Toward the end of the nineteenth century scientists (Fritsch and Hitzig, Ferrier) demonstrated
that localization of function was indeed a viable theory, by showing that stimulation of different areas of the cortex produced contraction of specific muscles in laboratory animals. Broca, working with aphasics who had lost their ability to speak as a result of brain injuries, postulated that the loss resulted from damage to the lower portion of the frontal lobe of the left hemisphere. Subsequent studies by Wernicke (1874) and Dejerine (1892) confirmed Broca's claim that injuries to the left cerebral hemisphere were related to difficulties in language understanding and production (Gardner, 1975).

Skeptics of localization persisted in their opposition. Pierre Marie re-examined Broca's studies and concluded that Broca's claims were not justified. Other neurologists began to endorse the holistic view of brain function (Gardner, 1987). In fact, these holistic views were reinforced by certain events from the field of Gestalt psychology. The concept of organisms reacting to the relationships among stimuli and overall patterns, and the assumption that the nervous system is organized in terms of neural fields operating across wide regions of the cortex, appeared to provide substantial support for the holistic theories of these neurologists (Gardner, 1987).
The intellectual differences between the holists and the "localizers" had reached a peak by the 1940s. Studies by Sperry, Weiss, Lashley, and Pribram in the early 1950s provided evidence indicating that certain claims made by both groups were untenable (Gardner, 1987). The results also had an affect on theories developed in psychology. Faced with the results of these experiments Gestaltist Wolfgang Kohler, the chief proponent of field theory, is said to have declared in desperation "that ruins not only my Direct Current field [theory] but every other neurological theory of perception" (Pribram, 1971, p. 110-111).

Neuropsychologist Donald Hebb presented a theory of behavior in 1949 that accommodated aspects of both localist and holist viewpoints. Hebb argued that early in development behavior patterns are built up of "cell assemblies". Behaviors defined by cell assemblies can be quite localized to specific regions or even cells of the brain. Subsequent development of "phase sequences", or more complex behaviors, are built up of sets of cell assemblies. Phase sequences are less localized and involve larger sets of cells from disparate sections of the nervous system (Gardner, 1987).

During a two decade period starting in the late 1950s, neurophysiologists Hubel and Wiesel
demonstrated that the function of the nervous system was highly specific. They accomplished this by showing that specific cells in the visual cortex of cats respond to specific forms of information in the environment. In addition they demonstrated that early experience played an important role in the development of the nervous system. Other workers (Lettvin, Mountcastle, Mishkin, Gross, Rocha-Miranda, and Bender) have complemented and strengthened the findings of Hubel and Wiesel (Gardner, 1987).

The work of Roger Sperry probably represents some of the most important, in terms of the knowledge it has contributed to the concept of localization of information processing in human beings. Working with epileptics in the late 1950s and 1960s, Sperry showed by experimental procedures the effects on behavior of those patients who had experienced a surgical procedure called cerebral commissurotomy, which essentially separates the two hemispheres of the brain by severing the connective tissue.

Sperry's experimental findings indicated that there was considerable lateralization of function in the human brain. The left hemisphere is dominant in language and other conceptual, classificatory or analytical functions, while the right hemisphere
assumes a dominant role for spatial functions and holistic or Gestalt-forms of perception. Another interesting finding was that each hemisphere of the same person could be tested separately, with one hemisphere being unaware of the testing going on in the other (Gardner, 1987).

By experimentation Sperry was able to show that it is possible for the right hemisphere to assume language function, even though language is typically dominated by the left hemisphere. Additionally Sperry showed that the younger patients were at the time of commissurotomy the more likely it was that they would possess well-developed capacities in both hemispheres (Gazzaniga, 1983). By contrast, older patients exhibited considerable lateralization of function as is common with most brain-damaged patients.

Other studies (Geschwind, 1974) have indicated that in young children the sensory regions are dominant, while in older individuals the association cortices and "planning regions" of the frontal lobes dominate. Thus injuries to these regions are more or less detrimental to individuals depending upon age.

Because of the plasticity of the brain tissue and its ability to adapt during development it would appear that an injury sustained early in life would have a less severe effect than one sustained later in
life. Unfortunately, early brain injuries causing relocation of certain functions within the hemispheres have detrimental effects on behaviors that typically develop later in life. In addition, underdevelopment of certain functions normally designated for a specific region or lack of optimalization of function may occur (Goldman-Rakic et al., 1982; Goldman and Galkin, 1978; Dennis and Whitaker, 1976).

Historical Development of Cognitive Styles

The development of the theory of cognitive styles had its origins in the laboratory studies conducted by Witkin and Asch in 1948 (Witkin & Goodenough, 1981). The trait extensively researched by Witkin and his associates was identified by many different terms during the course of the research. Examples of such terms are field independence/dependence, psychological differentiation, and analytic/global cognitive style. The research proceeded from an initial investigation of individual differences of perception of an upright object and was expanded over the years to include analytic/global cognitive style. Witkin developed his theories over a period of nearly forty years, starting in the late 1940s. Witkin's career included both teaching and research appointments at Brooklyn College.
(1940-1952), the State University of New York Downstate Medical Center (1952-1971), and the Educational Testing Service (1971 until his death in 1979). Some of the research was done by Witkin with his students and generated theses at the bachelors, masters, and doctoral levels, some of which were later published. His numerous publications periodically reviewed and summarized the work that he, his associates, and others had attempted, in terms of new terminology and developments of the theory. In general Witkin's latest publication (for a given period) was a cumulative review of the research conducted in cognitive style theory; thus his publications provide a historical development of cognitive style theory that lends perspective to the evolution.

Visual Perception

As is probably the case with many psychologists, Witkin began his studies with an interest in perception. Historically and developmentally there has been considerable interest by society and individuals in the many questions associated with perception. How does information concerning movement, colors, and the third dimension get processed? What role do memories, mental images, and imagination play in the way individuals perceive the world around them? Questions
like these have been pondered ever since philosophy began (Gardner, 1987). The Greeks were concerned with the relationship between vision and knowledge and how it contributed to understanding; in more recent times Descartes minimized the importance of sensory organs in perception, while others saw the sensory organs as the point of origin for all knowledge. Helmholtz believed that unconscious inferences about previously observed scenes are stored in the nervous system and supplement information provided by visual stimuli during perception of certain events. The Gestaltists asserted to the contrary that one perceives overall forms. With the advent of computers, artificial intelligence researchers, like Winograd and Marr, have developed models of perception by designing "seeing" machines and programs implemented and tested on computers. Marr proposed three levels of explanation of a model of perception in which he described the responses of neural cells, predicted experimental results in psychophysics, and wrote computer programs to interpret the results (Marr, 1982).

The Influence of Gestalt Psychology

The foundations for the research conducted by Witkin and his associates was established by the school of Gestalt psychology. The Gestalt movement was
officially established by Max Wertheimer, who published a paper on visual perception of movement in 1912. This movement is seen as the most direct link to the cognitive psychology of today, and represents the major efforts of cognitive psychology during a period (the first half of the twentieth century) dominated by behaviorism (Gardner, 1987).

Experiments by Wertheimer and his assistants showed that the standard psychological account of apparent motion at that time, that perception of movement was due to eye movements, was untenable. This was achieved by showing that movement is perceived in an interval too brief to permit eye movement, even if the subject maintains a rigid fixation on the moving object. The Gestalt psychologists examined many problems pertaining to the organization of perception, and favored a view of perceptual organization in which the way the parts of a visual field are seen is determined by the configuration of the entire field (Gardner, 1987).

The Gestalt psychologist probably most responsible for the origins of Witkin's work is Wolfgang Kohler, who undertook a landmark set of investigations with chimpanzees. Kohler's interest centered on the way that chimpanzees tackled problems whose solution required "restructuring" of the elements in a problem
situation in order to obtain a solution to the problem. Kohler (1925) found that the chimpanzees exhibited human-like thought processes, in which the apes stopped, reflected, and then, as if struck by a sudden flash of insight, reached for objects that provided solutions to the problems posed. According to Kohler's analysis, the chimpanzees "restructured" the visual field that had been presented to them, and did so courtesy of a moment of insight or an "A-ha experience", as it was termed by the Gestaltists.

Kohler also distinguished between bright and stupid apes. Bright apes actually experienced the moments of insight mentioned above; stupid apes, even after observing the correct behavior for obtaining a solution to a posed problem, could only imitate the component actions required for solution. They never appeared to appreciate how such actions were linked together.

The experimental findings of Wertheimer and Kohler greatly influenced Witkin's early studies. In these studies Witkin and his associates sought to determine how people so quickly and accurately locate an upright position in space. Three tests, the Rod and Frame Test (RFT), the Body Adjustment Test (BAT), and the Rotating Room Test (RRT), were employed to measure the
ability of people to locate the upright. The RFT is a perceptual test in which the subject, seated in a totally darkened room, must adjust to the upright a tilted luminous rod centered within a tilted luminous frame, while the frame remains in its initial position of tilt (Witkin, 1948; Witkin and Asch, 1948, cited in Witkin et al., 1971). The BAT is a perceptual test in which the subject is seated in a tilted position within a tilted room and must adjust his chair to a position where he perceives himself as upright, the room remaining tilted (Witkin, 1948, 1949, cited in Witkin et al., 1971). The RRT is a perceptual test in which the force on the subject's body is changed while the visual field remains upright. This is achieved by seating the subject in a chair, which can be tilted left or right within a small upright room driven around a circular track, so that the direction of the effective force on the body is the result of the outwardly-acting centrifugal pull and the downward pull of gravity. These studies revealed that subjects were markedly different from one another in performance on the orientation tasks, were self-consistent across tasks, and suggested that people have preferred ways of integrating the diverse sources of information available to them when seeking an upright position.
Experiments showed that individual differences did exist as measured by the BAT, RFT, and RRT. As a result, the two contrasting perceptual styles of field independence and field dependence were defined. Since scores from each of the three orientation tests formed a continuous distribution, the characteristic of field dependence-independence represented contrasting tendencies for subjects to rely on either body cues or cues from the surrounding field to find the upright, rather than distinct types of performance. Results of the RRT made the designations of field independence/field dependence value-neutral, since reliance on body cues would lead to more accurate performance in some situations but not in others (Witkin & Goodenough, 1981).

Further study led Witkin to believe that performance on the three orientation tasks might conceptually involve one's ability to separate a simple object from an organized field. Studies were conducted to test cognitive skills using instruments different than those used for the perceptual domain. Correlational studies were then conducted to show the relationship between functioning in the different domains. These studies conducted by Witkin and others have since been replicated, and the general consensus
is that the methods used were sound and the findings obtained are valid.

The Embedded Figures Test (EFT) (Witkin, 1950, cited in Witkin & Goodenough, 1981) was designed to measure this ability. The EFT is a perceptual test in which subjects must locate a previously seen simple figure which has been obscured or embedded in a larger complex figure (Witkin et al., 1971). Subjects who had difficulty separating simple figures from complex designs found in the EFT were consistently determined to be field dependent as measured by the BAT and RFT, while those who found it easy to locate the simple figures were consistently classified as field independent by both tests (Witkin & Goodenough, 1981). These findings indicated that field independence-dependence was more general than it was first defined, and paved the way for extension of the theory from the perceptual domain into the cognitive domain. Research on perceptual-analytical ability or 'disembedding', addressed two issues: first, the relation between disembedding and intellectual functioning, i.e., those human functions dealing with symbolic representations (Witkin et al., 1971); and secondly, the relation between embedding and cognitive restructuring ability.
Disembedding and Intellectual Functioning

Numerous studies addressing the two issues stated above (Fenchel 1958; Gardner, 1961; Gardner, Jackson, & Messick, 1960; Goodenough & Karp, 1961; Loeff, 1961; Pascual-Leone, 1969; Witkin, Dyk, Faterson, Goodenough, & Karp, 1962) have shown that disembedding in perception is linked to disembedding ability in intellectual functioning, i.e., those human functions that pertain to symbolic representations (Witkin et al., 1971). Intellectual functioning is problem solving that involves not only perceptual but also cognitive and metacognitive skills. Therefore it involves a more comprehensive set of skills than cognitive functioning does. As it concerns the first issue, field-dependent subjects were found to have greater difficulty than field-independent subjects in solving problems whose solution depends on taking an element critical to the solution out of the context in which it is presented and restructuring the problem material so that the element could be used in a different context (Glucksberg, 1956; Witkin et al., 1962). Research on the second issue employed problem-solving tasks that lacked a clear inherent structure. In these cases field independent subjects actively imposed structure on such tasks and experienced the problem as organized, while field-dependent subjects
could not impose such structure and experienced the problem "as is" (Witkin, et al., 1962).

Witkin's early work into the relation between perceptual disembedding ability and intellectual functioning was influenced by Wertheimer's (1945) contention that solutions of intellectual problems not directly involving perception often require the separation or "disembedding" of parts from context to define new relationships. Some early evidence (Woerner and Levine, 1950) seemed consistent with Wertheimer's expectations. Working with a group of 12-year-olds, Woerner and Levine found a significant relationship between scores on Witkin's perceptual battery and scores on the Wechsler Intelligence Scale for Children (WISC). This result raised the possibility that field independence might be associated with superior general intelligence. However this was not the case since further analysis indicated that perceptual measures were more highly associated with WISC performance scores than to WISC verbal scores, suggesting that such an interpretation was too simple.

Witkin's first studies (1954) provided an additional early source of interest in comparing intellectual and perceptual performance. Informal interviews with youngsters led to the hypothesis that
the ability to separate an item from its context expresses itself in an individual's intellectual as well as perceptual activities. Witkin's first step in verifying this hypothesis was to explore the relationship between field dependence and general intelligence as measured by performance on the 1937 Revised Stanford-Binet (Form L) by administering these tests to a group of forty-eight 10 year-olds (24 of each sex). A significant relationship in the expected direction was found between IQs and perceptual index scores for both boys ($r = .57$, $P < .01$) and girls ($r = .76$, $P < .01$), thus confirming Woerner and Levine's results with the WISC. Many other studies (Bell, 1955; Bound, 1957; Carden, 1958; Crutchfield, et al., 1958; Fenchel, 1958; Jackson, 1955 & 1957; Miller, 1953; Podell and Phillips, 1959; Rosenfeld, 1958; Seder, 1957) have since been reported on the relation between perceptual field dependence and performance on standard tests of intelligence and achievement.

An attempt to systematically explore the basis of this relation was not possible since the same test items in the Stanford Binet are not given to each subject. Further analysis of the WISC was possible since subtests are given to each subject in a consistent manner. A group of 25 boys and 25 girls of age 12 and a group of 30 boys of age 10 were tested,
with a significant relationship found between total IQ and perceptual index scores for the boys ($r = .55$ at 10, $r = .73$ at 12, $P < .01$ for both). For the girls the relation was in the expected direction but not significant ($r = .36$).

A comparison of relations between perceptual scores with WISC verbal and WISC performance scores gave results consistent with those reported by Woerner and Levine. The results appeared to lend support to the idea of a common component in perceptual and intellectual functioning; thus to determine more definitely which subtests of the WISC were most related to the various perceptual tests a factor analysis was conducted. For the analysis of both age groups WISC scores and scores on a standard battery of perceptual tests were available. In addition to these scores, data pertaining to perceptual and problem-solving situations was available for the individuals in the 10-year-old group. Upon analysis using methods described by Cattell (1952) three major factors, consistent with factors repeatedly found in other analyses of Weschler Scales (see Davis, 1956; Cohen 1957, 1959), emerged. These factors were (I) verbal comprehension, (II) attention-concentration, and (III) analytical field approach.
High loadings on three subtests of the WISC (Block Design, Picture Completion, and Object Assembly) for the same factor as the perceptual tests administered provided strong support for the hypothesis that a general cognitive style pervades perceptual and intellectual functioning. The high loading of Block Design on factor III (analytical field approach) provided further evidence of overlap among field dependence, spatial decontextualization, and flexibility of closure according to Witkin (Witkin, et al., 1962). Studies by Thurstone (1944) and Podell and Phillips (1959) provided additional support for Witkin's findings.

The factor III result prompted Witkin to expand the cognitive style terminology previously used. Since the result appeared to indicated subjects' ability to overcome embedding contexts, those who could overcome embedding contexts were said to experience the field in an analytical way, while those who could not were said to experience the field in a global way. In other words, this dimension of individual differences represents different ways of approaching a field, whether the field is present or symbolically represented. The term "field dependence" was now meant to connote the perceptual component of this more general cognitive style.
The results of Witkin's factor-analytic study provided an understanding of the relation between field dependence and intellectual functioning. As part of this understanding it should be mentioned that results reported between perceptual measures and full-scale intelligence test scores cannot be interpreted to mean that field-independent children are of generally superior intelligence (Witkin, et al., 1962).

The factor-analytic study by Karp (1962) provides considerable support for Witkin's conclusions, as do a number of other studies (noted below). Karp's study is of interest since it was conducted on a group of 150 male college students. Three factors which correspond very closely to those factors obtained in Witkin's study with children were highly loaded.

Guilford and his associates conducted a series of factor-analytic studies (1952, 1955a, 1955b, 1957) that demonstrated that the capacity to overcome an embedding context can be identified in both the perceptual and intellectual functioning of individuals. These studies provided further support for Witkin's work. Guilford and associates identified the "adaptive flexibility" factor similar to Witkin's factor III (analytical field approach), showing a high
loading for six different tests, including tests for insight, matching, and hidden figures. Each of these tests requires that a predominant organization or context be overcome. The intellectual tasks showed this characteristic quite clearly, having been chosen by Guilford and associates on the basis of specific hypotheses about the nature of adaptive flexibility.

Witkin and his associates, interested in the linkage between adaptive flexibility and the analytic field approach (factor III), administered a series of nine cognitive tests to 31 college men. These tests were the BAT, RFT, EFT, Block Design subtest of the Weschler Adult Intelligence Scale (WAIS), Picture Completion subtest of the WAIS, Guilford's Match Problems, Guilford's Insight problems, Vocabulary subtest of the WAIS, and the Comprehension subtest of the WAIS. The pattern of intercorrelations among measures of the nine tests offered strong support for the hypothesis that measures from tests representing mode of field approach would be interrelated but would tend not to be related to measures of the verbal factor. These findings were also supported by the factor-analytic study of Karp (1962), previously mentioned, which used the nine tests described above as part of a larger battery of tests. The results were consistent with those cited.
Among the tests which define adaptive flexibility in Guilford's Factor analysis, the Insight Problems Test attracted the most interest because of the close parallel between the problems on this test and tests of field dependence. A particularly clear illustration that insight problems require subjects to restructure problem material is found in the problems employed by Duncker (1945) in his studies of "functional fixedness". To solve these problems the subject needs to use familiar objects in new ways. A study by Harris provides substantial support for the hypothesis that the ability to solve problems of this nature is related to field dependence.

Studies with Einstellung problems (in which a subject must manipulate the contents of three jars of water in order to obtain a given amount in one of them) also provide evidence that ability to overcome contexts is expressed in both perceptual and intellectual functioning of individuals. Fenchel (1958) gave the Einstellung test and the EFT to a group of 63 outpatients in a VA Clinic and found a significant relation ($r = .36, P < .01$), in the expected direction. Guetzkow (1951), Guilford, et al. (1957), Jeffreys, (1953), and Zaks (1954) individually carried out studies in which forms of the Einstellung
problem were used to relate performance on such problems to mode of field approach. Although some of the findings were contradictory, Witkin suggested (Witkin et al., 1962) that the trend of the evidence suggested that performance on the critical Einstellung problem may not be related to mode of field approach, but performance on a version of this problem, called the extinction problem, is related to capacity for analytical functioning.

Additional evidence for Witkin's conclusions was later provided by a study conducted in Witkin's laboratory by Goodman (1960), where college students were administered Witkin's battery of perceptual tests, other perceptual tests, and the Einstellung test. The results confirmed and extended the findings of Fenchel and Guetzkow, indicating that set-breaking ability in the Einstellung situation tends to be related to mode of field approach as expected.

Since 1962, research has concentrated on establishing a relationship between disembedding and cognitive restructuring ability, including spatial, auditory, and verbal restructuring. It has been shown that a relation does exist between restructuring in the visual-perceptual and cognitive domains when verbal tasks were analogous to visual disembedding tasks (Witkin & Goodenough, 1981). Because there are
similarities between learning to program and learning to read as mentioned earlier, studies conducted on sentence disambiguating in linguistics appear to lend insight into learning to program. Sentence disambiguating is a process which requires subjects to consider several meanings in an ambiguous sentence and to establish a single semantic or grammatical interpretation (Witkin & Goodenough, 1981). Sentence ambiguity may occur at three levels: At the lexical level, when a word or phrase has different meanings; at a surface level, when alternate groupings of phrases are possible; and at an underlying structure level, when alternative logical relations exist at a deep level (Chomsky, 1965; Foss, Bever, and Silver, 1968; MacKay, 1966).

The Influence of Linguistics on Cognitive Psychology

The works of Noam Chomsky provide some perspective on the role of sentence disambiguating in language usage and cognition. In 1957 Chomsky published Syntactic Structures where he identified those characteristics of cognition that allow individuals to intuitively understand different levels of ambiguities in sentences used in everyday conversations. Chomsky identified mechanisms underlying human abilities to detect and unravel sentence ambiguities such as
"Flying planes can be dangerous", "The shooting of hunters disturbs me", and "I didn't shoot John because I like him" (Gardner, 1987).

Insights such as these were not unique to Chomsky; others such as Lewis Carroll and Edward Sapir (1921) had called attention to many of the same relationships that Chomsky investigated. The primary difference is that Chomsky went beyond his predecessors, even the insightful Sapir, by identifying the rules that individuals use to make sense of sentences that are both grammatically and ungrammatically correct (Gardner, 1987). The study of syntax conducted by Chomsky, and his success in developing methods to achieve the goals set in his studies, have illustrated that there are several domains of the mind that operate in terms of rules or principles that can be formally stated to establish cognitively scientific explanations of information processing (Gardner, 1987).

Experimental Results on Linguistic Ambiguity

Sentence disambiguating is a restructuring task analogous to disembedding tasks in the visual-perceptual domain. As ambiguities reach deeper linguistic levels, sentences become more difficult to disambiguate (MacKay & Bever, 1967), i.e., ambiguity
of greater complexity provides a greater challenge to restructuring skill. Researchers have supported the hypothesis that field independence is positively correlated with performance on sentence disambiguating tasks. Lefever and Ehri (1976) supported the hypothesis that field independence was found to be positively correlated \( (R= 0.44, p < .001) \) with the total number of sentences correctly disambiguated. Experiments conducted by Goodman (1971) showed that EFT performance was not significantly related to disambiguating ability for lexically ambiguous sentences since demand on subjects' restructuring skills was minimal. When subjects were presented with the more challenging restructuring task of disambiguating sentences possessing deeper levels of ambiguity, EFT scores related significantly to disambiguating ability as expected (Witkin & Goodenough, 1981).

The Analytic/Global Cognitive Style Construct

Evidence has been provided indicating that analytic functioning manifests itself throughout perceptual and intellectual activities; the nature of the analytic/global construct carries with it certain implications worth noting. One of the implications is that the set of characteristics associated with one
cognitive style is not inherently better than those characteristics associated with the other style. The dimension is value-neutral, each pole having qualities that are adaptive in particular circumstances. While analytic people possess cognitive restructuring skills, they also tend to be somewhat impersonal, i.e., they lack a well-developed set of skills allowing them to actively interact with others (Witkin & Goodenough, 1981). In contrast, while global people have lesser developed cognitive restructuring skills, they typically exhibit highly developed interpersonal skills. Obviously there exist situations in which interpersonal skills are more desirable traits to possess than cognitive restructuring skills, and vice versa.

Another implication of analytic/global cognitive style is the stability of the construct over a significant portion of one's life span, although studies have shown that training in the direction of one pole or the other is possible. Several studies (Dolecki, 1976; Hurwitz, Wolf, Bornick, and Kokas, 1975) have indicated that cognitive restructuring skills could be enhanced through appropriately designed educational programs. Research studies on training were conducted at different times during the development of the analytic/global construct; the main
objective of these studies was to increase the subjects' scores on the EFT toward the analytic pole. The early studies which were short in duration indicated that performance on the EFT improved by practice with the test materials. This improvement did not change the underlying perceptual functioning, i.e., the training effect did not increase cognitive restructuring ability. Therefore, another set of extensive training programs used instructional materials that were independent of the EFT content. A number of these training programs incorporated instructions in identifying and/or manipulating components of perceptual gestalten (gestalten refers to a physical, psychological or symbolic configuration having properties that cannot be derived from its parts). In general the results of such studies indicate that perceptual training in visual discrimination had a positive effect on EFT performance.

Although most of the training studies were aimed at improving restructuring and disembedding abilities, training in the area of interpersonal competencies has also been attempted. The number of studies is smaller since school curricula tend to emphasize elements of the analytical style. The ultimate objective of a
training program should be to equip individuals with abilities that are associated with both styles since "The person who has access to the characteristics of both modes ... has the potential for adapting to a wider array of circumstances" (Witkin & Goodenough, 1981, p. 63).

Developmental Factors

The demonstrations of Jean Piaget have had profound effects on the work in cognitive developmental psychology. Piaget sought to unravel the basic laws of thought by observing behavior in children of different ages involved in informal tasks. His principle contribution to psychology was the discovery of the basic structures of thought that characterize children as they develop and the mechanisms that enable children to make the transition to higher stages of development. These mechanisms include the sensorimotor stage of infancy, the intuitive stage of early childhood, the concrete operational stage of middle childhood, and the formal operational stage of adolescence. Although some of Piaget's claims and some of his logical formalisms have been either criticized or found to be invalid, he will be remembered for founding the field of human cognitive development and for a research agenda that
continues to provide problems for present day studies. In addition the research programs of Piaget kept the concerns of Gestalt psychology alive in the English-speaking world during the height of the behaviorist era (Gardner, 1987). Since Witkin's work was heavily influenced by the Gestaltists, it is apparent that the work of Piaget has also played an important role, if for no other reason than to maintain the credibility of Gestalt Psychology.

Research on developmental factors in cognitive styles has encompassed biological elements (Dawson, 1972, Waber, 1977, Bock and Kolakowski, 1973, Hartlage, 1970, and Stafford, 1961), child-rearing practices, and cultural variables. Two biological models were proposed in these studies: hormonal and genetic. The first assumes that hormonal levels influence the development of specialization of hemispheric functions at various ages during the growth years, which affects the development of cognitive restructuring skills (Witkin and Goodenough, 1981). The second assumes that a genetic determinant contributes to restructuring ability. Unfortunately, the findings from these studies are inconclusive and therefore do not provide evidence to verify the above hypotheses. On the other hand, evidence from studies
on child-rearing practices and cultural variables is clear and the results are more generalizable.

Child rearing practices

The early work of Witkin and his associates (1962) on socialization led to the hypothesis that child-rearing practices that encourage separate autonomous functioning foster development toward the analytic pole of cognitive style. Reliance on parental authority is more likely to foster development toward the global pole of cognitive style (Witkin, et al., 1962). Exploratory studies which were conducted to test this hypothesis primarily used interviews, questionnaires, and observational techniques for data gathering.

Interviews indicated that mothers of global children encouraged them to remain dependent upon her during development. Some examples of the methods employed by such mothers to encourage dependency are limiting the child's activities in the community, emphasizing conformity, discouraging assertive and aggressive behavior, particularly when it was directed towards the mother, and not stimulating children to assume responsibilities (Witkin and Goodenough, 1981).

In cross-cultural studies (Abelew, 1974; Baran, 1971; Bruner, 1977) interviews sought retrospective
information about childhood experiences from the subjects themselves. Findings indicated that parental strictness in socialization is associated with low disembedding ability in children.

Those studies (Claeys and DeBoeck, 1976; Edgerton, 1975; Seder, 1957) that used standardized questionnaires included questions about parents' actual behavior toward their children. A study by Seder (1957) included questions about autonomy and the extent to which it was encouraged by parents. Seder compared answers of parents of ten-year-old males and females who were identified as global or analytic according to EFT scores. It was found that global children were subject to child-rearing practices that stressed conformity and authority.

Observational studies have considered the power structure within families (Dreyer, 1975), parent-infant interactions (Dyk, 1969), the father's role within the family (Lee, 1974; Louden, 1973; Trent, 1974), and gender-role modeling (Biller, 1971; Lee, 1974). Although not entirely consistent, the results from this research provide substantial support for the hypothesis that a family environment that encourages independence is likely to produce children who at an early age develop in the direction of analytical functioning.
Cultural Variables

Cultural influences on cognitive styles must be considered since, historically, a debate has raged as to whether different peoples think differently than Western man or in a similar fashion but at an inferior level. The life studies of anthropologist Lucien Levy-Bruhl present a noteworthy example of the issues central to cognitively oriented anthropology (Gardner, 1987).

In the late nineteenth century when Levy-Bruhl began his study of the thinking processes of primitive people, it was assumed that the reasoning abilities of members of Western society was the standard by which all other peoples were measured. The thinking of other individuals around the world was considered as similar, but inferior to that of Western man. Bruhl challenged this opinion and proposed that primitive peoples reason not badly, but differently (Cazeneuve, 1972). Bruhl proposed two laws which defined the characteristics of thought for primitive peoples. First of all, the law of participation indicated that primitives think of themselves as simultaneously possessing qualities like and dislike objects, beings, and external phenomena. In addition Bruhl stated that primitives could emit and receive mystical forces
external to their own being, without ceasing to be what they are (Cazeneuve, 1972). Bruhl repeatedly tried to defend these new ideas which were highly criticized. Eventually Bruhl realized that his viewpoint was essentially impossible to defend and abandoned this line of thinking for the more traditional stance (Cazeneuve, 1972).

A decisive step in the advancement of cognitive anthropology was made by Claude Levi-Strauss. His contribution was to demonstrate that the key aspects of cultural differences among people are best thought of as being linguistic and not cognitive by nature, and are best approached and studied by the methods used by the structural linguist (Gardner, 1987). Psychological studies (Dasen, 1972; Dasen & Heron, 1981) were conducted which were sympathetic to the finding that non-westerners (especially those who lacked formal schooling) performed far more poorly on tests of abstraction, conceptualization, and classification than westerners. These studies set out to dispel the notion that differences found could be attributed to the inferior thinking exhibited by non-westerners. Methodological adjustments (using familiar materials and explanation of the requested behavior, for example) made it possible to show that the cultural differences found in earlier studies were
definitely not the result of differences in thinking (Cole and Scribner, 1974). As a result much greater stress is now placed on using materials that are "culturally-fair" and testing capacities that are basic to individuals (Gardner, 1987).

A large body of cross-cultural research has undertaken the hypothesis that members of societies in which there is a great encouragement toward autonomy would be relatively better at disembedding ability and vice versa. These studies have examined groups that differ in extent of social conformity in terms of disembedding ability (Witkin, et al., 1974; Kagan and Zahn, 1975). In these studies, there is substantial support for the above hypothesis.
CHAPTER III

REVIEW OF THE LITERATURE

By its nature computer science is a rapidly evolving discipline. Ralston and Shaw (1980) mention some important learning implications of this fact, applicable to those considering the computer science major in college:

Specific skills learned today will really become obsolete. The principles that underlie these skills, however, will continue to be relevant. Only by giving the student a firm grounding in these principles can he or she be protected from galloping obsolescence. Even a student who aspires only to be a programmer needs more than just programming skills. He or she needs to understand issues of design, of the capability and potential of software, hardware, and theory, and of algorithms and information organization in general (p. 67).

The point of this argument is that general principles, not specifically tied to any single "best" programming language or programming machine should be an educational goal for establishing programming literacy. As mentioned earlier cognitive theories have been developed that articulate a set of distinctive cognitive activities occurring during the development of a computer program. These activities represent a set of fundamental programming concepts necessary for establishing a basic programming literacy and may be summarized as follows: 1) problem understanding - the understanding of data classes, types of data
processing operations, procedurality, the logic of sequential instructions and program function; 2) designing and planning the program - designing definitive procedures of various types, symbolic representations of plans, and the detection and correction of errors in plans, 3) program coding - translating plans into program code using syntactic rules and programming basic language structures and logic forms, and 4) program comprehension and debugging - understanding and being able to correct errors in program functions, procedures, and program errors related to the original problem statement.

To learn these concepts what demands will be made on learners in a cognitive sense, i.e., what aptitudes or cognitive prerequisites are necessary for learning computer programming? According to Pea and Kurland (1983) no such research has been directly aimed at defining the cognitive prerequisites for learning programming; instead most research has been concerned with predicting which individuals will be most successful at programming, rather than who can do it at all. Still the following five factors have been mentioned frequently in the literature: (1) mathematical ability, (2) memory capacity, (3) analogical reasoning skills, (4) conditional reasoning skills, and (5) procedural thinking skills.
The literature on the cognitive prerequisites for learning computer programming has concentrated on either the ability of aptitude test scores to predict course performance or the importance of specific abilities or traits for learning programming (Nowacyzk, 1984; Pea & Kurland, 1983; Webb, 1985). Many studies (Bauer, Mehrens, & Vionhaler, 1968; DeNelsky & McKee, 1974; Hollenbeck & McNamara, 1965; McNamara & Hughes, 1961) have correlated scores on programming aptitude tests, including the IBM Programmers Aptitude Test (PAT) and Aptitude Assessment Battery-Programming (AABP) with performance during training courses or on-the-job. Moderate correlations in the range of .30 to .70 have been found. Since programming aptitude tests typically assess a combination of abilities, it is difficult to determine the relative importance of such abilities for predicting performance (Pea & Kurland, 1983; Webb, 1985). The PAT, for example, assesses verbal, figural, and arithmetic reasoning (letter series, figure series, number series) and the AABP requires manipulation of precisely defined symbols, logical reasoning, strict adherence to instructions, and the use of flow charts (Webb, 1985).

Only a few studies (Cheney, 1980; Mayer, 1975, 1976; McNamara, 1967; Ricardo, 1983; Snow, 1980) have
investigated the importance of specific abilities or cognitive style variables for learning computer programming. The specific variables examined in these studies include inductive and deductive reasoning, mathematics, and field independence. In addition, some studies (Mazlack, 1980; Ricardo, 1983) have investigated the importance of demographic characteristics as predictors of programming achievement. Most studies of reasoning ability have shown positive correlations with programming performance. For example, Snow (1980) found that fluid analytic reasoning and visualization (defined by tests of nonverbal reasoning and paper folding) correlated with achievement in BASIC. In a study of an introductory college level programming course in PL-1, Ricardo (1983) reported that both inductive and deductive reasoning related to achievement. For inductive reasoning, discovering a rule from what is given, Ricardo used figural relationships and number series. Her measure of deductive reasoning, inferring a particular conclusion from what is given, was verbal syllogisms.

Although a popular notion is that highly developed mathematical ability is necessary for learning computer programming, several studies have shown that
success in mathematics may not be a strong predictor of programming performance (McNamara, 1967; Nowaczyk, 1984; Tillman, 1979). In fact, Pea and Kurland (1983, p. 36) concluded that "To our knowledge, there is no evidence that any relationship exists between general mathematics ability and computer programming skill once general ability has been factored out." However, Mayer (1975, 1976) found moderate correlations between computational skill and ability to learn BASIC. Ricardo (1983) reported a significant relationship between mathematics scores on the Scholastic Aptitude Test (SAT) and achievement in a PL-1 programming course. Wileman, Konvalina, & Stephans (1981) cited mathematical ability as a factor used to predict success in programming. In their study eight factors, including reading comprehension, sequence completion, logical reasoning, and algorithmic execution were shown to be related to performance on a final exam including topics in machine language, assembly language, and PL-1 programming. A stepwise multiple regression procedure was performed to determine the most important factors in predicting success in a beginning computer science course. When all eight factors were included approximately 25% of the variability in the final exam scores was explained. Studies such as these indicate that the evidence on
Mathematics as a predictor for learning programming is too inconsistent to draw any general conclusions (Webb, 1985).

Mathematical ability is only one characteristic that may be correlated with an individual's ability to program (Howell, Vincent, & Gay, 1967; Wileman, Konvalina, & Stephans, 1981). Cognitive style is another factor that has been shown to predict a person's programming ability, and this variable is not biased in favor of applicants with an educational advantage in mathematics (Barkin, 1974; Cheney, 1980; Witkin & Goodenough, 1981).

The studies (Cheney, 1980; Snow, 1980; Testa, 1973; Webb, 1985) examining the relationship between cognitive style and programming have concentrated on field independence. The interest in field independence stems from findings in non-computer related studies that field independent students perform better on analytic tasks than do field-dependent students (Adejumo, 1983; Gaines, 1975; Goldman & Hudson, 1973; Katz & Postal, 1964; Lefever & Ehri, 1976; MacKay & Bever, 1967).

Because computer programming involves restructuring tasks, researchers have expected and found a positive relationship between field
independence and programming performance. Testa (1973) found a significant correlation between scores on Witkin's embedded figures task (EFT) and overall achievement in a college COBOL programming course. Using a questionnaire developed by Barkin (1974) that was assumed to relate to performance on the EFT, Cheney (1980) found a significant correlation between field independence and overall achievement in a college course in BASIC. A similar finding appeared with the Hidden Figures Test (HFT; French, Esktrom, & Price, 1963) among junior high school students learning LOGO (Webb, 1985). Snow (1980) reported a positive relationship between a personality variable reflecting self-reported flexibility and independence in academic work and performance in a short college course in BASIC.

The study conducted by Cheney involved 35 students in undergraduate classes at a large state university learning BASIC. Student cognitive style was tested using an instrument developed by Barkin (1974). A two-part programming examination was administered. The first part required subjects to determine the output for different programming segments, and the second part was a programming composition task. The programming test scores for each group were correlated with the cognitive style questionnaire scores. A
Spearman rank-correlation coefficient of .82 was produced for the combined programming score (the Spearman test was used since distributions could not be assumed to be normal). The coefficient for part one was .87 while the coefficient for part two was .75. A computed t value of 8.29 indicated that the null hypothesis \( H_0 = \) no relationship between cognitive style scores and programming scores) should be rejected at .01 level of significance.

Based upon these results Cheney concluded that cognitive style could serve as a predictor of student success in learning a computer programming language. Cheney also concluded that the results of this study had implications for the methods used to teach individuals possessing different cognitive styles how to program. Cheney concluded that individuals with an analytic cognitive style may require minimal teacher guidance and intervention, while those with a global cognitive style may need a more structured learning environment and considerable teacher assistance.

The few studies relating demographic characteristics to programming performance are inconsistent. Ricardo (1983) and Mazlack (1980) found insignificant correlations between programming and year in school and gender. The findings for gender
were replicated in a recent study of achievement in LOGO (Webb, 1984) but that study found a significant positive correlation for age.

Most of the above studies relating characteristics of the student to learning of computer programming have used total scores on achievement tests or grades in programming courses. Some researchers (Collins & White, 1984; Pea & Kurland, 1983; Shneiderman & Mayer, 1979; Webb, 1984, 1985;), however, maintain that it may not make sense to view achievement in computer programming as a unitary variable. Pea & Kurland (1983), for example, classified programming concepts into four types: understanding the problem, designing and planning the program, coding the program, and comprehending and debugging the program. Shneiderman (1976) identified four relevant cognitive skills involved in the progression through the learning stages of programming: comprehension, composition, debugging, and modification. Mayer (1975) further distinguished between interpreting programs already written and generating new programs, as well as distinguishing between different levels of knowledge. Showing different prerequisite profiles of abilities for different programming outcomes would dispel the notion that a computer programmer must have a certain
configuration of abilities (Pea & Kurland, 1983; Webb, 1984, 1985) to be successful.

In one of the studies examining this question, Webb (1984) found that for achievement in LOGO different abilities predicted different outcomes. Mathematical ability was the best predictor of knowledge of syntax, interpreting graphics programs, and generating non-graphics programs. Spatial ability predicted knowledge of commands. A combination of spatial ability and field independence best predicted generating graphics programs. In a related study Webb (1985) explored whether different abilities predicted different outcomes among students learning to BASIC programming in individual and small group settings. Programming outcomes measured included knowledge of commands and syntax and ability to interpret and generate programs. No difference on any programming outcome appeared between individual and group settings. The cognitive prerequisites of learning computer programming, however, differed between learning settings and across computer programming outcomes. Quantitative and verbal ability were the best predictors of programming outcomes in the individual setting. Nonverbal reasoning, spatial ability, and age were additional potent predictors of learning in the group setting.
A study conducted by Collins & White (1984) indicates that the cognitive skills involved in diagnostic programming tasks (debugging tasks) seem to be related to the ability to compose programs, while performance on comprehension tasks does not insure that debugging ability is well developed in the same set of subjects. Forty subjects enrolled in an intermediate programming course were administered three programming tasks to test competence in program comprehension, program composition, and program debugging. Based upon the performance on the composition task three distinct groups emerged which were used for further study. An analysis of variance was performed to describe the relations between the tasks for the three groups. An F-value of 3.03 on programming comprehension was not significant. On the debugging task, the results were quite different. Although an F-value of 3.73 was not significant overall, a significant difference at the .05 level in the ability of two of the groups to find and correct errors was identified.

From these results it was concluded that the component skills of comprehension, composition, and debugging are required in programming and all need to be specifically addressed during instruction. Collins
and White concluded that the necessary cognitive skills required in program comprehension are skills that students learn quite early in their formal education. Thus students found programming comprehension to be a relatively easy task. Collins and White also found that comprehension skills were not related to students' ability to compose programs, while the cognitive skills involved in debugging tasks seemed to be related to ability to compose programs. It was concluded that the cognitive skills required for program comprehension are common to the current educational experiences of students, while the cognitive skills for diagnostic skill are not.

A continued search, using ERIC and the Social Sciences Citation Index, failed to reveal any information pertaining to studies conducted to show a relationship between measures of cognitive style and diagnostic skills (ability to locate and correct logic errors in computer programs) for novice programmers. Therefore, it will be assumed that studies such as the present one have not been previously attempted.
CHAPTER IV

METHOD

Operational Definitions

For the purposes of this study the terms listed below will be defined as follows:

**Syntactic error diagnostic task** - A set of two performance tests designed to measure student programmers' ability to locate and correct syntactic errors in computer programs. These tasks are computer programs that have been compiled and tested for correctness until they met problem specifications. The correct programs had syntactic errors entered into them, were validated, and administered to subjects in a controlled setting during the course of this study. Performance on these tests was determined by tabulating 1) the number of errors that were located and corrected by subjects, 2) the number of errors that were located but not corrected by subjects, and 3) the number of program statements subjects marked as being syntactically incorrect when the syntax of the statement was actually correct. The tests were administered separately and on different occasions.

**Logic error diagnostic task** - A set of five performance tests designed to determine student programmers' ability to locate and correct logic errors.
errors in computer programs. These tasks are computer programs obtained from course teaching materials that have been compiled and tested for correctness until they met problem specifications. The correct programs had logic errors entered into them, were validated, and administered to subjects in a controlled setting during the course of this study. Performance on the tests was determined by tabulating 1) the number of errors that were accurately located and corrected by subjects, 2) the number of errors that were accurately located, but not corrected by subjects, and 3) the number of program statements subjects marked as being logically incorrect when the logic of the statement was actually correct in the context in which it was used. These tests were administered on three separate occasions. The first test was administered separately while the second and third tests and the fourth and fifth tests were administered jointly on separate occasions. The second two testing periods were twice as long the first.

Scoring schemes - five methods used to compute composite scores for diagnostic tasks to determine if scores attained by weighting the measures of performance in different ways had an effect on the relationship between GEFT scores and diagnostic task scores. After completion of each task subjects
received their results which were scored using the performance criteria outlined in the first two operational definitions. Subjects were unaware that different scoring schemes were employed to weight the performance measures. The scoring schemes are defined below.

**Scoring Scheme One**
1) One point was assigned to the test score when the subject located and corrected an error.
2) One-half point was assigned to the test score when the subject located but failed to correct an error.
3) Negative one-half point was assigned to the test score when the subject modified a statement not in error.

**Scoring Scheme Two**
1) One point was assigned to the test score when the subject located and corrected an error.
2) One-half point was assigned to the test score when the subject located but failed to correct an error.
3) Negative one point was assigned to the test score when the subject modified a statement not in error.

**Scoring Scheme Three**
1) One point was assigned to the test score when the subject located and corrected an error.
2) Zero points was assigned to the test score when the subject located but failed to correct an error.
3) Negative one-half point was assigned to the test score when the subject modified a statement not in error.

**Scoring Scheme Four**
1) One point was assigned to the test score when the subject located and corrected an error.
2) Three-fourths of a point was assigned to the test score when the subject located but failed to correct an error.
3) Negative one-half point was assigned to the test score when the subject modified a statement not in error.

**Scoring Scheme Five**
1) One point was assigned to the test score when the subject located and corrected an error.
2) One point was assigned to the test score when the subject located but failed to correct an error.
3) Negative one point was assigned to the test score when the subject modified a statement not in error.

**Justification of Scoring Schemes**

The literature on cognitive style provides a theory for justification of the above scoring scheme definitions. Subjects possessing an analytic cognitive
style attempt to reduce problem situations to a core set of underlying causal relationships. Analytic problem solvers attempt to choose optimal alternatives. On the other hand, subjects possessing a global cognitive style emphasize common sense and intuition; such subjects are characterized by a trial-and-error approach and the use of feedback to adjust the course of actions chosen (Cheney, 1980). For the case of computer programming, analytic problem solvers find that error correction is almost automatic upon location of the error, since they employ a structured approach to decision making. Global problem solvers need to first make program modifications, test these modifications, and receive feedback to verify their decisions since they reason by analogy.

To differentiate between these two types of problem solvers, a scoring scheme for diagnostic tasks should possess the following two characteristics: (1) it should assign the same or nearly the same weight to the location and correction of errors, as it does to the location of errors only and (2) it should assess a penalty, in terms of points subtracted from the overall score, to those subjects who make use of trial-and-error methods. With the exception of scoring scheme three, the schemes defined above follow these criteria. Scoring scheme three violates characteristic
one above since it does not give credit to individuals who locate but do not correct an error. Still, this scheme will provide a valid test of the theory since, if supported, it will provide some contradictory evidence to the theory and if not it will lend support to the theory. Variations in the weights assigned for the different schemes should illustrate which scoring scheme or schemes are optimal, i.e., which provide results most consistent to the cognitive style theory stated above.

Hypotheses

Based upon the theoretical and empirical considerations outlined in Chapters I and II and the operational definitions given above, the following hypotheses were defined and tested experimentally in the present study:

Null Hypothesis I: There is no linear correlation between Group Embedded Figures Test scores and scores attained by subjects on a set of two diagnostic computer programming tasks containing only syntactic errors as measured by (1) the number of syntactic errors subjects locate and correct, (2) the number of syntactic errors subjects locate and fail to correct, and (3) the number of program statements subjects mark
as being incorrect when the syntax of the statement is actually correct.

**Null Hypothesis II:** There is no linear correlation between Group Embedded Figures Test scores and scores attained by subjects on a set of five diagnostic computer programming tasks containing only logic errors as measured by (1) the number of logic errors subjects locate and correct, (2) the number of logic errors subjects locate and fail to correct, and (3) the number of program statements subjects mark as being incorrect when the logic of the statement is actually correct in the context in which it is used.

The scoring schemes outlined in the Operational Definitions will be used to weight the measures mentioned in the above hypotheses to determine the effect of a scaling mechanism on correlational values.

**Design**

Thirty-nine students enrolled in two sections of an introductory COBOL programming course (IS 360) at Boise State University during the Spring Semester, 1987 participated as subjects for the study. IS 360 is a required course for Information Science majors at Boise State University. Both sections were taught by the same instructor, using the same teaching techniques, materials, and methods of evaluation for
the entire semester. Instruction in diagnostic
techniques and program composition was given during
the semester by the instructor.

Subjects were administered the GEFT to determine
their cognitive styles. The data collected served as a
measure of the independent variable of this study.
During the semester subjects completed seven
diagnostic programming tasks administered on different
occasions (See Appendices A-C). The data collected
served as measures of the dependent variables of this
study. The measures of these variables were
correlated to determine their relationship.

The following demographic data were collected on
each student: Name, age, gender, names and number of
semesters of previous high school and college level
programming courses completed, and previous work-
related programming experience. These data were used
to determine which students, if any, did not meet the
novice student programmer criteria as defined for this
study. In addition subjects were asked to list the
names and number of semesters of previous high school
and college level mathematics courses they had
completed prior to participating in this study. At
the conclusion of the study subjects were asked to
briefly describe the process they used to locate and
correct syntactic errors and logic errors in computer
programs. This information was used in conjunction with a supplemental analysis conducted as part of this study.

Instrumentation

The Group Embedded Figures Test (GEFT) was the instrument chosen to measure the cognitive styles of subjects. The GEFT is a speed test consisting of 18 items. Norms for the GEFT, obtained from a validation study in which 155 college age men and 242 college age women participated, indicate that the mean score for the test is 12.0 (S.D. = 4.1) for men and 10.8 (S.D. = 4.2) for women (Witkin et al., 1971). For men GEFT scores were distributed as follows: 0-9 first quartile, 10-12, second quartile; 13-15, third quartile; 16-18, fourth quartile. For women the scores were distributed as follows: 0-8, first quartile; 9-11, second quartile; 12-14, third quartile; 15-18, fourth quartile. These data indicate the degree of field independence (dependence) an individual possesses.

An estimate for the reliability of the GEFT was obtained by correlating parallel forms which have identical time limits. Correlations between the 9-item First Section scores and the 9-item Second Section scores were computed and corrected by the
Spearman-Brown prophecy formula, producing a reliability estimate of .82 for both males (N=80) and females (N=97). These reliability estimates compare favorably to those of the EFT (Witkin et al., 1971).

Validity for the GEFT was obtained in studies comparing results of the GEFT to results on the EFT and the PRFT, a portable version of the RFT. Correlations between GEFT and the RFT (-.82 for men and -.62 for women; values are negative since better test performances were assigned lower scores) are reasonably high particularly for men. Correlations between GEFT and PRFT (-.39 for men and -.34 for women; values are negative since the better test performances were assigned lower scores) fall toward the lower end of the range of correlations typically found between EFT and RFT. The correlations between GEFT and ABC (.71 for men and .55 for women) are substantial, particularly for male subjects, and are generally comparable with those that have been reported for the EFT. According to the authors of the validation studies, the combined evidence suggests that the GEFT is a useful substitute for the EFT when individual testing is impractical (Witkin et al., 1971).
A study conducted by Pierson and Horn (1984) provided guidelines for constructing the syntactic error diagnostic tasks. In their study Pierson and Horn examined the source listings of unsuccessful compilations of COBOL programs by college-level student programmers. The listings were analyzed to determine (1) the locations of errors within the programs, (2) the causes of syntactical errors, and (3) the most common syntactical errors. Their findings provided a basis for modifying previously compiled COBOL programs for constructing the syntactic error diagnostic tasks. The source code for these programs was obtained from teaching materials that accompany the IS 360 COBOL textbook (Welburn, 1986).

The syntactic error diagnostic tasks are performance tests, i.e., instruments intended to evaluate an aspect of subject performance on a given task that is common to a particular process. By design, such a test measures that performance programmers need to exhibit to verify diagnostic skills. Thus, the content of these instruments must be validated.

To obtain content validity, these instruments were subjected to the scrutiny of a panel of established experts. This panel consisted of the Department Chair of the Computer Systems Department and a professor
from the Department of Mathematics at Boise State University. The chairperson has been a college-level instructor and COBOL programmer for the past 23 years and has written COBOL code in all versions and on most major systems that have existed during that period. The mathematics professor has nine years of teaching experience in computer programming at both the undergraduate and graduate college levels. His teaching and programming experience include work in such procedural languages as Pascal and C. He has considerable knowledge in assembly language programming and computer architecture.

The panel members reviewed the syntactic error diagnostic tasks to determine if the included errors were (1) located in the proper divisions of the program, (2) if the errors were the kind that COBOL programmers encountered most often in practice, and (3) if the level of difficulty of each task was appropriate. The syntactic error diagnostic tasks were validated with minor modifications in both cases.

Fuori, Gaughran, Gioia, & Fuori (1986) identified the most common logic errors that occur in COBOL programs in their introductory college level textbook. The source code for five programs, obtained from teaching materials which accompany the IS 360 COBOL
textbook (Welburn, 1986), was entered into computer memory, compiled, and tested. When these programs met initial program specifications certain program statements were changed using the aforementioned guidelines (Fuori, Gaughran, Gioia, & Fuori, 1986) to enter logic errors in each program. These "bugged" programs became the set of logic error diagnostic tasks that subjects performed for the purposes of evaluating their diagnostic skills.

The logic error diagnostic tasks are performance tests, as are the syntactic error diagnostic tasks. To validate these tasks the panel members reviewed them to determine if task errors were (1) located in the proper divisions of the program, (2) the kind that COBOL programmers encounter most often in practice, and (3) of the appropriate level of difficulty. Each of the five logic error diagnostic tasks was validated after minor modifications were made.

Experimental Procedure and Data Collection

In the Spring Semester 1987, thirty-nine Boise State University students enrolled in IS 360, Introduction to COBOL Programming, were administered the GEFT by their instructor. This researcher scored the GEFT and secured the scores until the completion
of the study when they were made available to subjects upon request.

Starting in the fourth week of the semester, and following at two week intervals, the course instructor administered the seven diagnostic tasks. One task per testing period was administered for the first three tasks, while two tasks were administered during each of the final two testing periods. All together there were five testing periods, the last two testing periods being twice as long as the first three. Six programming problems were assigned, one every two weeks, beginning in the second week of the semester. The content of the diagnostic tasks corresponded to that of the programming problems assigned. The following data were collected for each of the seven tasks: (1) The number of errors that were located and corrected, (2) the number of errors that were located, but not corrected, (3) the number of program statements subjects marked as being in error when the statement was actually correct.
CHAPTER V

RESULTS

Analysis of the data collected in this study was conducted using the Statistical Analysis System (SAS), a computerized statistical analysis package. The XEDIT text editor was used for creating the data files needed for analysis. Both systems are implemented on the IBM 4341 mainframe computer at Boise State University. Data collected during the study were weighted and summarized. Subjects were classified into two groups based upon previous programming experience. Groups were analyzed to determine if their performance differed significantly on either the GEFT or the diagnostic tasks. Correlation coefficients for GEFT scores and composite scores were computed, using five different weighting methods. Correlation analysis of these data failed to yield results significant at the .05 level. A detailed discussion of the statistical analysis follows.

Preparation of Data

Thirty of the original thirty-nine subjects completed the GEFT, the demographic surveys, and the seven tests (two syntactic error tasks and five logic error tasks) administered to test diagnostic skills. The following data were collected from each test: (1)
The number of errors that were located and corrected, (2) the number of errors that were located, but not corrected, and (3) the number of statements marked as being in error that were actually correct.

A total score for each of these three categories was calculated producing three separate totals for syntactic measures and three separate totals for logic measures for each subject (see Appendix D). The three totals for syntactic measures were then weighted using scoring schemes one through five (see Operational Definitions) and summed to produce five separate composite scores for each subject. A similar procedure was used to produce five separate composite scores of logic measures for each subject. For the syntactic error diagnostic tasks the maximum composite score was 30, while the maximum composite score for the logic error diagnostic tasks was 9. Actual composite scores attained by subjects, under scoring schemes one through five ranged from 15.5 to 27.75 on the syntactic error tasks, and 0.5 to 9.0 on the logic error tasks. These two sets of five values were then used to compare subjects' performance on diagnostic tasks with their GEFT scores.
Classification of Subjects

Examination of subject surveys regarding past programming education and programming experience indicated that 12 of the 30 subjects exceeded the novice programmer criteria. Since a reduction in the sample size of this magnitude was thought to be undesirable, the mean performance on the GEFT and diagnostic tasks for the two groups was analyzed to determine if the means differed significantly at the .05 level.

To determine if mean performance of the two groups differed significantly, Student t tests were performed using SAS. A preliminary test indicated that the variance of the two groups could be assumed equal (level of significance = .05). The results of this analysis indicated that no significant difference at the .05 level existed between the two groups on mean composite scores for either diagnostic task (regardless of scoring scheme used to compute composite scores) or means of GEFT scores (see Table 1).

The values obtained for the means of the syntactic error diagnostic task are nearly identical for both groups in all cases (see Table 1). Findings from previous research (Vessey, 1986; Weiser, 1985; Wertz, 1982) provide a qualitative explanation of these
TABLE 1: A COMPARISON OF MEAN PERFORMANCE ON DIAGNOSTIC TASKS AND GEFT SCORES FOR THOSE WITH EXPERIENCE AND THOSE WITHOUT EXPERIENCE

<table>
<thead>
<tr>
<th>Task and Scoring Scheme</th>
<th>Experienced Subjects</th>
<th>Inexperienced Subjects</th>
<th>T value</th>
</tr>
</thead>
<tbody>
<tr>
<td>syntax scheme 1</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.15, p &gt; t = .89</td>
</tr>
<tr>
<td></td>
<td>x = 23.83</td>
<td>x = 23.69</td>
<td></td>
</tr>
<tr>
<td>syntax scheme 2</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = .16, p &gt; t = .87</td>
</tr>
<tr>
<td></td>
<td>x = 22.50</td>
<td>x = 22.69</td>
<td></td>
</tr>
<tr>
<td>syntax scheme 3</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = .13, p &gt; t = .90</td>
</tr>
<tr>
<td></td>
<td>x = 23.08</td>
<td>x = 23.22</td>
<td></td>
</tr>
<tr>
<td>syntax scheme 4</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.30, p &gt; t = .77</td>
</tr>
<tr>
<td></td>
<td>x = 24.21</td>
<td>x = 23.93</td>
<td></td>
</tr>
<tr>
<td>syntax scheme 5</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.07, p &gt; t = .94</td>
</tr>
<tr>
<td></td>
<td>x = 23.25</td>
<td>x = 23.17</td>
<td></td>
</tr>
<tr>
<td>logic scheme 1</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.74, p &gt; t = .47</td>
</tr>
<tr>
<td></td>
<td>x = 5.25</td>
<td>x = 4.81</td>
<td></td>
</tr>
<tr>
<td>logic scheme 2</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.94, p &gt; t = .36</td>
</tr>
<tr>
<td></td>
<td>x = 5.08</td>
<td>x = 4.42</td>
<td></td>
</tr>
<tr>
<td>logic scheme 3</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.54, p &gt; t = .60</td>
</tr>
<tr>
<td></td>
<td>x = 4.67</td>
<td>x = 4.28</td>
<td></td>
</tr>
<tr>
<td>logic scheme 4</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.83, p &gt; t = .41</td>
</tr>
<tr>
<td></td>
<td>x = 5.54</td>
<td>x = 5.07</td>
<td></td>
</tr>
<tr>
<td>logic scheme 5</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -1.12, p &gt; t = .28</td>
</tr>
<tr>
<td></td>
<td>x = 5.67</td>
<td>x = 4.94</td>
<td></td>
</tr>
<tr>
<td>GEFT</td>
<td>n = 12</td>
<td>n = 18</td>
<td>t = -.11, p &gt; t = .91</td>
</tr>
<tr>
<td></td>
<td>x = 13.25</td>
<td>x = 13.06</td>
<td></td>
</tr>
</tbody>
</table>
results. Experienced programmers have learned to take advantage of the power of compilers to assist them in eliminating syntactic errors while novices have not. Novices find all types of errors a challenge to their diagnostic skills while experienced programmers have learned to focus their attention on developing strategies that make them more productive in finding logic errors. Observation of the mean logic error task values in Table 1 supports this conjecture, since in all cases the means for experienced programmers are higher than those for inexperienced programmers. Even though the means for the logic error tasks are higher for experienced programmers, there is not enough difference to be of concern in this study, as indicated by the t values and the associated probabilities (see Table 1). Based upon these results the experience variable was discarded to maintain a sample size of 30.

Preliminary Analysis

Graphical representations of the relationships between GEFT scores and composite diagnostic task scores were constructed using SAS. Ten different plots were obtained (see Figures 1-10). No readily apparent relationship between GEFT scores and syntactic error diagnostic task scores exists, regardless of scoring
FIGURE 1: SYNTAX TASK – SCORING SCHEME 1 (S1) BY GEFT
FIGURE 2: SYNTAX TASK – SCORING SCHEME 2 (S2) BY GEFT
FIGURE 3: SYNTAX TASK – SCORING SCHEME 3 (S3) BY GEFT
FIGURE 4: SYNTAX TASK – SCORING SCHEME 4 (S4) BY GEFT
FIGURE 5: SYNTAX TASK – SCORING SCHEME 5 (S5) BY GEFT
FIGURE 6: LOGIC TASK – SCORING SCHEME 1 (L1) BY GEFT
FIGURE 7: LOGIC TASK – SCORING SCHEME 2 (L2) BY GEFT
FIGURE 8: LOGIC TASK – SCORING SCHEME 3 (L3) BY GEFT
FIGURE 9: LOGIC TASK – SCORING SCHEME 4 (L4) BY GEFT
scheme employed. The relationship between logic error
diagnostic task scores and GEFT scores is somewhat
better defined but still rather vague. To obtain more
definitive results a correlation analysis of the
relationship between GEFT scores and diagnostic task
scores was conducted.

Correlation Analysis

The data collected counted responses, therefore an
interval measurement scale was used. Since the
distribution of scores could not be assumed normal,
the measure of correlation used was Spearman's Rho,
defined as follows:

$$\rho = \frac{1 - 6T}{n(n^2 - 1)}$$

where $T = \sum_{i=1}^{n} [R(X_i) - R(Y_i)]^2$,

R means the rank of observations, and X and Y are the
observations (Conover 1971, p. 246).

Spearman correlation coefficients for the
relationship between GEFT scores and composite
diagnostic task scores using scoring schemes one
through five were calculated using SAS (see Table 2).
The results of this analysis are reported below by
hypothesis and scoring scheme. In all cases $N = 30$.
### TABLE 2: SPEARMAN CORRELATION COEFFICIENTS

**GEFT VERSUS DIAGNOSTIC TASK SCORES**

*(N = 30)*

<table>
<thead>
<tr>
<th>Syntax Task</th>
<th>Scoring Scheme 1</th>
<th>Scoring Scheme 2</th>
<th>Scoring Scheme 3</th>
<th>Scoring Scheme 4</th>
<th>Scoring Scheme 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEFT</td>
<td>.160</td>
<td>.198</td>
<td>.124</td>
<td>.173</td>
<td>.220</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logic Task</th>
<th>Scoring Scheme 1</th>
<th>Scoring Scheme 2</th>
<th>Scoring Scheme 3</th>
<th>Scoring Scheme 4</th>
<th>Scoring Scheme 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEFT</td>
<td>.276</td>
<td>.256</td>
<td>.228</td>
<td>.352</td>
<td>.332</td>
</tr>
</tbody>
</table>
Hypothesis I

Scoring Scheme one: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the syntactic error diagnostic task using scoring scheme one was .160, insignificant at the .05 level.

Scoring Scheme Two: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the syntactic error diagnostic task using scoring scheme two was .198, insignificant at the .05 level.

Scoring Scheme Three: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the syntactic error diagnostic task using scoring scheme three was .124, insignificant at the .05 level.

Scoring Scheme Four: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the syntactic error diagnostic task using scoring scheme four was .173, insignificant at the .05 level.

Scoring Scheme Five: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the syntactic error diagnostic task using scoring scheme five was .220, insignificant at the .05 level.
Based upon the above results, Hypothesis I cannot be rejected, regardless of scoring scheme employed to calculate diagnostic task scores. This result is expected since the literature indicates that embedded figures test scores should not be related to scores on a syntactic error diagnostic task because the need for restructuring skill is minimal.

**Null Hypothesis II**

Scoring Scheme One: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the logic error diagnostic task using scoring scheme one was .276, insignificant at the .05 level.

Scoring Scheme Two: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the logic error diagnostic task using scoring scheme two was .256, insignificant at the .05 level.

Scoring Scheme Three: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the logic error diagnostic task using scoring scheme three was .228, insignificant at the .05 level.

Scoring Scheme Four: The Spearman correlation coefficient for the relationship between GEFT scores
and a composite score on the logic error diagnostic task using scoring scheme four was .352, insignificant at the .05 level.

Scoring Scheme Five: The Spearman correlation coefficient for the relationship between GEFT scores and a composite score on the logic error diagnostic task using scoring scheme five was .332, insignificant at the .05 level.

Based upon the above results, Hypothesis II cannot be rejected, regardless of scoring scheme employed to calculate diagnostic task scores. This result is inconsistent with research findings indicating that scores on an embedded figures test should be related to scores on diagnostic tasks that require considerable restructuring skill. It should be noted that if it could be assumed that higher scores on the GEFT could be paired with higher scores on the diagnostic tasks then a one-tailed test of significance could be performed. Under these conditions the values of Spearman's rho obtained when scoring schemes three and four are used to compute composite logic error diagnostic task scores would have indicated that the null hypothesis be rejected.

If a level of significance of .10 had been chosen and the pairing assumption had been made, then values
for Spearman’s rho obtained when scoring schemes one, two, four, and five are used in computation of scores would have indicated rejection of the null hypothesis. These findings provide evidence for rejection of hypothesis II if the higher alpha level is acceptable. The results obtained using scoring scheme three may indicate that a non-zero weight must be assigned to the number of errors located but not corrected, as indicated by implications of the findings of cognitive style research (see Justification of Scoring Schemes).

Supplemental Analysis

Subjects were surveyed to obtain data concerning the number of semesters of high school and college-level mathematics and computer science courses completed, work-related programming experience, age, and gender. The data collected for their college-level mathematics and computer programming experience was verified by consulting records in the Registrar’s Office at Boise State University. Spearman correlation coefficients were calculated to determine the relationship between each of the variables mentioned above and each diagnostic task score (see Table 3). Results indicated that the number of semesters of high school level mathematics courses completed was significantly related (at the .05 level) to composite
TABLE 3: SPEARMAN CORRELATION COEFFICIENTS

TASK SCORES VERSUS SEVERAL VARIABLES
(N = 30)

<table>
<thead>
<tr>
<th></th>
<th>SYNTAX TASK</th>
<th>LOGIC TASK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>scoring scheme 1</td>
<td>scoring scheme 2</td>
</tr>
<tr>
<td>HS CS</td>
<td>-.319</td>
<td>-.287</td>
</tr>
<tr>
<td>COL CS</td>
<td>-.268</td>
<td>-.319</td>
</tr>
<tr>
<td>HS MTH</td>
<td>.173</td>
<td>.153</td>
</tr>
<tr>
<td>COL MTH</td>
<td>-.107</td>
<td>-.105</td>
</tr>
<tr>
<td>EXPER</td>
<td>.068</td>
<td>.067</td>
</tr>
<tr>
<td>CM GPA</td>
<td>.072</td>
<td>.124</td>
</tr>
<tr>
<td>AGE</td>
<td>.083</td>
<td>.047</td>
</tr>
<tr>
<td></td>
<td>.288</td>
<td>.280</td>
</tr>
<tr>
<td>COL CS</td>
<td>.061</td>
<td>.084</td>
</tr>
<tr>
<td>HS MTH</td>
<td>.395*</td>
<td>.384*</td>
</tr>
<tr>
<td>COL MTH</td>
<td>.170</td>
<td>.146</td>
</tr>
<tr>
<td>EXPER</td>
<td>.224</td>
<td>.193</td>
</tr>
<tr>
<td>CM GPA</td>
<td>.160</td>
<td>.189</td>
</tr>
<tr>
<td>AGE</td>
<td>.253</td>
<td>.176</td>
</tr>
</tbody>
</table>

* Significant at the .05 level

HS CS - # of semesters of high school computer science
COL CS - # of semesters of college computer science
HS MTH - # of semesters of high school mathematics
COL MTH - # of semesters of college mathematics
EXPER - # of years of job-related programming experience
CM GPA - College Grade Point Average, mathematics only
AGE - chronological age in years
logic error diagnostic task scores, for all scoring schemes used to calculate composite score, except scoring scheme three. It should be noted that these data were not verified, primarily because high school records were not readily accessible. Since providing evidence to support this finding was not the primary purpose of this study it is not possible to generalize this result (see Chapter VI). No other significant correlations were obtained from the analysis performed.
CHAPTER VI

DISCUSSION

The major conceptual hypothesis of this study is that the cognitive styles of novice programmers are related to their ability to locate and correct logic errors in computer programs. Based upon findings reported in the literature, two hypotheses relating GEFT scores to diagnostic task scores and five scoring schemes for weighting composite diagnostic task scores were developed and tested (see Chapter 3). The results of correlation analysis of GEFT scores and diagnostic task scores failed to support the hypotheses of this study.

Restructuring Tasks and Error Type

Syntactic errors in computer programs are deviations from the rules governing the allowable forms of computer program statements (Wertz, 1982). Programs called compilers are available that locate syntactic errors. Although compiler diagnostics are occasionally vague and misleading, novice programmers quickly learn to use them. It would appear that locating and correcting syntactic errors in computer programs would be a fairly trivial task (Wertz, 1982).

Findings from the available literature (Goodman, 1971) indicate that performance on an embedded figures
test was not significantly related to diagnostic tasks when the need for restructuring skill was relatively limited. The first hypothesis of this study states that the cognitive styles of novice programmers are not related to their ability to locate and correct syntactic errors in computer programs. This hypothesis was supported by results from the present study. Thus, within the limitations of this study, a result consistent with the available literature appears also to exist for novice COBOL student programmers.

Logic errors occur when a computer program does not run as planned (Wertz, 1982). Run-time diagnostics are provided by most systems (Welburn, 1986) but offer little assistance in diagnosing logic errors. Thus programmers must use other methods. One method commonly employed is to trace the logical flow of data through a program by testing program operation with trial data values (Gould, 1975; Welburn, 1986). It appears that cognitive skills different than those needed to correct syntactic errors are needed to correct logic errors (Wertz, 1982).

Continued studies (Adejumo, 1983; Cheney, 1980; Glucksberg, 1956; Goodman, 1971; Witkin et al., 1962; Katz & Postal, 1964; Lefever & Ehri, 1976; MacKay & Bever, 1967; Tillman, 1979) have shown that field
dependent subjects have more difficulty than field
independent subjects in solving problems classified as
restructuring tasks. The task of locating and
correcting logic errors in computer programs qualifies
as a restructuring task since it requires that an
initial problem plan be reformulated to eliminate
faulty program performance (Collins & White, 1984;

The second hypothesis of this study states that
cognitive styles of novice programmers are related to
their ability to locate and correct logic errors in
computer programs. Results from the present study
failed to support this hypothesis. Thus, within the
limitations of this study, a result consistent with
the available literature does not appear to exist for
novice COBOL student programmers.

Educational Implications

Recent developments in technology have greatly
increased the use of computers for information
management. Technology alone will not insure the
success of information systems without utilizing human
resources to their full potential (Carroll, 1982;
Moosbrucker & Loflin, 1982; Steele & White, 1982). One
way to fully utilize human resources is to stress a
team effort (Burstein, 1985; Kaiser & Bostrom, 1982;
Locander, Napier, & Scamell, 1970; White, 1984a). In many work settings, programs are being written by design teams made up of program designers, managers, and coders working with formative researchers in a collaborative environment (Pea & Kurland, 1983).

Research in Management Information Systems (MIS) indicates that for optimal performance programming teams should contain a mix of personality types (White, 1984b). Jung (1923) postulated that personality types have different components, one of which is an information gathering component, characteristic of the cognitive styles described in the present research. Research indicating that the perceptual nature of a programming team is a factor in team productivity signifies that research in computer programming should address cognitive style differences of students. The purpose of the present study was to determine if cognitive style differences were an important factor in subjects' ability to debug computer programs. There is no evidence from the present study to support the claim that individuals possessing a global cognitive style will be at a disadvantage when involved in the cognitive subtask of program comprehension and debugging.

Research (Collins & White, 1984; Cheney, 1980) indicates that students do not necessarily exhibit the
same level of performance on the different component skills of programming during training. The opinion of some researchers (Cheney, 1980; Collins & White, 1984) is that cognitive style differences should be considered when designing methods of instruction. Unfortunately the present study failed to support the hypothesis that analytic problem solvers will perform better on diagnostic restructuring tasks than will global problem solvers, perhaps indicating that additional work in this area is necessary to support the opinions of the previously cited researchers. On the other hand, the results of the present study may indicate that cognitive style differences represent an unsatisfactory basis for designing methods of instruction that pertain to the cognitive subtask of program comprehension and debugging.

Discussion of Supplemental Analysis

The results of the supplemental analysis indicate a significant linear relationship (at the .05 level) between the number of semesters of high school mathematics and composite scores on the logic error diagnostic tasks. This result applies for all scoring schemes except scheme three. This result appears to provide evidence accounting for approximately 16 percent of the variance involved, indicating that
previous mathematics training may be a more important factor in predicting success in learning diagnostic skills than cognitive style differences. This result appears consistent with previous studies (Wileman, Konvalina, & Stephans, 1981) indicating that high school mathematics experience is a factor contributing to overall performance in introductory programming courses.

Suggestions for further research

One of the limitations of this study was that diagnostic skills were measured by paper-and-pencil exams. It would be interesting to develop an interactive debugging environment so that these skills could be evaluated in a more realistic setting. Development of a logic error classification scheme, in terms of levels of difficulty of such errors, should be attempted to determine if such a taxonomy could provide a means for refining the scoring schemes used in this study.

Additional research into the cognitive style construct itself is needed. As indicated by Huber (1983) the present definition of this construct is far too general. Further research may indicate that subcomponents of cognitive style exist. In addition research is needed to determine if individuals exhibit
different cognitive styles under different problem situations. For example, could an individual display analytic skills (field independence) when solving a problem in finance, but take a global (field dependence) approach when confronted with a problem in physics?

Additional research into training effects is also needed. Can subjects' "natural" propensities be changed by training? With maturation and/or the proper training can subjects develop an analytical approach to problem solving in certain domains, perhaps similar to the way one develops from the concrete operational stage to the stage of formal operations as described by Jean Piaget?

The relationship of cognitive style differences of individuals to the other cognitive skills involved in the progression through the learning stages of programming should be investigated. These skills, as defined by Shneiderman (1976) are comprehension, composition, debugging, and modification. Although the present study failed to produce significant results for the skill of debugging, further research to learn more about these skills is needed. Collins and White (1984) for example, have shown a significant relationship between debugging ability and
composition. A study investigating the role of cognitive style differences in composition may be indicated.

Further investigation into the relationship between affinity for high school mathematics and component cognitive skills of programming should be attempted. The results of the present study could serve as a basis for such research. It might be of interest to investigate the relationship between other high school subjects, besides mathematics, to determine what role they play in developing the necessary cognitive skills of computer programming.


Guethzow, H. (1951). An analysis of the operation of
Psychology. 45: 219-244.

Guilford, J., Berger, R., & Christensen, P. (1955a). A
factor-analytic study of planning, II.
Administration of test and analysis of results.
(Rep. psychol. Lab., No. 12). Los Angeles:
University of Southern California.

Guilford, J., Frick, J., Christensen, P., &
Merrifield, P. (1957). A factor-analytic study of
flexibility in thinking. (Rep. psychol. Lab., No. 18). Los Angeles: University of Southern
California.

The relation of certain thinking factors to
training criteria in the U. S. Coast Guard Academy.
(Rep. psychol. Lab., No. 13). Los Angeles:
University of Southern California.

Guilford, J., Wilson, R., & Christensen, P. (1952). A
factor-analytic study of creative thinking, II.
Administration of test and analysis of results.
(Rep. psychol. Lab., No. 8). Los Angeles:
University of Southern California.

Hartlage, L. (1970). Sex-linked inheritance of spatial
ability. *Perceptual and Motor Skills. 31: 610.*


Laosa, L., & DeAvila, E. (1979). Development of cognitive styles among Chicanos in traditional and


Loeff, R. G. (1961). *Embedding and distracting field contexts as related to the field dependence*


Sieber, J. & Lanzetta, J. (1966). Some determinants of individual differences in predecision information-


Conference of the Special Interest Group for Computer Personnel Research. 11, 49-61.


APPENDIX A

COBOL Source Code
Syntactic Error Diagnostic Tasks
The source code for the two COBOL programs that constituted the syntactic error diagnostic task for this study follows:

**SYNTAX TASK - PART ONE**

```cobol
IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM-TWO2.
AUTHOR. BRENDER.
DATE-WRITTEN. SEPTEMBER 2, 1986.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT SALESPERSON-FILE-IN
ASSIGN TO SYS008-UT-3340-S.
SELECT SALESPERSON-LIST-OUT
ASSIGN TO SYS005-UR-1403-S.
DATA DIVISION.
FILE SECTION.
FD SALESPERSON-FILE-IN
RECORD CONTAINS 80 CHARACTERS
LABEL RECORDS ARE STANDARD
DATA RECORD IS SR-SALESPERSON-RECORD-IN

01 SR-SALESPERSON-RECORD-IN.
  05 SR-RECORD-CODE-IN PIC 9(2).
  05 SR-REGION-IN PIC X(26).
  05 SR-TERRITORY-IN PIC 9(4).
  05 SR-SALESPERSON-NUMBER-IN PIC 9(3).
  05 SR-SALESPERSON-NAME-IN PIC X(26).
  05 FILLER PIC X(12).
  05 SR-PRODUCT-UNITS-SOLD-IN PIC 9(3).
  05 FILLER PIC 9(30).

FD SALESPERSON-LIST-OUT
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED
```
DATA RECORD IS SL-SALESPERSON-LINE-OUT.

01 SL-SALESPERSON-LINE-OUT.
  05 FILLER PIC X(1).
  05 FILLER PIC X(6).
  05 SL-REGION-OUT PIC X(2).
  05 FILLER PIC X(3).
  05 SL-TERRITORY-OUT PIC 9(4).
  05 FILLER PIC X(3).
  05 SL-SALESPERSON-NUMBER-OUT PIC 9(3).
  05 FILLER PIC X(3).
  05 SL-SALESPERSON-NAME-OUT PIC X(26).
  05 FILLER PIC X(3).
  05 SL-PRODUCT-UNITS-SOLD-OUT PIC 9(3).
  05 FILLER PIC X(76).

WORKING-STORAGE SECTION.

01 WS-PROGRAM-SWITCHES.
  05 WS-END-OF-FILE-SWITCH PIC X(3).

PROCEDURE DIVISION.

MAINLINE-PRINT-SALESPERSON-LIST.

OPEN INPUT SALESPERSON-FILE-IN
OUTPUT SALESPERSON-LIST-OUT.
PERFORM INITIALIZE-VARIABLE-FIELDS.
READ SALESPERSON-FILE-IN
AT END MOVE "YES" TO WS-EOF-SWITCH.
PERFORM PROCESS-SALESPERSON-RECORD
UNTIL WS-END-OF-FILE-SWITCH IS EQUAL TO "YES".
CLOSE SALESPERSON-FILE-IN
SALESPERSON-LIST-OUT.
STOP RUN.

INITIALIZE-VARIABLE-FIELDS.

MOVE "NO " TO WS-END-OF-FILE-SWITCH.

PROCESS-SALESPERSON-RECORD.

MOVE SPACES TO SL-SALESPERSON-LINE-OUT.
MOVE SR-REGION-IN TO SL-REGION-OUT.
MOVE SR-TERRITORY-IN TO SL-TERRITORY-OUT.
MOVE SR-SALESPERSON-NUMBER-IN TO SL-SALESPERSON-NUMBER-OUT.
MOVE SR-SALESPERSON-NAME-IN TO SL-SALESPERSON-NAME-OUT.
MOVE SR-PRODUCT-UNITS-SOLD-IN TO SL-PRODUCT-UNITS-SOLD-OUT.
WRITE SL-SALESPERSON-LINE-OUT AFTER ADVANCING 2 LINES.
READ SALESPERSON-FILE-IN.
AT END MOVE "YES" TO WS-END-OF-FILE SWITCH

SYNTAX TASK - PART TWO

IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM-73 SALES QUOTA REPORT.
AUTHOR. BRENDER.
DATE-COMPILED.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
SELECT SALESPERSON-FILE
   ASSIGN TO SYS008-UT-3340-S.
SELECT SALES-QUOTA-REPORT
   ASSIGN TO SYS005-UR-1403-S.

DATA DIVISION.

FILE SECTION.
FD SALESPERSON-FILE
   RECORD CONTAINS 80 CHARACTERS
LABEL RECORDS ARE STANDARD.

01 SALESPERSON-RECORD.
  05 FILLER PIC X(80).

FD SALES-QUOTA-REPORT
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED.

01 REPORT-LINE
  05 FILLER PIC X(133).

WORKING STORAGE SECTION.

01 WS-SWITCHES.
  05 WS-END-OF-FILE-SWITCH PIC X(3).

01 WS-REPORT-CONTROLS.
  05 WS-PAGE-COUNT PIC S9(3).
  05 WS-LINES-USED PIC S9(2).
  05 WS-LINE-SPACING PIC S9(2).

01 WS-WORK-AREAS.
  05 WS-DATE-WORK PIC 9(6).
  05 WS-DATE-REFORMAT REDEFINES WS-DATE-_WORK.
    10 WS-YEAR PIC 9(2).
    10 WS-MONTH PIC 9(2).
    10 WS-DAY PIC 9(3).
  05 WS-PERCENT-OF-QUOTA S9(3)V99.

01 WS-TOTAL-ACCUMULATORS.
  05 WS-TOTAL-SALESPERSONS ACCUM PIC S9(4).
  05 WS-TOTAL-SALES-QUOTA ACCUM PIC S9(8)V.
  05 WS-TOTAL-SALES-REVENUE ACCUM PIC S9(9)V99.

01 SR-SALESPERSON-RECORD.
  05 SR-RECORD-CODE PIC 9(2).
  05 SR-REGION PIC X(2).
  05 SR-TERRITORY PIC 9(4).
  05 SR-SALESPERSON-NUMBER PIC 9(3).
  05 SR-SALESPERSON-NAME PIC X(26).
  05 FILLER PIC X(23).
05 SR-SALES-REVENUE PIC S9(8)V99.
05 SR-SALES-QUOTA PIC S9(7)V99.
05 FILLER PIC X(1).
01 H1-HEADING-LINE-1.
05 FILLER PIC X(1) VALUE SPACE.
05 FILLER PIC X(20) VALUE "PYRAMID SALES COMPANY".
05 FILLER PIC X(20) VALUE "Y".
05 FILLER PIC X(20) VALUE " ".
05 FILLER PIC X(12) VALUE "RUN DATE ".
05 H1-MONTH PIC 9(2).
05 FILLER PIC X(1) VALUE "/".
05 H1-DAY PIC 9(2).
05 FILLER PIC X(20) VALUE "/".
05 H1-YEAR PIC 9(2) VALUE "327".
05 FILLER PIC X(52) VALUE SPACES.
01 H2-HEADING-LINE-2.
05 FILLER PIC X(1) VALUE SPACE.
05 FILLER PIC X(20) VALUE "SALES QUOTA REPORT (".
05 FILLER PIC X(20) VALUE "7-3)"
05 FILLER PIC X(20) VALUE " ".
05 FILLER PIC X(17) VALUE "PAGE ".
05 H2-PAGE-NBR PIC ZZ9.
05 FILLER PIC X(52) VALUE SPACES.
01 H3-HEADING-LINE-3.
05 FILLER PIC X(1) VALUE SPACE.
05 FILLER PIC X(20) VALUE "REGION----------S".
05 FILLER PIC X(20) VALUE "ALESPERSON----------".
05 FILLER PIC X(20) VALUE " SALES ".
05 FILLER PIC X(20) VALUE " SALES PERCENT".
05 FILLER PIC X(20) VALUE " ".
05 FILLER PIC X(20) VALUE " ".
05 FILLER PIC X(12) VALUE ".".
114  *  H4-HEADING-LINE-4.
115    01  FILLER PIC X(1) VALUE SPACE.
116    05  FILLER PIC X(20) VALUE
117        " TERR. NBR NAME".
118    05  FILLER PIC X(20) VALUE
119        " ".
120    05  FILLER PIC X(20) VALUE
121        " QUOTA ".
122    05  FILLER PIC X(20) VALUE
123        " ".
124  *  DL-Detail-Line.
125    01  FILLER VALUE SPACES.
126    05  FILLER PIC X(1).
127    05  DL-REGION PIC X(2).
128    05  FILLER PIC X(4)
129        VALUE ".-".
130    05  FILLER PIC X(2)
131        VALUE SPACES.
132    05  DL-TERRITORY PIC 9(4).
133    05  FILLER VALUE SPACES.
134    05  DL-SALESPERSON-NUMBER PIC 9(3).
135    05  FILLER VALUE SPACES.
136    05  DL-SALESPERSON-NAME PIC X(26).
137    05  DL-SALES-QUOTA Z,ZZZ,ZZZ.99.
138    05  FILLER VALUE SPACES.
139    05  DL-SALES-REVENUE ZZ,ZZZ,ZZ9.29.
140    05  FILLER VALUE SPACES.
141    05  DL-PERCENT-OF-QUOTA PIC ZZ9.
142  *  TL-TOTAL-LINE.
143    01  FILLER VALUE SPACES.
144    05  FILLER PIC X(1)
145    05  FILLER VALUE SPACES.
PROCEDURE DIVISION.

000-PRINT-SALES-QUOTA-REPORT.

OPEN INPUT SALESPERSON-FILE
OUTPUT SALES-QUOTA-REPORT.
PERFORM 100-INITIALIZE-VARIABLE-FIELD.
PERFORM 800-READ-SALESPERSON-RECORD.
PERFORM 200-PROCESS-SALESPERSON-RECORD
UNTIL WSENDOFFILESWITCH IS EQUAL TO "YES".
PERFORM 700-PRINT-TOTAL-LINE.
CLOSE SALESPERSON-FILE
SALES-QUOTA-REPORT.
STOP RUN.

100-INITIALIZE-VARIABLE-FIELDS.

MOVE "NO" TO WSENDOFFILESWITCH.
MOVE ZEROS TO WSPAGECOUNT.
MOVE WSLINESPERPAGE TO WSLINESUSED.
ACCEPT WSDATEWORK FROM DATE.
MOVE WSMONTH TO H1MONTH.
MOVE WSDAY TO H1DAY.
MOVE WSYEAR TO H1YEAR.
MOVE ZEROS TO WS-TOTAL-ACCUMULATORS.

200-PROCESS-SALESPERSON-RECORD.

IF WS-LINES-USED IS EQUAL TO WS-LINES-PER-PAGE
  PERFORM 870-PRINT-REPORT-HEADINGS.
  MOVE SR-REGION TO DL-REGION.
  MOVE SR-TERRITORY TO DL-TERRITORY.
  MOVE SR-SALESPERSON-NUMBER TO DL-SALESPERSON-NUMBER.
  MOVE SR-SALESPERSON-NAME TO DL-SALESPERSON-NAME.
  MOVE SR-SALES-QUOTA TO DL-SALES-QUOTA.
  MOVE SR-SALES-REVENUE TO DL-SALES-REVENUE.
  IF SR-SALES-QUOTA IS NOT EQUAL TO ZERO
    DIVIDE SR-SALES-REVENUE BY SR-SALES-QUOTA
    GIVING WS-PERCENT-OF-QUOTA ROUNDED
    MULTIPLY 100 BY WS-PERCENT-OF-QUOTA
    MOVE WS-PERCENT-OF-QUOTA TO DL-PERCENT-OF-QUOTA.
  ELSE
    MOVE ZERO TO DL-PERCENT-OF-QUOTA.
  MOVE DL-DETAIL-LINE TO REPORT-LINE.
  MOVE 2 TO WS-LINE-SPACING.
  PERFORM 890-WRITE-REPORT-LINE.
  ADD 1 TO WS-TOTAL-SALESPERSONS-ACCUM.
  ADD SR-SALES-QUOTA TO WS-TOTAL-SALES-QUOTA-ACCUM.
  ADD SR-SALES-REVENUE TO WS-TOTAL-SALES-REVENUE-ACCUM.
  PERFORM 800-READ-SALESPERSON-RECORD.

700-PRINT-TOTAL-LINE.

MOVE WS-TOTAL-SALESPERSONS-ACCUM TO TL-TOTAL-SALESPERSONS.
MOVE WS-TOTAL-SALES-QUOTA-ACCUM TO TL-TOTAL-SALES-QUOTA.
MOVE WS-TOTAL-SALES-REVENUE-ACCUM TO TL-TOTAL-SALES-REVENUE.
DIVIDE WS-TOTAL-SALES-REVENUE-ACCUM
  BY WS-TOTAL-SALES-QUOTA-ACCUM
  GIVING WS-PERCENT-OF-QUOTA ROUNDED.
MULTIPLY 100 BY WS-PERCENT-OF-QUOTA.
MOVE WS-PERCENT-OF-QUOTA TO TL-PERCENT-OF-QUOTA.
MOVE TL-TOTAL-LINE TO REPORT-LINE.
MOVE 3 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
*
800-READ-SALESPERSON-RECORD.
*
READ SALESPEOPRSON-FILE INTO SR-SALESPERSON-RECORD
AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.
*
870-PRINT-REPORT-HEADINGS.
*
ADD 1 TO WS-PAGE-COUNT.
MOVE WS-PAGE-COUNT TO H2-PAGE-NBR.
MOVE H1-HEADING-LINE-1 TO REPORT-LINE.
PERFORM 880-WRITE-REPORT-TOP-LINE.
MOVE HEADING-LINE-2 TO REPORT-LINE.
MOVE 1 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
MOVE HEADING-LINE-3 TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
MOVE H4-HEADING-LINE-4 TO REPORT-LINE.
MOVE 1 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
*
880-WRITE-REPORT-TOP-LINE.
*
WRITE REPORT-LINE
AFTER ADVANCING PAGE.
MOVE 1 TO WS-LINES-USED.
*
890-WRITE-REPORT-LINE.
*
WRITE REPORT-LINE
AFTER ADVANCING WS-LINE-SPACING LINES.
ADD WS-LINE-SPACING TO WS-LINES-USED.
APPENDIX B

COBOL Source Code
Logic Error Diagnostic Tasks
The source code for the five COBOL programs that constituted the logic error diagnostic task for this study follows:

LOGIC TASK - PART ONE

1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. PROGRAM-10.
3 AUTHOR. BRENDER.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
SELECT EARNINGS-FILE
ASSIGN TO SYS008-UT-3340-S.
SELECT DEPARTMENTAL-EARNINGS-REPORT
ASSIGN TO SYS005-UR-1403-S.

DATA DIVISION.

FILE SECTION.

FD EARNINGS-FILE
RECORD CONTAINS 80 CHARACTERS
LABEL RECORDS ARE STANDARD
DATA RECORD IS EARNINGS-RECORD.

01 EARNINGS-RECORD.
  05 FILLER PIC X(80).

FD DEPARTMENTAL-EARNINGS-REPORT
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED
DATA RECORD IS REPORT-LINE.

01 REPORT-LINE.
  05 FILLER PIC X(133).

WORKING- STORAGE SECTION.
01 WS-SWITCHES.
   05 WS-END-OF-FILE-SWITCH PIC X(3).
   88 END-OF-FILE VALUE "YES".

01 WS-CONTROL-FIELDS.
   05 WS-PREVIOUS-PLANT-CODE PIC X(3).
   05 WS-PREVIOUS-DEPARTMENT-NUMBER PIC X(4).

01 WS-REPORT-CONTROLS.
   05 WS-PAGE-COUNT PIC S9(4) COMP-3.
   05 WS-LINES-PER-PAGE PIC S99 VALUE +54 COMP SYNC.
   05 WS-LINES-USED PIC S9(2) COMP SYNC.
   05 WS-LINE-SPACING PIC S9(2).

01 WS-GROUP-INDICATE-CONTROLS.
   05 WS-DEPT-CODE-CONTROL PIC X(3) VALUE "YES".
   05 WS-PLANT-CODE-CONTROL PIC X(3) VALUE "YES".

01 WS-WORK-AREAS.
   05 WS-DATE-WORK PIC 9(6).
   05 WS-DATE-REFORMAT REDEFINES WS-DATE-WORK.
   10 WS-YEAR PIC 9(2).
   10 WS-MONTH PIC 9(2).
   10 WS-DAY PIC 9(2).

01 WS-TOTAL-ACCUMULATORS.

01 ER-EARNINGS-RECORD.
ER-RECORD-CODE PIC X(2).
FILLER PIC X(1).
ER-PLANT-CODE PIC X(3).
ER-DEPARTMENT-NUMBER PIC X(4).
ER-EMPLOYEE-NUMBER PIC X(9).
ER-LAST-NAME PIC X(12).
ER-FIRST-NAME PIC X(9).
ER-MIDDLE-NAME PIC X(8).
ER-SEX-CODE PIC X(1).
ER-MARITAL-STATUS PIC X(1).
ER-NUMBER-OF-EXEMPTIONS PIC X(2).
ER-PAY-CODE PIC X(1).
ER-PAY-RATE.
10 ER-PAY-RATE-HOURLY PIC 9(2)V9999.
10 ER-PAY-RATE-SALARIED REDEFINES ER-
PAY-RATE-HOURLY PIC 9(4)V99.
ER-THIS-PERIOD-EARNINGS PIC 9(5)V99.
ER-YEAR-TO-DATE-EARNINGS PIC 9(6)V99.
ER-DATE-LAST-ACTIVITY PIC X(6).
DR-DATE-RECORD REDEFINES ER-EARNINGS-
RECORD.
ER-RECORD-CODE PIC X(2).
DATE-RECORD VALUE "01".
ER-PERIOD-ENDING-DATE PIC X(6).
FILLER PIC X(72).
* H1-HEADING-LINE-1.
FILLER PIC X(1) VALUE SPACES.
DEPARTMENTAL EARNING.

"S REPORT (10-3)

"PE".

"RIOD ENDING ".

"DEPARTMENTAL EARNING:

"S REPORT (10-3)

"PE".

"RIOD ENDING ".

"DEPARTMENTAL EARNING:

"S REPORT (10-3)

"PE".

"RIOD ENDING ".

"DEPARTMENTAL EARNING:

"S REPORT (10-3)

"PE".

"RIOD ENDING ".

"DEPARTMENTAL EARNING:

"S REPORT (10-3)

"PE".

"RIOD ENDING ".

"DEPARTMENTAL EARNING:

"S REPORT (10-3)

"PE".

"RIOD ENDING ".

"DEPARTMENTAL EARNING:

"S REPORT (10-3)

"PE".

"RIOD ENDING ".
05 FILLER PIC X(1) VALUE SPACES.
05 FILLER PIC X(20) VALUE "PLANT DEPT EMPL N".
05 FILLER PIC X(20) VALUE "UMBER LAST NAME ".
05 FILLER PIC X(20) VALUE " FIRST NM THIS ".
05 FILLER PIC X(20) VALUE "PER YEAR-TO-DT ".
05 FILLER PIC X(20) VALUE " ".
05 FILLER PIC X(12) VALUE " ".
01 DL-DETAIL-LINE.
05 FILLER VALUE SPACES.
05 DL-PLANT-CODE PIC X(3).
05 FILLER VALUE SPACES.
05 DL-DEPARTMENT-NUMBER PIC X(4).
05 FILLER VALUE SPACES.
05 DL-EMPLOYEE-NUMBER XXXBXXBXXXX.
05 FILLER VALUE SPACES.
05 DL-LAST-NAME PIC X(12).
05 FILLER VALUE SPACES.
05 DL-FIRST-NAME PIC X(9).
05 FILLER VALUE SPACES.
05 DL-THIS-PERIOD-EARNINGS ZZ,ZZZ.99.
05 FILLER VALUE SPACES.
05 DL-YEAR-TO-DATE-EARNINGS ZZ,ZZZ.99.
05 FILLER VALUE SPACES.
01 CT-CONTROL-TOTAL-LINE.
05 FILLER X(22) VALUE SPACES.
05 CT-TOTAL-DESCRIPTION PIC X(15).
05 CT-DEPARTMENT-PLANT-AREA PIC X(15).
05 CT-DEPARTMENT-AREA REDEFINES CT-DEPARTMENT-PLANT-AREA.
PROCEDURE DIVISION.

000-PRINT-DEPT-EARNINGS-REPORT.

OPEN INPUT EARNINGS-FI LE
OUTPUT DEPARTMENTAL-EARNINGS-REPORT.
PERFORM 100-INITIALIZE-VARIABLE-FIELDS.
PERFORM 110-PROCESS-DATE-RECORD.
PERFORM 200-PROCESS-MAJOR-PLANT-GROUP
UNTIL END-OF-FILE.
PERFORM 700-PRINT-REPORT-TOTAL-LINE.
CLOSE EARNINGS-FI LE
DEPARTMENTAL-EARNINGS-REPORT.
STOP RUN.

100-INITIALIZE-VARIABLE-FIELDS.

MOVE "NO " TO WS-END-OF-FILE-SWITCH.
MOVE ZEROS TO WS-PAGE-COUNT.
ACCEPT WS-D ATE-WORK FROM DATE.
MOVE WS-M ONTH TO H2-MONTH.
MOVE WS-D AY TO H2-D AY.
MOVE WS-Y EAR TO H2-YEAR.
MOVE ZEROS TO WS-TOTAL-DEPT-PER-EARN-ACCUM.
MOVE WS-TOTAL-DEPT-YTD-EARN-ACCUM.
MOVE WS-TOTAL-PLANT-PER-EARN-ACCUM.
MOVE WS-TOTAL-PLANT-YTD-EARN-ACCUM.
* 110-PROCESS-DATE-RECORD.
  PERFORM 800-READ-EARNINGS-RECORD.
  IF DATE-RECORD
    MOVE DR-PERIOD-ENDING-DATE TO H1-PER-END-DATE
  ELSE
    PERFORM 800-READ-EARNINGS-RECORD
  END-OF-FILE-SWITCH.

* 200-PROCESS-MAJOR-PLANT-GROUP.
  MOVE ER-PLANT-CODE TO WS-PREVIOUS-PLANT-CODE.
  MOVE WS-LINES-PER-PAGE TO WS-LINES-USED.
  PERFORM 210-PROCESS-MINOR-DEPT-GROUP
  UNTIL ER-PLANT-CODE
    IS NOT EQUAL TO WS-PREVIOUS-PLANT-CODE.
  MOVE "YES" TO WS-PLANT-CODE-CONTROL.
  PERFORM 320-PRINT-PLANT-TOTAL-LINE.

* 210-PROCESS-MINOR-DEPT-GROUP.
  MOVE ER-DEPARTMENT-NUMBER TO WS-PREVIOUS-DEPARTMENT-NUMBER.
  MOVE ZEROS TO WS-TOTAL-DEPT-PER-EARN-ACCUM
  WS-TOTAL-DEPT-YTD-EARN-ACCUM.
  PERFORM 300-PRINT-DETAIL-LINE
  UNTIL ER-DEPARTMENT-NUMBER
    IS NOT EQUAL TO WS-PREVIOUS-DEPARTMENT-NUMBER
  OR ER-PLANT-CODE IS NOT EQUAL TO WS-PREVIOUS-PLANT-CODE.
  MOVE "YES" TO WS-DEPT-CODE-CONTROL.
  PERFORM 310-PRINT-DEPT-TOTAL-LINE.

* 300-PRINT-DETAIL-LINE.
  IF WS-LINES-USED IS NOT LESS THAN WS-LINES-PER-PAGE
    PERFORM 870-PRINT-REPORT-HEADINGS.
  IF WS-PLANT-CODE-CONTROL IS EQUAL TO "YES"
    MOVE ER-PLANT-CODE TO DL-PLANT-CODE
  ELSE MOVE SPACES TO DL-PLANT-CODE.
  IF WS-DEPT-CODE-CONTROL IS EQUAL TO "YES"
MOVE ER-DEPARTMENT-NUMBER TO DL-DEPARTMENT-NUMBER
ELSE
MOVE SPACES TO DL-DEPARTMENT-NUMBER.
MOVE ER-EMPLOYEE-NUMBER TO DL-EMPLOYEE-NUMBER.
INSPECT DL-EMPLOYEE-NUMBER REPLACING ALL SPACES BY "-".
MOVE ER-LAST-NAME TO DL-LAST-NAME.
MOVE ER-FIRST-NAME TO DL-FIRST-NAME.
MOVE ER-THIS-PERIOD-EARNINGS TO DL-THIS-PERIOD-EARNINGS.
MOVE ER-YEAR-TO-DATE-EARNINGS TO DL-YEAR-TO-DATE-EARNINGS.
MOVE DL-DETAIL-LINE TO REPORT-LINE.
PERFORM 890-WRITE-REPORT-LINE.
MOVE 1 TO WS-LINE-SPACING.
ADD ER-THIS-PERIOD-EARNINGS TO WS-TOTAL-DEPT-PER-EARN-ACCUM.
ADD ER-YEAR-TO-DATE-EARNINGS TO WS-TOTAL-DEPT-YTD-EARN-ACCUM.
PERFORM 800-READ-EARNINGS-RECORD.

MOVE "TOTAL FOR DEPT " TO CT-TOTAL-DESCRIPTION.
MOVE SPACES TO CT-DEPARTMENT-PLANT-AREA.
MOVE WS-PREVIOUS-DEPARTMENT-NUMBER TO CT-DEPARTMENT-NUMBER.
MOVE WS-TOTAL-DEPT-PER-EARN-ACCUM
to CT-TOTAL-THIS-PERIOD-EARNINGS.
MOVE WS-TOTAL-DEPT-YTD-EARN-ACCUM
to CT-TOTAL-YEAR-TO-DATE-EARNINGS.
MOVE "+" TO CT-ASTERISKS.
MOVE CT-CONTROL-TOTAL-LINE TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
ADD WS-TOTAL-DEPT-PER-EARN-ACCUM
to WS-TOTAL-PLANT-PER-EARN-ACCUM.
ADD WS-TOTAL-DEPT-YTD-EARN-ACCUM
to WS-TOTAL-PLANT-YTD-EARN-ACCUM.

MOVE "TOTAL FOR PLANT" TO CT-TOTAL-DESCRIPTION.
MOVE SPACES TO CT-DEPARTMENT-PLANT-AREA.
MOVE WS-PREVIOUS-PLANT-CODE TO CT-PLANT-CODE.
MOVE WS-TOTAL-PLANT-PER-EARN-ACCUM TO CT-TOTAL-THIS-PERIOD-EARNINGS.
MOVE WS-TOTAL-PLANT-YTD-EARN-ACCUM TO CT-TOTAL-YEAR-TO-DATE-EARNINGS.
MOVE "** " TO CT-ASTERISKS.
MOVE CT-CONTROL-TOTAL-LINE TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
ADD WS-TOTAL-PLANT-PER-EARN-ACCUM TO WS-TOTAL-RPT-PER-EARN-ACCUM.
ADD WS-TOTAL-PLANT-YTD-EARN-ACCUM TO WS-TOTAL-RPT-YTD-EARN-ACCUM.
*
700-PRINT-REPORT-TOTAL-LINE.
*
MOVE "REPORT TOTAL " TO CT-TOTAL-DESCRIPTION.
MOVE SPACES TO CT-DEPARTMENT-PLANT-AREA.
MOVE WS-TOTAL-RPT-PER-EARN-ACCUM TO CT-TOTAL-THIS-PERIOD-EARNINGS.
MOVE WS-TOTAL-RPT-YTD-EARN-ACCUM TO CT-TOTAL-YEAR-TO-DATE-EARNINGS.
MOVE "***" TO CT-ASTERISKS.
MOVE CT-CONTROL-TOTAL-LINE TO REPORT-LINE.
MOVE 3 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
*
800-READ-EARNINGS-RECORD.
*
READ EARNINGS-FILE INTO ER-EARNINGS-RECORD
AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH
MOVE HIGH-VALUES TO ER-PLANT-CODE
ER-DEPARTMENT-NUMBER.
*
870-PRINT-REPORT-HEADINGS.
*
ADD 1 TO WS-PAGE-COUNT.
MOVE WS-PAGE-COUNT TO H3-PAGE-NBR.
IF ER-PLANT-CODE IS EQUAL TO "ATL"
MOVE "ATLANTIC" TO H4-PLANT-NAME
ELSE IF ER-PLANT-CODE IS EQUAL TO "CTL"
MOVE "CENTRAL " TO H4-PLANT-NAME
ELSE IF ER-PLANT-CODE IS EQUAL TO "MTN"
MOVE "MOUNTAIN" TO H4-PLANT-NAME
ELSE IF ER-PLANT-CODE IS EQUAL TO "PAC"
    MOVE "PACIFIC" TO H4-PLANT-NAME
    ELSE MOVE "********" TO H4-PLANT-NAME.
    MOVE H1-HEADING-LINE-1 TO REPORT-LINE.
    PERFORM 880-WRITE-REPORT-TOP-LINE.
    MOVE H2-HEADING-LINE-2 TO REPORT-LINE.
    MOVE 1 TO WS-LINE-SPACING.
    PERFORM 890-WRITE-REPORT-LINE.
    MOVE H3-HEADING-LINE-3 TO REPORT-LINE.
    MOVE 1 TO WS-LINE-SPACING.
    PERFORM 890-WRITE-REPORT-LINE.
    MOVE H4-HEADING-LINE-4 TO REPORT-LINE.
    MOVE 2 TO WS-LINE-SPACING.
    PERFORM 890-WRITE-REPORT-LINE.
    MOVE H5-HEADING-LINE-5 TO REPORT-LINE.
    MOVE 2 TO WS-LINE-SPACING.
    PERFORM 890-WRITE-REPORT-LINE.
    MOVE 2 TO WS-LINE-SPACING.
  880-WRITE-REPORT-TOP-LINE.
  WRITE REPORT-LINE
  AFTER ADVANCING PAGE.
  MOVE 1 TO WS-LINES-USED.
  890-WRITE-REPORT-LINE.
  WRITE REPORT-LINE
  AFTER ADVANCING WS-LINE-SPACING.
  ADD WS-LINE-SPACING TO WS-LINES-USED.
LOGIC TASK - PART TWO

IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM-114.
AUTHOR. BRENDER.
DATE-WRITTEN. NOVEMBER 17, 1986.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
SELECT STUDENT-FILE
ASSIGN TO SYS008-UT-3340-S.
SELECT SORT-FILE
ASSIGN TO SYS001-UT-3340-S-SORTWK1.
SELECT STUDENT-GPA-YEAR-LIST
ASSIGN TO SYS005-UR-1403-S.

DATA DIVISION.
FILE SECTION.
FD STUDENT-FILE
RECORD CONTAINS 80 CHARACTERS
LABEL RECORDS ARE STANDARD.

FD STUDENT-GPA-YEAR-LIST
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED.

FD REPORT-LINE.
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED.

SD SORT-FILE
RECORD CONTAINS 80 CHARACTERS.

FD STUDENT-RECORD.
05 FILLER PIC X(80).

FD STUDENT-GPA-YEAR-LIST
05 FILLER PIC X(133).

SD REPORT-LINE.
05 FILLER PIC X(2).
05 FILLER PIC X(9).
05 FILLER PIC X(14).
05 FILLER PIC X(10).
05 FILLER PIC X(9).
WORKING-STORAGE SECTION.

01 WS-SWITCHES.
   05 WS-END-OF-FILE-SWITCH PIC X(3).
   88 END-OF-FILE VALUE "YES".

01 WS-REPORT-CONTROLS.
   05 WS-PAGE-COUNT PIC S9(3) COMP-3.
   05 WS-LINES-PER-PAGE PIC S9(2) VALUE +57 COMP.
   05 WS-LINES-USED PIC S9(2) COMP.
   05 WS-LINE-SPACING PIC S9(2).

01 WS-WORK-AREAS.
   05 WS-DATE-WORK PIC S9(6).
   05 WS-DATE-REFORMAT REDEFINES WS-DATE-WORK.
      10 WS-YEAR PIC 9(2).
      10 WS-MONTH PIC 9(2).
      10 WS-DAY PIC 9(2).

01 WS-TOTAL-ACCUMULATORS.
   05 WS-TOTAL-FRESHMEN-ACCUM PIC S9(5).
   05 WS-TOTAL-SOPHOMORES-ACCUM PIC S9(5).
   05 WS-TOTAL-JUNIORS-ACCUM PIC S9(5).
   05 WS-TOTAL-SENIORS-ACCUM PIC S9(5).

01 SR-STUDENT-RECORD.
   05 SR-RECORD-CODE PIC 9(2).
   05 SR-STUDENT-NUMBER PIC X(9).
   05 SR-STUDENT-LAST-NAME PIC X(14).
   05 SR-STUDENT-FIRST-NAME PIC X(10).
   05 FILLER PIC X(9).
   05 SR-GRADE-POINTS PIC 9(3).
   05 SR-UNITS-COMPLETED PIC 9(3).
   05 FILLER PIC X(26).
   05 SR-IP-CLASSIFICATIONS.
      10 SR-COLLEGE-YEAR-CODE PIC 9(1).
      10 SR-GPA PIC 9(1)V99.
HEADING-LINE-1.
FILLER PIC X(1) VALUE SPACE.
FILLER PIC X(20) VALUE "STUDENT LIST BY GPA ".
FILLER PIC X(20) VALUE "BY COLLEGE YEAR (11-4)".
FILLER PIC X(20) VALUE "PAGE ".
FILLER PIC ZZ9.
FILLER PIC X(52) VALUE SPACES.

HEADING-LINE-2.
FILLER PIC X(1) VALUE SPACE.
FILLER PIC X(20) VALUE "COLLEGE STUD".
FILLER PIC X(20) VALUE "ENT STUD".
FILLER PIC X(20) VALUE "ENT NAME G"
FILLER PIC X(20) VALUE "RADE UNITS ".
FILLER PIC X(20) VALUE "".
FILLER PIC X(20) VALUE "".
FILLER PIC X(12) VALUE "".

HEADING-LINE-3.
FILLER PIC X(1) VALUE SPACE.
FILLER PIC X(20) VALUE "YEAR NUMB".
FILLER PIC X(20) VALUE "LAST ".
FILLER PIC X(20) VALUE "FIRST ".
FILLER PIC X(20) VALUE "QINTS COMP. GPA ".
FILLER PIC X(20) VALUE "".
FILLER PIC X(20) VALUE "".
FILLER PIC X(20) VALUE "".
FILLER PIC X(12) VALUE "".
PROCEDURE DIVISION.

OPEN INPUT STUDENT-FILE
OUTPUT STUDENT-GPA-YEAR-LIST.
SORT SORT-FILE
DESCENDING KEY SX-COLLEGE-YEAR-CODE
DESCENDING KEY SX-GPA
ASCENDING KEY SX-STUDENT-LAST-NAME
ASCENDING KEY SX-STUDENT-FIRST-NAME
ASCENDING KEY SX-STUDENT-NUMBER
INPUT PROCEDURE IS 2000-IP-CLASSIFY-RECORDS
OUTPUT PROCEDURE IS 3000-OP-PROCESS-SORTED-RECORDS.
CLOSE STUDENT-FILE
STUDENT-GPA-YEAR-LIST.
STOP RUN.

2000-IP-CLASSIFY-RECORDS SECTION.

2000-CLASSIFY-RECORDS.

MOVE "NO " TO WS-END-OF-FILE-SWITCH.
PERFORM 2800-READ-STUDENT-RECORD.
PERFORM 2200-CLASSIFY-STUDENT-YEAR UNTIL END-OF-FILE.
GO TO 2999-EXIT.

2200-CLASSIFY-STUDENT-YEAR.

IF SR-UNITS-COMPLETED IS GREATER THAN 089
   MOVE 4 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS GREATER THAN 059
   AND
   SR-UNITS-COMPLETED IS LESS THAN 090
   MOVE 3 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS GREATER THAN 029
   AND
   SR-UNITS-COMPLETED IS LESS THAN 060
   MOVE 2 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS LESS THAN 030
   MOVE 1 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS NOT EQUAL TO ZERO
   DIVIDE SR-GRADE-POINTS BY SR-UNITS-COMPLETED
   GIVING SR-GPA ROUNDED
ELSE
   MOVE ZERO TO SR-GPA.
PERFORM 2850-RELEASE-STUDENT-RECORD.
PERFORM 2800-READ-STUDENT-RECORD.

2800-READ-STUDENT-RECORD.
READ STUDENT-FILE INTO SR-STUDENT-RECORD
AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.
214  *  
215  *  
216  2850-RELEASE-STUDENT-RECORD.  
217  *  
218  RELEASE SX-SORT-RECORD FROM SR-STUDENT-RECORD.  
219  *  
220  *  
221  2999-EXIT.  
222  *  
223  EXIT.  
224  *  
225  *  
226  3000-OP-PROCESS-SORTED-RECORDS SECTION.  
227  *  
228  *  
229  3000-PROCESS-SORTED-RECORDS.  
230  *  
231  PERFORM 3100-INITIALIZE-VAR-FIELDS.  
232  PERFORM 3800-RETURN-SORTED-RECORD.  
233  PERFORM 3200-PROCESS-SORTED-RECORD  
234  UNTIL END-OF-FILE.  
235  GO TO 3999-EXIT.  
236  *  
237  *  
238  3100-INITIALIZE-VAR-FIELDS.  
239  *  
240  MOVE "NO " TO WS-END-OF-FILE-SWITCH.  
241  MOVE ZERO TO WS-PAGE-COUNT.  
242  MOVE WS-LINES-PER-PAGE TO WS-LINES-USED.  
243  ACCEPT WS-DATE-WORK FROM DATE.  
244  MOVE WS-MONTH TO H1-MONTH.  
245  MOVE WS-DAY TO H1-DAY.  
246  MOVE WS-YEAR TO H1-YEAR.  
247  MOVE ZEROS TO WS-TOTAL-ACCUMULATORS.  
248  *  
249  *  
250  3200-PROCESS-SORTED-RECORD.  
251  *  
252  IF WS-LINES-USED IS NOT LESS THAN WS-LINES-PER-PAGE  
253  PERFORM 3870-PRINT-REPORT-HEADINGS.  
254  IF SR-COLLEGE-YEAR-CODE IS EQUAL TO 1  
255  MOVE "FRESHMAN " TO DL-COLLEGE-YEAR  
256  ADD 1 TO WS-TOTAL-FRESHMEN-ACCUM  
257  ELSE IF SR-COLLEGE-YEAR-CODE IS EQUAL TO 2  
258  MOVE "SOPHOMORE" TO DL-COLLEGE-YEAR  
259  ADD 1 TO WS-TOTAL-SOPHOMORES-ACCUM  
260  *  
261  ELSE IF SR-COLLEGE-YEAR-CODE IS EQUAL TO 3  
262  MOVE "JUNIOR " TO DL-COLLEGE-YEAR
ADD 1 TO WS-TOTAL-JUNIORS-ACCUM
ELSE
  MOVE "SENIOR " TO DL-COLLEGE-YEAR
  ADD 1 TO WS-TOTAL-SENIORS-ACCUM.
  MOVE SR-STUDENT-NUMBER TO DL-STUDENT-NUMBER.
  INSPECT DL-STUDENT-NUMBER
  REPLACING ALL SPACES BY "-"
  MOVE SR-STUDENT-LAST-NAME TO DL-STUDENT-LAST-NAME.
  MOVE SR-STUDENT-FIRST-NAME TO DL-STUDENT-FIRST-NAME.
  MOVE SR-GRADE-POINTS TO DL-GRADE-POINTS.
  MOVE SR-UNITS-COMPLETED TO DL-UNITS-COMPLETED.
  MOVE SR-GPA TO DL-GPA.
  MOVE DL-DETAIL-LINE TO REPORT-LINE.
  MOVE 1 TO WS-LINE-SPACING.
  PERFORM 3890-WRITE-REPORT-LINE.
  MOVE SPACES TO SR-IP-CLASSIFICATIONS.
  PERFORM 3800-RETURN-SORTED-RECORD.
* 3800-RETURN-SORTED-RECORD.
* RETURN SORT-FILE INTO SR-STUDENT-RECORD
** AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.
* 3870-PRINT-REPORT-HEADINGS.
* ADD 1 TO WS-PAGE-COUNT.
  MOVE WS-PAGE-COUNT TO H1-PAGE-NBR.
  MOVE H1-HEADING-LINE-1 TO REPORT-LINE.
  PERFORM 3880-WRITE-REPORT-TOP-LINE.
  MOVE H2-HEADING-LINE-2 TO REPORT-LINE.
  MOVE 2 TO WS-LINE-SPACING.
  PERFORM 3890-WRITE-REPORT-LINE.
  MOVE H3-HEADING-LINE-3 TO REPORT-LINE.
  MOVE 1 TO WS-LINE-SPACING.
  PERFORM 3890-WRITE-REPORT-LINE.
  MOVE SPACES TO REPORT-LINE.
  PERFORM 3890-WRITE-REPORT-LINE.
* 3880-WRITE-REPORT-TOP-LINE.
* WRITE REPORT-LINE AFTER ADVANCING PAGE.
* MOVE 1 TO WS-LINES-USED.
309  * 3890-WRITE-REPORT-LINE.
310 * WRITE REPORT-LINE
311  AFTER ADVANCING WS-LINE-SPACING LINES.
312 ADD WS-LINE-SPACING TO WS-LINES-USED.
313 *
314 *
315 3999-EXIT.
316 *
317 EXIT.
318 *
319 *
320 *
321 *
322 7000-PRINT-TOTAL-LINE SECTION.
323 *
324 *
325 7700-PRINT-TOTAL-LINE.
326 *
327 MOVE "FRESHMEN   " TO TL-COLLEGE-YEAR.
328 MOVE WS-TOTAL-FRESHMEN-ACCUM TO TL-TOTAL-
329 YEAR-STUDENTS.
330 MOVE TL-TOTAL-LINE TO REPORT-LINE.
331 WRITE REPORT-LINE
332  AFTER ADVANCING 3 LINES.
333 *
334 MOVE "SOPHOMORES " TO TL-COLLEGE-YEAR.
335 MOVE WS-TOTAL-SOPHOMORES-ACCUM TO TL-TOTAL-
336 YEAR-STUDENTS.
337 MOVE TL-TOTAL-LINE TO REPORT-LINE.
338 WRITE REPORT-LINE
339  AFTER ADVANCING 1 LINES.
340 *
341 MOVE "JUNIORS   " TO TL-COLLEGE-YEAR.
342 MOVE WS-TOTAL-JUNIORS-ACCUM TO TL-TOTAL-
343 YEAR-STUDENTS.
344 MOVE TL-TOTAL-LINE TO REPORT-LINE.
345 WRITE REPORT-LINE
346  AFTER ADVANCING 1 LINES.
347 *
348 MOVE "SENIORS   " TO TL-COLLEGE-YEAR.
349 MOVE WS-TOTAL-SENIORS-ACCUM TO TL-TOTAL-
350 YEAR-STUDENTS.
351 MOVE TL-TOTAL-LINE TO REPORT-LINE.
352 WRITE REPORT-LINE
353  AFTER ADVANCING 1 LINES.
LOGIC TASK - PART THREE

IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM-114.
AUTHOR. BRENDER.
DATE-WRITTEN. NOVEMBER 17, 1986.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT STUDENT-FILE
ASSIGN TO SYS008-UT-3340-S.
SELECT SORT-FILE
ASSIGN TO SYS001-UT-3340-S-SORTWK1.
SELECT STUDENT-GPA-YEAR-LIST
ASSIGN TO SYS005-UR-1403-S.

DATA DIVISION.
FILE SECTION.
FD STUDENT-FILE
RECORD CONTAINS 80 CHARACTERS
LABEL RECORDS ARE STANDARD.
01 STUDENT-RECORD.
   05 FILLER PIC X(80).
FD STUDENT-GPA-YEAR-LIST
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED.
01 REPORT-LINE.
   05 FILLER PIC X(133).
SD SORT-FILE
RECORD CONTAINS 80 CHARACTERS.
01 SX-SORT-RECORD.
   05 FILLER PIC X(2).
   05 SX-STUDENT-NUMBER PIC X(9).
   05 SX-STUDENT-LAST-NAME PIC X(14).
   05 SX-STUDENT-FIRST-NAME PIC X(10).
   05 FILLER PIC X(9).
50 05 SX-GRADE-POINTS PIC 9(3).
51 05 SX-UNITS-COMPLETED PIC 9(3).
52 05 FILLER PIC X(26).
53 05 SX-COLLEGE-YEAR-CODE PIC 9(1).
54 05 SX-GPA PIC 9(1)V99.
55 *
56 *
57 WORKING-STORAGE SECTION.
58 *
59 *
60 01 WS-SWITCHES.
61 05 WS-END-OF-FILE-SWITCH PIC X(3).
62 88 END-OF-FILE VALUE "YES".
63 *
64 01 WS-REPORT-CONTROLS.
65 05 WS-PAGE-COUNT PIC S9(3) COMP-3.
66 05 WS-LINES-PER-PAGE PIC S9(2) VALUE +57 COMP.
67 05 WS-LINES-USED PIC S9(2) COMP.
68 05 WS-LINE-SPACING PIC S9(2).
69 *
70 01 WS-WORK-AREAS.
71 05 WS-DATE-WORK PIC S9(6).
72 05 WS-DATE-REFORMAT REDEFINES WS-DATE-WORK.
73 10 WS-YEAR PIC 9(2).
74 10 WS-MONTH PIC 9(2).
75 10 WS-DAY PIC 9(2).
76 *
77 01 WS-TOTAL-ACCUMULATORS.
78 05 WS-TOTAL-FRESHMEN-ACCUM PIC S9(5).
79 05 WS-TOTAL-SOPHOMORES-ACCUM PIC S9(5).
80 05 WS-TOTAL-JUNIORS-ACCUM PIC S9(5).
81 05 WS-TOTAL-SENIORS-ACCUM PIC S9(5).
82 *
83 01 SR-STUDENT-RECORD.
84 05 SR-RECORD-CODE PIC 9(2).
85 05 SR-STUDENT-NUMBER PIC X(9).
86 05 SR-STUDENT-LAST-NAME PIC X(14).
87 05 SR-STUDENT-FIRST-NAME PIC X(10).
88 05 FILLER PIC X(9).
89 05 SR-GRADE-POINTS PIC 9(3).
90 05 SR-UNITS-COMPLETED PIC 9(3).
91 05 FILLER PIC X(26).
92 05 SR-IP-CLASSIFICATIONS.
93 10 SR-COLLEGE-YEAR-CODE PIC 9(1).
94 10 SR-GPA PIC 9(1)V99.
95 *
01 H1-HEADING-LINE-1.
   05 FILLER PIC X(1) VALUE SPACE.
   05 FILLER PIC X(20) VALUE
   "STUDENT LIST BY GPA ".
   05 FILLER PIC X(20) VALUE
   "BY COLLEGE YEAR (11-".
   05 FILLER PIC X(20) VALUE
   "4)
   05 FILLER PIC X(1) VALUE " ".
   05 H1-MONTH PIC 9(2) VALUE
   05 FILLER PIC X(1) VALUE " ".
   05 H1-DAY PIC 9(2) VALUE
   05 FILLER PIC X(1) VALUE " ".
   05 H1-YEAR PIC 9(2) VALUE
   " PAGE ".
   05 H1-PAGE-NBR PIC ZZ9.
   05 FILLER PIC X(52) VALUE SPACES.

* 01 H2-HEADING-LINE-2.
   05 FILLER PIC X(1) VALUE SPACE.
   05 FILLER PIC X(20) VALUE
   " COLLEGE STUD".
   05 FILLER PIC X(20) VALUE
   "ENT "
   05 FILLER PIC X(20) VALUE
   "ENT NAME---- "
   05 FILLER PIC X(20) VALUE
   "RADE UNITS ".
   05 FILLER PIC X(20) VALUE
   " ".
   05 FILLER PIC X(20) VALUE
   " ".
   05 FILLER PIC X(20) VALUE
   " ".
   05 FILLER PIC X(12) VALUE

* 01 H3-HEADING-LINE-3.
   05 FILLER PIC X(1) VALUE SPACE.
   05 FILLER PIC X(20) VALUE
   " YEAR NUMB".
   05 FILLER PIC X(20) VALUE
   "ER LAST ".
   05 FILLER PIC X(20) VALUE
   "FIRST P".
   05 FILLER PIC X(20) VALUE
   "QINTS COMP. GPA ".
   05 FILLER PIC X(20) VALUE
   ".
   05 FILLER PIC X(20) VALUE
   ".
   05 FILLER PIC X(12) VALUE
PROCEDURE DIVISION.

0000-ML-SORT-RECORDS SECTION.

OPEN INPUT STUDENT-FILE
OUTPUT STUDENT-GPA-YEAR-LIST.
SORT-SORT-FILE
  DESCENDING KEY SX-COLLEGE-YEAR-CODE
  DESCENDING KEY SX-GPA
  ASCENDING KEY SX-STUDENT-LAST-NAME
ASCENDING KEY SX-STUDENT-FIRST-NAME
ASCENDING KEY SX-STUDENT-NUMBER
INPUT PROCEDURE IS 2000-IP-CLASSIFY-RECORDS
OUTPUT PROCEDURE IS 3000-OP-PROCESS-SORTED-RECORDS.
CLOSE STUDENT-FILE
STUDENT-GPA-YEAR-LIST.
STOP RUN.

2000-IP-CLASSIFY-RECORDS SECTION.

2000-CLASSIFY-RECORDS.

MOVE "NO " TO WS-END-OF-FILE-SWITCH.
PERFORM 2800-READ-STUDENT-RECORD.
PERFORM 2200-CLASSIFY-STUDENT-YEAR UNTIL END-OF-FILE.
GO TO 2999-EXIT.

2200-CLASSIFY-STUDENT-YEAR.

IF SR-UNITS-COMPLETED IS GREATER THAN 089
MOVE 4 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS GREATER THAN 059
AND
SR-UNITS-COMPLETED IS LESS THAN 090
MOVE 3 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS GREATER THAN 029
AND
SR-UNITS-COMPLETED IS LESS THAN 060
MOVE 2 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS LESS THAN 030
MOVE 1 TO SR-COLLEGE-YEAR-CODE.
IF SR-UNITS-COMPLETED IS NOT EQUAL TO ZERO
DIVIDE SR-GRADE-POINTS BY SR-UNITS-COMPLETED
GIVING SR-GPA ROUNDED
ELSE
MOVE ZERO TO SR-GPA.
PERFORM 2850-RELEASE-STUDENT-RECORD.
PERFORM 2800-READ-STUDENT-RECORD.

2800-READ-STUDENT-RECORD.

READ STUDENT-FILE INTO SR-STUDENT-RECORD
AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.
214  * 
215  * 
216  ** 2850-RELEASE-STUDENT-RECORD. 
217  * 
218  ** RELEASE SX-SORT-RECORD FROM SR-STUDENT-RECORD. 
219  * 
220  * 
221  ** 2999-EXIT. 
222  * 
223  * EXIT. 
224  * 
225  * 
226  ** 3000-OP-PROCESS-SORTED-RECORDS SECTION. 
227  * 
228  * 
229  ** 3000-PROCESS-SORTED-RECORDS. 
230  * 
231  ** PERFORM 3100-INITIALIZE-VAR-FIELDS. 
232  ** PERFORM 3800-RETURN-SORTED-RECORD. 
233  ** PERFORM 3200-PROCESS-SORTED-RECORD UNTIL END-OF-FILE. 
234  ** GO TO 3999-EXIT. 
235  * 
236  * 
237  * 
238  ** 3100-INITIALIZE-VAR-FIELDS. 
239  * 
240  ** MOVE "NO " TO WS-END-OF-FILE-SWITCH. 
241  ** MOVE ZERO TO WS-PAGE-COUNT. 
242  ** MOVE WS-LINES-PER-PAGE TO WS-LINES-USED. 
243  ** ACCEPT WS-DATE-WORK FROM DATE. 
244  ** MOVE WS-MONTH TO H1-MONTH. 
245  ** MOVE WS-DAY TO H1-DAY. 
246  ** MOVE WS-YEAR TO H1-YEAR. 
247  ** MOVE ZEROS TO WS-TOTAL-ACCUMULATORS. 
248  * 
249  * 
250  ** 3200-PROCESS-SORTED-RECORD. 
251  * 
252  ** IF WS-LINES-USED IS NOT LESS THAN WS-LINES-PER-PAGE 
253  ** PERFORM 3870-PRINT-REPORT-HEADINGS. 
254  ** IF SR-COLLEGE-YEAR-CODE IS EQUAL TO 1 
255  ** MOVE "FRESHMAN " TO DL-COLLEGE-YEAR 
256  ** ADD 1 TO WS-TOTAL-FRESHMEN-ACCUM 
257  ** IF SR-COLLEGE-YEAR-CODE IS EQUAL TO 2 
258  ** MOVE "SOPHOMORE" TO DL-COLLEGE-YEAR 
259  ** ADD 1 TO WS-TOTAL-SOPHOMORES-ACCUM 
260  ** IF SR-COLLEGE-YEAR-CODE IS EQUAL TO 3 
261  ** MOVE "JUNIOR " TO DL-COLLEGE-YEAR 
262  ** ADD 1 TO WS-TOTAL-JUNIORS-ACCUM
ELSE
    MOVE "SENIOR " TO DL-COLLEGE-YEAR
    ADD 1 TO WS-TOTAL-SENIORS-ACCUM.
    MOVE SR-STUDENT-NUMBER TO DL-STUDENT-NUMBER.
    INSPECT DL-STUDENT-NUMBER
    REPLACING ALL SPACES BY "."
    MOVE SR-STUDENT-LAST-NAME TO DL-STUDENT-LAST-NAME.
    MOVE SR-STUDENT-FIRST-NAME TO DL-STUDENT-FIRST-NAME.
    MOVE SR-GRADE-POINTS TO DL-GRADE-POINTS.
    MOVE SR-UNITS-COMPLETED TO DL-UNITS-COMPLETED.
    MOVE SR-GPA TO DL-GPA.
    MOVE DL-DETAIL-LINE TO REPORT-LINE.
    MOVE 1 TO WS-LINE-SPACING.
    PERFORM 3890-WRITE-REPORT-LINE.
    MOVE SPACES TO SR-IP-CLASSIFICATIONS.
    PERFORM 3800-RETURN-SORTED-RECORD.
    PERFORM 3800-RETURN-SORTED-RECORD.
    RETURN SORT-FILE INTO SR-STUDENT-RECORD
    AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.

    MOVE WS-PAGE-COUNT TO H1-PAGE-NBR.
    MOVE H1-HEADING-LINE-1 TO REPORT-LINE.
    PERFORM 3880-WRITE-REPORT-TOP-LINE.
    MOVE H2-HEADING-LINE-2 TO REPORT-LINE.
    MOVE 2 TO WS-LINE-SPACING.
    PERFORM 3890-WRITE-REPORT-LINE.
    MOVE H3-HEADING-LINE-3 TO REPORT-LINE.
    MOVE 1 TO WS-LINE-SPACING.
    PERFORM 3890-WRITE-REPORT-LINE.
    MOVE SPACES TO REPORT-LINE.
    PERFORM 3890-WRITE-REPORT-LINE.
    ADD 1 TO WS-LINES-USED.
    WRITE REPORT-LINE AFTER ADVANCING PAGE.
    MOVE 1 TO WS-LINES-USED.
309 3890-WRITE-REPORT-LINE.
310 *
311 WRITE REPORT-LINE
312  AFTER ADVANCING WS-LINE-SPACING LINES.
313 ADD WS-LINE-SPACING TO WS-LINES-USED.
314 *
315 *
316 3999-EXIT.
317 *
318  EXIT.
319 *
320 *
321 7000-PRINT-TOTAL-LINE-SECTION.
322 *
323 *
324 7700-PRINT-TOTAL-LINE.
325 *
326  MOVE "FRESHMEN " TO TL-COLLEGE-YEAR.
327  MOVE WS-TOTAL-FRESHMEN-ACCUM TO TL-TOTAL-
328  YEAR-STUDENTS.
329  MOVE TL-TOTAL-LINE TO REPORT-LINE.
330  WRITE REPORT-LINE
331  AFTER ADVANCING 3 LINES.
332 *
333  MOVE "SOPHOMORES " TO TL-COLLEGE-YEAR.
334  MOVE WS-TOTAL-SOPHOMORES-ACCUM TO TL-TOTAL-
335  YEAR-STUDENTS.
336  MOVE TL-TOTAL-LINE TO REPORT-LINE.
337  WRITE REPORT-LINE
338  AFTER ADVANCING 1 LINES.
339 *
340  MOVE "JUNIORS " TO TL-COLLEGE-YEAR.
341  MOVE WS-TOTAL-JUNIORS-ACCUM TO TL-TOTAL-
342  YEAR-STUDENTS.
343  MOVE TL-TOTAL-LINE TO REPORT-LINE.
344  WRITE REPORT-LINE
345  AFTER ADVANCING 1 LINES.
346 *
347  MOVE "SENIORS " TO TL-COLLEGE-YEAR.
348  MOVE WS-TOTAL-SENIORS-ACCUM TO TL-TOTAL-
349  YEAR-STUDENTS.
350  MOVE TL-TOTAL-LINE TO REPORT-LINE.
351  WRITE REPORT-LINE
352  AFTER ADVANCING 1 LINES.
LOGIC TASK - PART FOUR

IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM-133.
AUTHOR. BRENDER.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT EARNINGS-FILE
   ASSIGN TO SYS008-UT-3340-S.
SELECT FEDERAL-INCOME-TAX-REGISTER
   ASSIGN TO SYS005-UR-1403-S.

DATA DIVISION.
FILE SECTION.
FD EARNINGS-FILE
   RECORD CONTAINS 80 CHARACTERS
   LABEL RECORDS ARE STANDARD.
   01 STUDENT-RECORD.
      05 FILLER PIC X(80).
FD FEDERAL-INCOME-TAX-REGISTER
   RECORD CONTAINS 133 CHARACTERS
   LABEL RECORDS ARE OMITTED.
   01 REPORT-LINE.
      05 FILLER PIC X(133).

WORKING-STORAGE SECTION.
01 WS-SWITCHES.
   05 WS-END-OF-FILE-SWITCH PIC X(3).
   88 END-OF-FILE VALUE "YES".
01 WS-REPORT-CONTROLS.
180

48  05 WS-PAGE-COUNT PIC S9(4) COMP-3.
49  05 WS-LINES-PER-PAGE PIC S9(2)
     VALUE +57
50  05 WS-LINES-USED PIC S9(2) COMP.
51  05 WS-LINE-SPACING PIC S9(2).
52  *
53  01 WS-WORK AREAS.
54  05 WS-DATE WORK PIC 9(6).
55  05 WS-DATE-REFORMAT REDEFINES WS-DATE-WORK.
56  10 WS-YEAR PIC 9(2).
57  10 WS-MONTH PIC 9(2).
58  10 WS-DAY PIC 9(2).
60  05 WS-STANDARD-DEDUCTION PIC S9(6)V99.
61  05 WS-TAX-AMOUNT PIC S9(5)V99.
62  *
63  01 ER-EARNINGS-RECORD.
64  05 ER-RECORD-CODE PIC X(2).
65  05 ER-PLANT-CODE PIC X(3).
66  05 ER-DEPARTMENT-NUMBER PIC X(4).
67  05 ER-EMPLOYEE-NUMBER PIC X(9).
68  05 ER-LAST-NAME PIC X(12).
69  05 ER-FIRST-NAME PIC X(9).
70  05 ER-MIDDLE-NAME PIC X(8).
71  05 ER-SEX-CODE PIC X(1).
72  05 ER-MARITAL-STATUS PIC X(1).
73  05 ER-NUMBER-OF-EXEMPTIONS PIC 9(2).
74  05 ER-PAY-RATE PIC X(1).
75  10 ER-PAY-RATE-HOURLY PIC 9(2)V9999.
76  10 ER-PAY-RATE-SALARIED REDEFINES ER-PAY-RATE-HOURLY PIC 9(4)V99.
77  05 ER-THIS-PERIOD-EARNINGS PIC S9(5)V99.
78  05 ER-YTD-EARNINGS PIC 9(6)V99.
79  05 ER-DATE-LAST-ACTIVITY PIC X(6).
80  *
81  01 H1-HEADING-LINE-1.
82  05 FILLER PIC X(1) VALUE SPACE.
83  05 FILLER PIC X(20) VALUE "FEDERAL INCOME TAX R".
84  05 FILLER PIC X(20) VALUE "REGISTER (13-3) ".
05 FILLER PIC X(20) VALUE
05 FILLER PIC X(11) VALUE
05 H1-MONTH PIC 9(2). VALUE "/".
05 FILLER PIC X(1) VALUE "/".
05 H1-YEAR PIC 9(2). VALUE
" PAGE ".
05 H1-PAGE-NBR PIC ZZZ9.
05 FILLER PIC X(42) VALUE SPACES.
01 H2-HEADING-LINE-2.
05 FILLER PIC X(1) VALUE SPACE.
05 FILLER PIC X(20) VALUE
" EMPLOYEE ------EM".
05 FILLER PIC X(20) VALUE
"LOYEE- NAME---- M ".
05 FILLER PIC X(20) VALUE
" NO. THIS PER. ANN".
05 FILLER PIC X(20) VALUE
"UALIZED TAXABLE ".
05 FILLER PIC X(20) VALUE
" TAX ".
05 FILLER PIC X(20) VALUE
" ".
05 FILLER PIC X(12) VALUE
01 H3-HEADING-LINE-3.
05 FILLER PIC X(1) VALUE SPACE.
05 FILLER PIC X(20) VALUE
" NUMBER LAST ".
05 FILLER PIC X(20) VALUE
" FIRST S ".
05 FILLER PIC X(20) VALUE
" EX. EARNINGS E".
05 FILLER PIC X(20) VALUE
"ARNINGS EARNINGS ".
05 FILLER PIC X(20) VALUE
" AMOUNT ".
05 FILLER PIC X(12) VALUE
01 DL-DETAIL-LINE.
05 FILLER VALUE SPACE.
PIC X(1)
05 DL-EMPLOYEE-NUMBER PIC XXXBXXBXXXXX.
THE COLUMNS BELOW CONTAIN THE FOLLOWING TAX DATA FIELDS:

- HHHHHH = HIGH-LIMIT DOLLAR AMOUNT (NO CENTS)
- BBBB BBB = BASE TAX DOLLAR AMOUNT (Dollars and Cents)
- PP = TAX PERCENTAGE FOR INCOME ABOVE LOW-LIMIT
- LLLLLL = LOW-LIMIT INCOME AMOUNT (NO CENTS)
* CENTS)

* HHHHHBBBBBBBPPLLLLL
05 FILLER PIC X(20) VALUE "001420000000000000".
05 FILLER PIC X(20) VALUE "0043700000001201420".
05 FILLER PIC X(20) VALUE "00960000354001504370".
05 FILLER PIC X(20) VALUE "01520001138501909600".
05 FILLER PIC X(20) VALUE "02290002202502515200".
05 FILLER PIC X(20) VALUE "02893004127503022900".
05 FILLER PIC X(20) VALUE "03435005936503428930".
05 FILLER PIC X(20) VALUE "09999997813303734350".
05 FILLER PIC X(20) VALUE "00250000000000000000".
05 FILLER PIC X(20) VALUE "0100000000001202500".
05 FILLER PIC X(20) VALUE "00199500900001710000".
05 FILLER PIC X(20) VALUE "02456002591502219950".
05 FILLER PIC X(20) VALUE "03008003605702524560".
05 FILLER PIC X(20) VALUE "03559004985702830080".
05 FILLER PIC X(20) VALUE "04662006528503335590".
05 FILLER PIC X(20) VALUE "999999910168403746620".

01 TT-TAX-TABLE REDEFINES TT-TAX-DATA.
05 TT-TAX-ENTRY OCCURS 16 TIMES

INDEXED BY

TT-INDEX.
10 TT-HIGH-INCOME-LIMIT PIC 9(6).
10 TT-TAX-PERCENTAGE PIC V99.
10 TT-BASE-INCOME-AMOUNT PIC 9(5).
PROCEDURE DIVISION.

000-PRINT-REGION-TERR-LIST.

OPEN INPUT EARNINGS-FILE
OUTPUT FEDERAL-INCOME-TAX-REGISTER.
PERFORM 100-INITIALIZE-VARIABLE-FIELDS.
PERFORM 800-READ-EARNINGS-RECORD.
PERFORM 200-PROCESS-EARNINGS-RECORD
UNTIL WS-END-OF-FILE-SWITCH IS EQUAL TO "YES".
CLOSE EARNINGS-FILE
FEDERAL-INCOME-TAX-REGISTER.
STOP RUN.

100-INITIALIZE-VARIABLE-FIELDS.

MOVE "NO " TO WS-END-OF-FILE-SWITCH.
MOVE ZERO TO WS-PAGE-COUNT.
MOVE WS-LINES-PER-PAGE TO WS-LINES-USED.
ACCEPT WS-DATE-WORK FROM DATE.
MOVE WS-MONTH TO H1-MONTH.
MOVE WS-DAY TO H1-DAY.
MOVE WS-YEAR TO H1-YEAR.

200-PROCESS-EARNINGS-RECORD.

IF WS-LINES-USED IS NOT LESS THAN WS-LINES-PER-PAGE
PERFORM 870-PRINT-REPORT-HEADINGS.
MOVE ER-EMPLOYEE-NUMBER TO DL-EMPLOYEE-NUMBER.
INSPECT DL-EMPLOYEE-NUMBER REPLACING ALL SPACES BY "-".
MOVE ER-LAST-NAME TO DL-LAST-NAME.
MOVE ER-FIRST-NAME TO DL-FIRST-NAME.
MOVE ER-MARITAL-STATUS TO DL-MARITAL-STATUS.
MOVE ER-NUMBER-OF-EXEMPTIONS TO DL-NUMBER-OF-EXEMPTIONS.
MOVE ER-THIS-PERIOD-EARNINGS TO DL-THIS-PERIOD-EARNINGS.
IF ER-THIS-PERIOD-EARNINGS IS NOT NEGATIVE
MULTIPLY ER-THIS-PERIOD-EARNINGS BY 12
GIVING WS-ANNUALIZED-TAXABLE-EARNINGS

ELSE
MOVE ZERO TO WS-ANNUALIZED-TAXABLE-EARNINGS.

MOVE WS-ANNUALIZED-TAXABLE-EARNINGS TO DL-ANNUALIZED-EARNINGS.
MULTIPLY ER-NUMBER-OF-EXEMPTIONS BY 1000 GIVING WS-STANDARD-DEDUCTION.

SUBTRACT WS-STANDARD-DEDUCTION FROM WS-ANNUALIZED-TAXABLE-EARNINGS.
IF WS-ANNUALIZED-TAXABLE-EARNINGS IS NEGATIVE
MOVE ZERO TO WS-ANNUALIZED-TAXABLE-EARNINGS.

MOVE WS-ANNUALIZED-TAXABLE-EARNINGS TO DL-TAXABLE-EARNINGS.

PERFORM 210-CALCULATE-TAX-AMOUNT.
MOVE WS-TAX-AMOUNT TO DL-TAX-AMOUNT.
MOVE DL-DETAIL-LINE TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
PERFORM 800-READ-EARNINGS-RECORD.

* 

PERFORM 220-LOOKUP-TAX-RANGE.
SUBTRACT TT-BASE-INCOME-AMOUNT (TT-INDEX) FROM WS-ANNUALIZED-TAXABLE-EARNINGS GIVING WS-TAX-AMOUNT.
MULTIPLY TT-TAX-PERCENTAGE (TT-INDEX) BY WS-TAX-AMOUNT ROUND.
ADD TT-BASE-TAX-DOLLAR-AMOUNT (TT-INDEX) TO WS-TAX-AMOUNT.
DIVIDE 12 INTO WS-TAX-AMOUNT ROUND.

* 

IF ER-MARITAL-STATUS IS EQUAL TO "M"
SET TT-INDEX TO 9
ELSE
SET TT-INDEX TO 9.

* 
SEARCH TT-TAX-ENTRY
WHEN WS-ANNUALIZED-TAXABLE-EARNINGS IS NOT GREATER THAN TT-HIGH-INCOME-LIMIT (TT-INDEX)
MOVE "YES" TO TT-ENTRY-FOUND-SWITCH.
800-READ-EARNINGS-RECORD.
READ EARNINGS-FILE INTO ER-EARNINGS-RECORD
AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.

870-PRINT-REPORT-HEADINGS.
ADD 1 TO WS-PAGE-COUNT.
MOVE WS-PAGE-COUNT TO H1-PAGE-NBR.
MOVE H1-HEADING-LINE-1 TO REPORT-LINE.
PERFORM 880-WRITE-REPORT-TOPLINE.
MOVE H2-HEADING-LINE-2 TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
MOVE H3-HEADING-LINE-3 TO REPORT-LINE.
MOVE 1 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.

880-WRITE-REPORT-TOPLINE.
WRITE REPORT-LINE
AFTER ADVANCING PAGE.
MOVE 1 TO WS-LINES-USED.

890-WRITE-REPORT-LINE.
WRITE REPORT-LINE
AFTER ADVANCING WS-LINE-SPACING LINES.
ADD WS-LINE-SPACING TO WS-LINES-USED.
LOGIC TASK - PART FIVE

IDENTIFICATION DIVISION.
PROGRAM-ID. PROGRAM-133.
AUTHOR. BRENDER.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-4341.
OBJECT-COMPUTER. IBM-4341.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
SELECT EARNINGS-FILE
   ASSIGN TO SYS008-UT-3340-S.
SELECT FEDERAL-INCOME-TAX-REGISTER
   ASSIGN TO SYS005-UR-1403-S.

DATA DIVISION.
FILE SECTION.
FD EARNINGS-FILE
   RECORD CONTAINS 80 CHARACTERS
   LABEL RECORDS ARE STANDARD.

01 STUDENT-RECORD.
   05 FILLER PIC X(80).

FD FEDERAL-INCOME-TAX-REGISTER
   RECORD CONTAINS 133 CHARACTERS
   LABEL RECORDS ARE OMITTED.

01 REPORT-LINE.
   05 FILLER PIC X(133).

WORKING-Storage SECTION.

01 WS-SWITCHES.
   05 WS-END-OF-FILE-SWITCH PIC X(3).
   88 END-OF-FILE
   VALUE "YES".

01 WS-REPORT-CONTROLS.
05 WS-PAGE-COUNT  PIC S9(4)  COMP-3.
05 WS-LINES-PER-PAGE  PIC S9(2)  VALUE +57
05 WS-LINES-USED  PIC S9(2)  COMP.
05 WS-LINE-SPACING  PIC S9(2).

01 WS-WORK AREAS.
  05 WS-DAY  PIC 9(2).
  05 WS-MONTH  PIC 9(2).
  05 WS-YEAR  PIC 9(2).

05 WS-DATE-WORK  PIC 9(6).
05 WS-DAY  PIC 9(2).
05 WS-MONTH  PIC 9(2).
05 WS-YEAR  PIC 9(2).

05 WS-DATE-REFORMAT REDEFINES WS-DATE-WORK.

05 WS-ANNUALIZED-TAXABLE-EARNINGS  PIC S9(6)V99.
05 WS-STANDARD-DEDUCTION  PIC S9(6)V99.
05 WS-TAX-AMOUNT  PIC S9(5)V99.

01 ER-EARNINGS-RECORD.
  05 ER-RECORD-CODE  PIC X(2).
  05 ER-PLANT-CODE  PIC X(3).
  05 ER-DEPARTMENT-NUMBER  PIC X(4).
  05 ER-EMPLOYEE-NUMBER  PIC X(9).
  05 ER-LAST-NAME  PIC X(12).
  05 ER-FIRST-NAME  PIC X(9).
  05 ER-MIDDLE-NAME  PIC X(8).
  05 ER-SEX-CODE  PIC X(1).
  05 ER-MARITAL-STATUS  PIC X(1).
  05 ER-NUMBER-OF-EXEMPTIONS  PIC 9(2).
  05 ER-PAY-CODE  PIC X(1).
  10 ER-PAY-RATE-HOURLY  PIC 9(2)V9999.

05 ER-THIS-PERIOD-EARNINGS  PIC S9(5)V99.
05 ER-YTD-EARNINGS  PIC 9(6)V99.
05 ER-DATE-LAST-ACTIVITY  PIC X(6).

01 H1-HEADING-LINE-1.
  05 FILLER  PIC X(1)  VALUE SPACE.
  05 FILLER  PIC X(20)  VALUE "FEDERAL INCOME TAX R".
  05 FILLER  PIC X(20)  VALUE "REGISTER (13-3) ".
89 05 FILLER PIC X(20) VALUE
  " ".
90 05 FILLER PIC X(11) VALUE
  " ".
91 05 H1-MONTH PIC 9(2).
92 05 FILLER PIC X(1) VALUE "/".
93 05 H1-DAY PIC 9(2).
94 05 FILLER PIC X(1) VALUE "/".
95 05 H1-YEAR PIC 9(2).
96 05 FILLER PIC X(7) VALUE
  " PAGE ".
97 05 H1-PAGE-NBR PIC ZZZ9.
98 05 FILLER PIC X(42) VALUE SPACES.
99  * 
100 01 H2-HEADING-LINE-2.
101 05 FILLER PIC X(1) VALUE SPACE.
102  05 FILLER PIC X(20) VALUE
  " EMPLOYEE -------EM".
103 05 FILLER PIC X(20) VALUE
  "PLOYEE- NAME------ M ".
104 05 FILLER PIC X(20) VALUE
  " NO. THIS PER. ANN".
105 05 FILLER PIC X(20) VALUE
  "UALIZED TAXABLE ".
106 05 FILLER PIC X(20) VALUE
  " TAX ".
107 05 FILLER PIC X(20) VALUE
  " ".
108  05 FILLER PIC X(12) VALUE
  " ".
109  * 
110 01 H3-HEADING-LINE-3.
111 05 FILLER PIC X(1) VALUE SPACE.
112  05 FILLER PIC X(20) VALUE
  " NUMBER LAST ".
113 05 FILLER PIC X(20) VALUE
  " FIRST S ".
114 05 FILLER PIC X(20) VALUE
  " EX. EARNINGS E".
115 05 FILLER PIC X(20) VALUE
  "ARNINGS EARNINGS ".
116 05 FILLER PIC X(20) VALUE
  " AMOUNT ".
117  05 FILLER PIC X(12) VALUE
  " ".
118  * 
119 01 DL-DETAIL-LINE.
120  05 FILLER VALUE SPACE.
121  05 DL-EMPLOYEE-NUMBER PIC XXXBXXBXXXX.
THE COLUMNS BELOW CONTAIN THE FOLLOWING TAX DATA FIELDS:

**HHHHHH** = HIGH-LIMIT DOLLAR AMOUNT (NO CENTS)

**BBBBBB** = BASE TAX DOLLAR AMOUNT (DOLLARS AND CENTS)

**PP** = TAX PERCENTAGE FOR INCOME ABOVE LOW-LIMIT

**LLLLLL** = LOW-LIMIT INCOME AMOUNT (NO
154  *
155  ********************************************
156  *
157  HHHHHHBBBBBBPPLLLLLL
158  05 FILLER PIC X(20) VALUE
159  "0014200000000000000000"
160  05 FILLER PIC X(20) VALUE
161  "00437000000001201420"
162  05 FILLER PIC X(20) VALUE
163  "00960000354001504370"
164  05 FILLER PIC X(20) VALUE
165  "01520001138501909600"
166  05 FILLER PIC X(20) VALUE
167  "02290002202502515200"
168  05 FILLER PIC X(20) VALUE
169  "02893004127503022900"
170  05 FILLER PIC X(20) VALUE
171  "03435005936503428930"
172  05 FILLER PIC X(20) VALUE
173  "09999997813303734350"
174  05 FILLER PIC X(20) VALUE
175  "00250000000000000000"
176  05 FILLER PIC X(20) VALUE
177  "0100000000001202500"
178  05 FILLER PIC X(20) VALUE
179  "019950090000171000"
180  05 FILLER PIC X(20) VALUE
181  "02456002591502219950"
182  05 FILLER PIC X(20) VALUE
183  "03008003605702524560"
184  05 FILLER PIC X(20) VALUE
185  "03559004985702830080"
186  05 FILLER PIC X(20) VALUE
187  "04662006528503335590"
188  05 FILLER PIC X(20) VALUE
189  "999999910168403746620"
190  *
191  01 TT-TAX-TABLE REDEFINES TT-TAX-DATA.
192  05 TT-TAX-ENTRY OCCURS 16 TIMES
193  INDEXED BY
194  TT-INDEX.
195  10 TT-HIGH-INCOME-LIMIT PIC 9(6).
197  9(5) V99.
198  10 TT-TAX-PERCENTAGE PIC V99.
199  10 TT-BASE-INCOME-AMOUNT PIC 9(5).
PROCEDURE DIVISION.

OPEN INPUT EARNINGS-FILE
OUTPUT FEDERAL-INCOME-TAX-REGISTER.
PERFORM 100-INITIALIZE-VARIABLE-FIELDS.
PERFORM 800-READ-EARNINGS-RECORD.
PERFORM 200-PROCESS-EARNINGS-RECORD
UNTIL WS-END-OF-FILE-SWITCH IS EQUAL TO "YES".
CLOSE EARNINGS-FILE
FEDERAL-INCOME-TAX-REGISTER.
STOP RUN.

100-INITIALIZE-VARIABLE-FIELDS.

MOVE "NO " TO WS-END-OF-FILE-SWITCH.
MOVE ZERO TO WS-PAGE-COUNT.
MOVE WS-LINES-PER-PAGE TO WS-LINES-USED.
ACCEPT WS-DATE-WORK FROM DATE.
MOVE WS-MONTH TO H1-MONTH.
MOVE WS-DAY TO H1-DAY.
MOVE WS-YEAR TO H1-YEAR.

200-PROCESS-EARNINGS-RECORD.

IF WS-LINES-USED IS NOT LESS THAN WS-LINES-
PER-PAGE
    PERFORM 870-PRINT-REPORT-HEADINGS.
    MOVE ER-EMPLOYEE-NUMBER TO DL-EMPLOYEE-
    NUMBER.
    INSPECT DL-EMPLOYEE-NUMBER
    REPLACING ALL SPACES BY "-".
    MOVE ER-LAST-NAME TO DL-LAST-NAME.
    MOVE ER-FIRST-NAME TO DL-FIRST-NAME.
    MOVE ER-MARITAL-STATUS TO DL-MARITAL-
    STATUS.
    MOVE ER-NUMBER-OF-EXEMPTIONS TO DL-NUMBER-
    OF-EXEMPTIONS.
    MOVE ER-THIS-PERIOD-EARNINGS TO DL-THIS-
    PERIOD-EARNINGS.
    MULTIPLY ER-THIS-PERIOD-EARNINGS BY 12
    GIVING WS-ANNUALIZED-TAXABLE-EARNINGS.
    MOVE WS-ANNUALIZED-TAXABLE-EARNINGS
MULTIPLY ER-NUMBER-OF-EXEMPTIONS BY 10000
GIVING WS-STANDARD-DEDUCTION.
SUBTRACT WS-STANDARD-DEDUCTION
FROM WS-ANNUALIZED-TAXABLE-EARNINGS.
MOVE WS-ANNUALIZED-TAXABLE-EARNINGS TO
DL-TAXABLE-EARNINGS.
PERFORM 210-CALCULATE-TAX-AMOUNT.
MOVE WS-TAX-AMOUNT TO DL-TAX-AMOUNT.
MOVE DL-DETAIL-LINE TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
PERFORM 800-READ-EARNINGS-RECORD.

210-CALCULATE-TAX-AMOUNT.

PERFORM 220-LOOKUP-TAX-RANGE.
SUBTRACT TT-BASE-INCOME-AMOUNT (TT-INDEX)
FROM WS-ANNUALIZED-TAXABLE-EARNINGS
GIVING WS-TAX-AMOUNT.
MULTIPLY TT-TAX-PERCENTAGE (TT-INDEX)
BY WS-TAX-AMOUNT ROUNDED.
ADD TT-BASE-TAX-DOLLAR-AMOUNT (TT-INDEX) TO
WS-TAX-AMOUNT.
DIVIDE 12 INTO WS-TAX-AMOUNT ROUNDED.

220-LOOKUP-TAX-RANGE.

IF ER-MARITAL-STATUS IS EQUAL TO "M"
SET TT-INDEX TO 9
ELSE
SET TT-INDEX TO 1.

SEARCH TT-TAX-ENTRY
WHEN WS-ANNUALIZED-TAXABLE-EARNINGS
IS NOT GREATER THAN TT-HIGH-INCOME-
LIMIT (TT-INDEX)
MOVE "YES" TO TT-ENTRY-FOUND-SWITCH.

800-READ-EARNINGS-RECORD.

READ EARNINGS-FILE INTO ER-EARNINGS-RECORD
AT END MOVE "YES" TO WS-END-OF-FILE-SWITCH.

870-PRINT-REPORT-HEADINGS.
ADD 1 TO WS-PAGE-COUNT.
MOVE WS-PAGE-COUNT TO H1-PAGE-NBR.
MOVE H1-HEADING-LINE-1 TO REPORT-LINE.
PERFORM 880-WRITE-REPORT-TOP-LINE.
MOVE H2-HEADING-LINE-2 TO REPORT-LINE.
MOVE 2 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.
MOVE H3-HEADING-LINE-3 TO REPORT-LINE.
MOVE 1 TO WS-LINE-SPACING.
PERFORM 890-WRITE-REPORT-LINE.

880-WRITE-REPORT-TOP-LINE.

WRITE REPORT-LINE
AFTER ADVANCING PAGE.
MOVE 1 TO WS-LINES-USED.

890-WRITE-REPORT-LINE.

WRITE REPORT-LINE
AFTER ADVANCING WS-LINE-SPACING LINES.
ADD WS-LINE-SPACING TO WS-LINES-USED.
APPENDIX C

Error Corrections for the Diagnostic Tasks
CORRECTIONS FOR DIAGNOSTIC TASKS

The following is a list of program corrections for the diagnostic tasks found in Appendices A and B. It should serve only as a guide, since alternative solutions exist (It should be noted that such alternatives were considered when study subjects' problems were graded). Although the number of modifications for certain tasks may appear to be minimal, this factor in itself does not necessarily reflect the difficulty of the task.

SYNTAX TASK ONE

<table>
<thead>
<tr>
<th>Line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IDENTIFICATION is misspelled</td>
</tr>
<tr>
<td>15</td>
<td>Indentation of 4 spaces is needed to move statement to area B</td>
</tr>
<tr>
<td>17</td>
<td>Indentation of 4 spaces is needed to move statement to area B</td>
</tr>
<tr>
<td>28</td>
<td>Line should end with a period</td>
</tr>
<tr>
<td>38</td>
<td>Should be PIC X(28)</td>
</tr>
<tr>
<td>63</td>
<td>Should be PIC X(3)</td>
</tr>
<tr>
<td>74</td>
<td>PERFORM is misspelled</td>
</tr>
<tr>
<td>76</td>
<td>Should be WS-END-OF-FILE-SWITCH instead of WS-EOF-SWITCH</td>
</tr>
<tr>
<td>99</td>
<td>Period should be omitted</td>
</tr>
<tr>
<td>100</td>
<td>Hyphen needed between FILE and SWITCH</td>
</tr>
<tr>
<td>100</td>
<td>Line should end with a period</td>
</tr>
</tbody>
</table>
### SYNTAX TASK TWO

<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>Line should end with a period</td>
</tr>
<tr>
<td>47</td>
<td>Hyphen needed between WORKING and STORAGE</td>
</tr>
<tr>
<td>55</td>
<td>Should be VALUE +55</td>
</tr>
<tr>
<td>64</td>
<td>Should be PIC 9(2)</td>
</tr>
<tr>
<td>68</td>
<td>Hyphen needed between SALESPEOPLE and ACCUM</td>
</tr>
<tr>
<td>69</td>
<td>Hyphen needed between QUOTA and ACCUM</td>
</tr>
<tr>
<td>70</td>
<td>Hyphen needed between REVENUE and ACCUM</td>
</tr>
<tr>
<td>87</td>
<td>8 more spaces needed between quotes</td>
</tr>
<tr>
<td>93</td>
<td>VALUE &quot;327&quot; should be omitted</td>
</tr>
<tr>
<td>123</td>
<td>Should be PIC X(12)</td>
</tr>
<tr>
<td>128</td>
<td>Should be PIC X(1)</td>
</tr>
<tr>
<td>137</td>
<td>Should be PIC ZZ,ZZZ,ZZZ.99</td>
</tr>
<tr>
<td>154</td>
<td>Should be PIC X(4)</td>
</tr>
<tr>
<td>168</td>
<td>Should be FIELDS instead of FIELD</td>
</tr>
<tr>
<td>171</td>
<td>Should be END-OF-FILE instead of ENDOFFILE</td>
</tr>
<tr>
<td>204</td>
<td>No period needed at end of line</td>
</tr>
<tr>
<td>243</td>
<td>Should be H2-HEADING</td>
</tr>
<tr>
<td>246</td>
<td>Should be H2-HEADING</td>
</tr>
</tbody>
</table>

### LOGIC TASK ONE

<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>214</td>
<td>Eliminate period at end of line</td>
</tr>
</tbody>
</table>
| 214   | Insert WS-TOTAL-RPT-PER-EARN-ACCUM  
      WS-TOTAL-RPT-YTD-EARN ACCUM.  
      between 214 and 215 |
<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>231</td>
<td>Insert MOVE ZEROS TO WS-PLANT-PER-EARN-ACCUM WS-PLANT-YTD-EARN-ACCUM. between 231 and 232</td>
</tr>
<tr>
<td>271</td>
<td>Insert MOVE &quot;NO&quot; TO WS-PLANT-CODE-CONTROL. MOVE &quot;NO&quot; TO WS-DEPT-CODE-CONTROL. between 271 and 272</td>
</tr>
</tbody>
</table>

**LOGIC TASK TWO**

<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>174</td>
<td>Insert PERFORM 7700-PRINT-TOTAL-LINE. Between 174 and 175</td>
</tr>
</tbody>
</table>

**LOGIC TASK THREE**

<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>258</td>
<td>Insert ELSE before IF</td>
</tr>
<tr>
<td>261</td>
<td>Insert ELSE before IF</td>
</tr>
</tbody>
</table>

**LOGIC TASK FOUR**

<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>263</td>
<td>Change 9 to 1</td>
</tr>
</tbody>
</table>

**LOGIC TASK FIVE**

<table>
<thead>
<tr>
<th>line #</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>222</td>
<td>Insert IF ER-THIS PERIOD EARNINGS IS NOT NEGATIVE between lines 222 and 223</td>
</tr>
<tr>
<td>224</td>
<td>Eliminate period at end of line</td>
</tr>
<tr>
<td>line #</td>
<td>Correction</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>224</td>
<td>Insert ELSE MOVE ZERO TO WS-ANNUALIZED-TAXABLE EARNINGS.</td>
</tr>
<tr>
<td></td>
<td>between lines 224 and 225</td>
</tr>
<tr>
<td>227</td>
<td>Change 10000 to 1000</td>
</tr>
<tr>
<td>230</td>
<td>Insert IF WS-ANNUALIZED-TAXABLE-EARNINGS IS NEGATIVE MOVE ZERO TO WS-</td>
</tr>
<tr>
<td></td>
<td>ANNUALIZED-TAXABLE-EARNINGS.</td>
</tr>
<tr>
<td></td>
<td>between lines 230 and 231</td>
</tr>
</tbody>
</table>
APPENDIX D

Diagnostic Task Scores,
GEFT scores,
and Experience Values
### RAW DATA

INCLUDING DIAGNOSTIC TASK VALUES, GEFT SCORES, AND EXPERIENCE VALUES

<table>
<thead>
<tr>
<th>ROW</th>
<th>SYNTAX TASK</th>
<th>LOGIC TASK</th>
<th>GEFT EXPERIENCE**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lc* inc pe</td>
<td>lc inc pe</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>27 0 9</td>
<td>6 0 0</td>
<td>12 0</td>
</tr>
<tr>
<td>2</td>
<td>25 1 0</td>
<td>5 0 0</td>
<td>14 0</td>
</tr>
<tr>
<td>3</td>
<td>26 0 1</td>
<td>7 0 0</td>
<td>18 1</td>
</tr>
<tr>
<td>4</td>
<td>26 1 1</td>
<td>5 0 1</td>
<td>14 0</td>
</tr>
<tr>
<td>5</td>
<td>23 2 1</td>
<td>4 3 2</td>
<td>18 1</td>
</tr>
<tr>
<td>6</td>
<td>26 2 6</td>
<td>9 0 0</td>
<td>16 0</td>
</tr>
<tr>
<td>7</td>
<td>25 0 1</td>
<td>3 2 0</td>
<td>13 1</td>
</tr>
<tr>
<td>8</td>
<td>26 1 2</td>
<td>7 1 0</td>
<td>12 1</td>
</tr>
<tr>
<td>9</td>
<td>23 1 3</td>
<td>4 2 2</td>
<td>10 1</td>
</tr>
<tr>
<td>10</td>
<td>20 0 0</td>
<td>6 0 0</td>
<td>9 0</td>
</tr>
<tr>
<td>11</td>
<td>28 1 2</td>
<td>5 0 0</td>
<td>9 1</td>
</tr>
<tr>
<td>12</td>
<td>23 2 3</td>
<td>2 2 0</td>
<td>13 0</td>
</tr>
<tr>
<td>13</td>
<td>27 0 0</td>
<td>4 2 0</td>
<td>13 0</td>
</tr>
<tr>
<td>14</td>
<td>19 4 5</td>
<td>2 2 0</td>
<td>14 1</td>
</tr>
<tr>
<td>15</td>
<td>26 1 1</td>
<td>7 0 0</td>
<td>18 1</td>
</tr>
<tr>
<td>16</td>
<td>20 1 2</td>
<td>3 1 3</td>
<td>12 0</td>
</tr>
<tr>
<td>17</td>
<td>19 3 5</td>
<td>3 2 1</td>
<td>7 0</td>
</tr>
<tr>
<td>18</td>
<td>24 1 5</td>
<td>5 0 0</td>
<td>17 0</td>
</tr>
<tr>
<td>19</td>
<td>24 1 3</td>
<td>6 2 2</td>
<td>18 1</td>
</tr>
<tr>
<td>20</td>
<td>25 2 1</td>
<td>3 3 1</td>
<td>18 0</td>
</tr>
<tr>
<td>21</td>
<td>25 1 2</td>
<td>5 0 1</td>
<td>4 0</td>
</tr>
<tr>
<td>22</td>
<td>24 1 1</td>
<td>7 0 0</td>
<td>16 0</td>
</tr>
<tr>
<td>23</td>
<td>26 0 2</td>
<td>4 1 0</td>
<td>8 0</td>
</tr>
<tr>
<td>24</td>
<td>25 2 1</td>
<td>7 0 0</td>
<td>17 0</td>
</tr>
<tr>
<td>25</td>
<td>23 3 1</td>
<td>2 4 0</td>
<td>15 1</td>
</tr>
<tr>
<td>26</td>
<td>22 1 4</td>
<td>4 1 2</td>
<td>2 1</td>
</tr>
<tr>
<td>27</td>
<td>27 1 1</td>
<td>4 3 0</td>
<td>5 1</td>
</tr>
<tr>
<td>28</td>
<td>27 0 2</td>
<td>4 0 0</td>
<td>16 1</td>
</tr>
<tr>
<td>29</td>
<td>25 1 1</td>
<td>5 1 0</td>
<td>18 0</td>
</tr>
<tr>
<td>30</td>
<td>23 1 2</td>
<td>4 1 0</td>
<td>18 0</td>
</tr>
</tbody>
</table>

* lc - # of errors located and corrected by subject  
  inc - # of errors located, but not corrected by subject  
  pe - # of statements marked in error by subject that were actually correct

** 0 - subject meets novice programmer criteria  
  1 - subject exceeds novice programmer criteria