An Approach to Teaching Multiple Computer Languages

Shouhong Wang
swang@umassd.edu
Department of Marketing/Business Information Systems
University of Massachusetts Dartmouth
Dartmouth, MA 02747-2300, USA

ABSTRACT

In the digital economy era, business information systems students need to be knowledgeable of multiple computer programming languages in order to meet the requirements of computer literacy. This paper outlines the pedagogy of an innovated course of multiple computer languages for business students in the major of information systems. It discusses the rationale of why the proposed pedagogy is different from and better than traditional ones, and describes the approach to teaching this course. Based on our experiences in the past three years, it is concluded that a single course of multiple computer languages is useful and feasible.

Keywords: IS education, pedagogy, multiple computer programming languages

1. INTRODUCTION

There have been dramatic changes in information technology during the last decade. Most notable is the advancement of computer literacy of millions of information systems (IS) professionals in business and management as a result of the proliferation of electronic commerce. The changes have considerable implications for institutions of higher education whose responsibility is to train the next generation of business IS professionals. In our view, business IS students must acquire fundamental theories of IS as well as essential practical skills in computer applications. They must also develop the life-long learning ability in information technology during their business education. Recently, there is a considerable need for redesigning business IS education curriculum (IS'2000 2001). Academic institutions are required to pay increasing attention to courses of practical IS skills based on long-term strategic considerations. This paper is to report how this challenge was met by designing the contents of an innovated course of multiple computer programming languages for business IS students.

2. THE INNOVATED COURSE OF COMPUTER PROGRAMMING LANGUAGES

The course described in this paper is entitled "Programming and Problem Solving," and is designed for sophomore students in the major of business information systems. There is no specific prerequisite other than a course of introduction to business. It is taught over one semester, normally 14 weeks. In its design, this course consists of two distinct modules. The Teaching Module provides an overview of representative computer programming languages in business computing. The Project Module involves hands-on projects.

The teaching of a comprehensive computer programming course is complicated by the extensive nature of the subject. As no report on teaching multiple computer languages in a single course has been found in the literature or on the Internet, the selection of computer languages for the course is the crucial task for the pedagogy design. The fact is that, in the modern computer age, the development of IS still heavily relies on applications of third generation computer languages regardless the advance in fourth generation computer languages and a variety of software packages. Interestingly, this is truer in Web-based applications development. To meet the challenge of the ever-changing computer technology, business IS students must be computer-literate in terms of understanding major programming languages. On the other hand, they cannot afford to learn multiple computer languages on

the one-language-one-course basis. The key solution to this problem is to make a pedagogical paradigm shift and develop an innovated course.

We design the components of this course based on the skill demands from the IS job market. There have been many surveys about hard skills required for IS jobs. According to Papp's survey (Papp 1998) for the New England region, the skills in demand as found in the classified IS job advertisements are 16.0% COBOL, 13.4% C++, 10.8% Visual Basic, 9.3% C, 5.0% Java, and 3.0% HTML, which comprise the total of 57.5% general design and development skills. Taking this into account, six major computer programming languages were selected for this course. They are COBOL, C++, HTML, JavaScript, Java, and Visual Basic.

In terms of training of information technology, the goal of education for business IS students is quite different from that for computer science students, although the boundary between the two may not be clear-cut. In computer science programs, the major goal of programming courses is to teach students how to decompose a task and then map this plan into constructs of the target programming language (Sleeman 1986). Task decomposition depends on the problem domain, and there are so many problem domains are essential for computer science students, including data structure, scientific computation, file processing, Web applications, multimedia, etc. More importantly, the constructs must be available in the particular programming language. For these reasons, computer science programming courses are typically designed on the one-language-one-course basis and emphasize the syntax of the language. Also, it is common in computer science programs that a first course in programming is not influenced by a specific programming language (Malik 2000). A recent study of teaching computer science suggested to keep things as close to pure logic as possible and to treat computer languages as notational systems (Palma 2001). Apparently, the goals and process of teaching programming in computer science programs are not fully applicable to the business IS education. We would like to teach business students to be more business oriented and business problems driven. Instead of teaching general technical details, such as formal syntax and computational algorithms, we should teach specifically how to match typical business problems and their computer solutions.

Due to time constraints, it is impossible for students to learn all these computer languages in great details. Nevertheless, students in this course are expected to have a bird's-eye view of computer programming languages as well as to develop practical skills of programming. The objective of this course is that students will understand the characteristics of traditional data file processing in legacy information systems, the philosophies of structured programming and object-oriented programming, the means of Web page development for the Internet, and the concept of human-computer interface design and decision support systems. The central methodology applied to this course is case study. Specifically, we teach typical problems of business computing and their solutions of these computer languages.

3. MOTIVATION OF THE PEDAGOGY DESIGN

Currently, many business undergraduate programs offer courses of computer languages to IS majors. However, these courses are usually single-language courses. During the past years, we have identified opportunities to develop the course of multiple computer languages for business IS students, and have found the new design of such an innovated course have many advantages, as discussed below.

3.1. The Needs for Learning Multiple Computer Languages

Due to historical and technical reasons, there have been many computer languages in the information age, and each computer language serves its particular purpose. The needs for learning multiple computer languages can be perceived from a variety of views. From the perspective of basic job skill requirements for IS students, COBOL is still important to learn since many legacy systems are COBOL-based. However, modern Internet-based computing heavily relies on HTML, JavaScript, and Java. Furthermore, from the viewpoint of learning IS tools for management information systems and decision support systems, Visual Basic is more important than other languages. Nevertheless, from the standpoint of computer languages themselves, C and C++ are fundamental in the software industry, and are the typical sister languages of the two different programming paradigms: function-oriented and object-oriented. Given that fact that almost every textbook of information systems analysis and design includes material on object-oriented approach, IS students must learn the basic concept at the language level.

There is little doubt that knowledge of various computer languages can contribute to the learning and further career development for IS students. Then, the question is how we can teach them in a feasible way.

3.2. The Feasibility of Teaching Multiple Computer Languages

It is impossible for us to teach multiple languages by using the traditional one-language-one-course approach. To compact material of multiple languages into a single course, we must identify what concepts are essential for business IS students, and how these concepts can be

delivered. After we reviewing the IS'97 model curriculum (IS'2000 2001), we determined that the following concepts are essential for IS students.

- Data processing (file systems);
- Simple data types and string manipulation;
- If-then control structure;
- Loop;
- Function calling and function orientation;
- Object orientation and message sending;
- Web page development;
- Client-server computing;
- Graphical user-computer interface development; and
- Various environments for computer languages.

The relationships between these key concepts offered in individual categories of languages and the general requirements of IS education is summarized in Table 1. As shown in Table 1, the IS'97 model curriculum defines computer literacy as a set of knowledge elements related to problem solving using the computer technology. Most knowledge elements listed in Table 1, such as abstract data types, modules and coupling, software development, human-computer interfaces, object-oriented methodologies, and programming languages can be acquired only through the learning of computer languages. Hence, being computer language capable contributes to computer literacy significantly.

We consider many concepts included in traditional language textbooks, such as pointers and sophisticated data types and structures, to be less important for IS undergraduate students and not be emphasized in this course. To better teach these essential concepts of languages, we must change the traditional teaching approach that begins with syntax and ends with disjoined examples of syntax explanations and use cases. The motivation to use the case study method was triggered by the fact that our IS students were generally unsatisfied with the programming course that had been taught using the traditional approach before the redesign. Appendix A shows such a typical case of payroll processing for COBOL data processing.

Despite the variety of syntax in the different languages, languages share many common key concepts, such as if-then control, loop, and function-calling/message-sending. Once a key concept is introduced in one language, it should be understandable for students in another language.

3.3. The Pedagogical Shift Adds More Value to Student Learning

This innovated course is fast-paced, and encourages self-learning on the students' part. Since the nature of this course is practice-oriented, multiple small-scale projects are required. The central point of the pedagogical shift from one-language-one-course to multiple-languages-one-course is to add more value to student learning by giving students discipline in self-learning and encouraging them to apply newly learned knowledge to the real world. According to our observations, students might feel stressful in any programming course, but never felt bored in this course. Next, we describe the course design in more detail.

4. TEACHING MODULE

For a long time, it has been difficult to find a single integrated textbook that meets our needs. In order to give students a comprehensive guideline for their studies, a lecture note was developed for this course during the past years. This manuscript has been revised many times, and has recently published as a textbook (Wang and Wang 2000) that can be downloaded from the Internet.

The Teaching Module is divided into four units based on the categories of business computing problem solving, each of which is followed by a written exam. The features of the course competency, relevant paradigms, and business applications are presented below.

4.1. Unit 1: Data Processing and COBOL

File processing is an essential type of business data processing. COBOL-based file processing is the backbone of legacy information systems. In this unit, the techniques involved in file processing, such as three organizations (i.e., sequential, random, and indexed), are discussed. Because these techniques are built in COBOL programming, emphasis is placed on the comparison of the three file organizations in terms of their advantages and disadvantages in various circumstances of data processing.

Since COBOL is still a commonly used computer language in the business community, and is the first computer language for most business IS students to learn, about 29% class time is devoted for COBOL. Instead of emphasizing the syntax, we present four typical COBOL programs related to payroll processing. These four typical programs give students an undivided prototype of business data processing functions: creating a master file, creating a transaction file, manipulating these data files, and maintaining the data files.

The teaching emphasis is placed on the following four aspects of COBOL: the four-division structure of COBOL programs, file declarations and file organization descriptions, data type (PIC) descriptions, and structured

programming in the procedure division. The COBOL format and various statements are explained in the context of these four examples.

Key Concepts	Requirements of IS Education (IS'97 - Competency Level and Body of Knowledge Elements)	COBOL	C/C++	HTML JavaScript Java applet	Visual Basic
Data processing	4 1.2.1 Formal problems and problem solving	✔			
Simple data types	4 1.2.4. Abstract data types	✔	✔	✔	✔
If-then control	4 1.2.1 Formal problems and problem solving	✔	✔	✔	✔
Loop	4 1.2.1 Formal problems and problem solving	✔	✔	✔	✔
Function orientation	3 1.2.4.4 Modules and coupling.	✔	✔		
Object orientation	3 3.3.6 Object-oriented methodologies		✔	✔	✔
Web page development	4 3.9.7 Software development			✔	
Client-server computing	3 3.1.2 Systems concepts			✔	
Graphical user-computer Interfaces	4 3.9.6 Human-computer interfaces				✔
Environments of languages	3 1.3.7 Programming languages, design, implementation.	✔	✔	✔	✔

Table 1. Relationships between the Key Concepts and Requirements of IS Education

The computing environment for this unit is COBOL on the Alpha mainframe computer. Students use PowerTerm, a terminal emulator for Windows, to edit and run COBOL programs.

4.2. Unit 2: Object-oriented Approach and C++

In order to meet the challenge of object-oriented methods in the computer world based on long-term considerations, we recognize that general knowledge of C++ will be an asset for IS students. The focus of this unit is placed on the difference between the traditionally structured approach and the object-oriented approach. Similarly to teaching COBOL, the features of essential notations and statements of the language, including data type, arithmetic operations, for-loop, if statement, and print statement are explained using examples. Typical examples for this unit range from a single-class program (e.g., print a flyer for a store) and two-class program (e.g., order processing) to multiple-class program (e.g., payroll processing).

C++ is an extension of C. In this unit, students actually

learn both C and C++ languages. Students compare the structured approach and object-oriented approach by learning two different themes in C and C++ programming. In the functional theme of C, students learn the structure of elementary function modules and the connections between the function modules through function calling. The principle of programming in this theme is functional decomposition. In the object-oriented theme of C++, students learn the concept and structure of classes, the characteristics of inheritance, and message sending in object-oriented programming.

We use 24% class time for this unit. The computing environment for this unit is Visual C++ of Microsoft Visual Studio 6.0.

4.3. Unit 3: Web Pages, HTML, JavaScript, and Java

The development of Web pages is probably the most interesting topic for business IS students. As future IS professionals, business IS students have to develop skills of computer applications in the Internet environment. These skills are acquired only when students understand client-server computing through learning HTML, JavaScript, and Java applets.

In this unit, students are required to learn the significant features and components of the integration of HTML, JavaScript, and Java applets using 26% of the entire class time. These components include essential tags of HTML, the structure of simple Web pages with JavaScript, Web pages with Java applets, and typical Java applets for animation with audio presentations. The course also gives introductions to the stand alone Java programming technique (AWT-based and non-AWT-based) as an optional part. The emphasis of this optional part is placed on a comparison of Java and C++.

In teaching this unit, typical examples of Web pages with cookie, FORM verification, and multimedia presentation supported by Java applets are presented. The requirement of this unit is the integration of HTML, JavaScript, and Java applets. The computing environment for this unit is Netscape or Microsoft Internet Explorer and Visual J++ of Microsoft Visual Studio 6.0.

4.4. Unit 4: Graphical User Interface, Decision Support System, and Visual Basic

In this stage of the course, students have developed basic skills in business data processing and computing on the Internet. This unit provides another important aspect of business computing: graphical user interfaces (GUI) and decision support systems (DSS).

Visual Basic is one of the popular programming languages for GUI. Students learn the major tools of developing GUI by mastering control elements including form, command button, label, text, combo menu, and program module. In the programming part, the concepts of do-loop, if statement, arithmetic operations, print and format functions are re-explained in the Visual Basic syntax by an example of on-line invoicing system. To develop their self-learning ability, students are required to use on-line help to learn built-in functions (such as NPV calculation for finance).

After learning Visual Basic, students learn Visual Basic for Applications in this unit. The design of this part is based on the considerations that business IS students must know the concept of DSS and the interaction between GUI and the model management.

A general overview of DSS is presented in this part. This is followed by a demonstration of a mini-prototype of a DSS developed in Excel Visual Basic for Applications. In this DSS example, models are the three worksheets containing variables and parameters as well as formulas. One worksheet for the Production Division calculates the production cost of a product based on material costs, labor costs, and overhead, which are all nonlinear functions of production of products. The second worksheet for the Marketing Division calculates the marketing cost of the product based on advertising costs, salesman wages, and marketing overhead, which are also nonlinear functions of production of products. The third worksheet generates an accounting statement, which determines the net income for the company based on the sale price of the product and the total cost of production and marketing. The DSS assists the planning manager in deciding an "optimal" production level through "what-if" trials by using the designed GUI. Through this example of Visual Basic for Applications, students learn how to integrate the GUI and the model management system to develop DSS.

The computing environment for this unit is Visual Basic 6.0 of Microsoft Visual Studio 6.0 and Microsoft Excel 97.

4.5. Examinations

After each of the four units, students are required to write an exam. Students are presented with uncompleted computer programs with numerous blanks. Students are supposed to complete the programs, write the expected execution results of the programs, and explain the purposes of these programs. The designs of the tests strongly suggest that the global aspects of applications, including context and outcomes, are more important than the local aspects (syntax) to learn for business IS students. A sample exam is exhibited in Appendix B.

5. PROJECT MODULE

The Project Module of the course concentrates much more on "action learning", and requires students to conduct their quasi-real-world projects. In this module, students work in teams, typically, of three people. Each group applies computer programming languages discussed in the Teaching Module for their four projects. We expect that, upon the completion of this course, students are able to write programs in these computer languages to solve simple problems related to business IS.

5.1. Project Configuration

After each unit presented in the Teaching Module, students form project teams to share common expectations of learning. Three stages can be outlined for each project. In the first stage, students choose project topics based on their general understanding of the computer languages and their cases. In the second stage, intensive programming activities are under way for each group's project. In the final stage, the project product is formalized in a report. Although the components of reports are significantly different from project to project, the report structure adopted in four projects is almost the same. The first part of the report describes the background of the business problem to be solved. The second part provides a technical description of the project and presents source code, module diagrams, and exemplars of program execution results. The third part is a user manual.

5.2. Project Requirements

Because the natures of computer programming languages are very different from one to another, there is no uniform requirement for the four projects. Nevertheless, we set criteria for each project.

For file processing and COBOL, the basic requirements are:

- Structured programs for creating master files and transaction files, and manipulating these files to generate designed print reports.
- Structured diagrams for the programs.

For C++, the basic requirements are:

- Object-oriented programs with two or more classes for data manipulations.
- Object-oriented diagrams for the programs.

For HTML, JavaScript, and Java, the basic requirement is:

- Code of HTML, JavaScript, and Java applets for Web pages, hyperlinks, FORM data captures.
- Multimedia presentations including images and animations.

For Visual Basic, the basic requirements are:

- Two or more forms of GUI with various control elements, and print report generation.
- Program modules.

Students are asked to follow the structures of the cases they learned and add their own creative components for their projects. For instance, after learning the typical payroll processing system, students are generally able to write COBOL programs for their own projects such as inventory processing, sales commission processing, student credits processing, etc. Based on our observations, we are convinced that students are able to write complete programs from scratch in each of the languages for typical simple business IS problems. In our view, computer literary for IS students means "be knowledgeable with various computer languages", and such an aptitude can be developed only through hands-on practices. Appendix C exhibits a part of the course syllabus that describes the requirements of the four projects.

6. FINDINGS AND CONCLUSION

This innovated course has been offered for three years in the College of Business at University of Massachusetts Dartmouth as a required BIS major course. Two hundred twenty three (223) students have completed this course during the past three years. After the course, each student was asked to answer the standard course evaluation questionnaire used at the university. Each question is rated on a five-point scale for students to answer. A four-point or higher mark for a question is considered a positive answer to the question for this study. Two hundred five (205) filled course evaluation forms from these students have been recorded. According to the course evaluations, 172 of 205 students gave a positive answer for the question of overall satisfaction with the course, indicating overall 84% of the students were satisfied with this course. 85% of the students considered that the material covered in this course was interesting and useful. 92% of the students favored the two-module approach to this type of course. Nevertheless, 8% students felt that the course workload is too high. According to our observations on the test results and projects, students who are eligible to take this course have no difficulty in learning the techniques outlined in this paper and conduct projects. After the course, some students have demonstrated their ability to participate more formal projects.

This approach to teaching programming for IS students is better than the traditional one-language-one-course approach in terms of skill development and knowledge acquisition. After this course, students are able to program in these different languages to solve simple business problems of data processing and Web page development, although they may not be so proficient in a particular language. These diversified basic skills will

enhance students' self-learning ability. Students after this course know more about traditional and modern computer languages than they would do after a single-language course. Knowledge of multiple languages will certainly be beneficial for them in studying other IS courses and developing hard skills for job.

Overall, the results indicate that the course discussed in this paper is a valuable component of the curriculum for business students with the major in IS. We believe that a single compact course of multiple computer languages for business IS students is useful as well as feasible. All computing resources (hardware and software) needed for this course are commonly available at education institutions. The plain textbook is available on the Internet. Thus, it is certain that our success can be replicated in other business schools.

7. ACKNOWLEDGEMENT

The comments of the Editor and two anonymous reviewers have contributed significantly to the revision of the paper.

8. REFERENCES

IS'2000 [2001], IS'97 Documents, <http://www.is2000.org/> [Accessed August 1, 2001].

Malik, M. A. [2000] "On the perils of programming," *Communications of the ACM*, 43(12), 95-97.

Palma, P. D. [2001], "Why women avoid computer science," *Communication of the ACM*, 44(6), 27-29.

Papp, R. [1998], "Job Skills in New England," <http://www.commerce.uq.edu.au/isworld/announce/msg.21-10-1998-1.html> [Accessed August 1, 2001].

Sleeman, D. [1986], "The challenges of teaching computer programming," *Communication of the ACM*, 29(9), 840-841.

Wang, Shouhong and Hai Wang [2000], *Problem Solving and Programming: Essentials of Computer Languages for Commerce*, Universal Publisher, Parkland, FL, <http://www.upublish.com/books/wang.htm> [Accessed August 1, 2001].

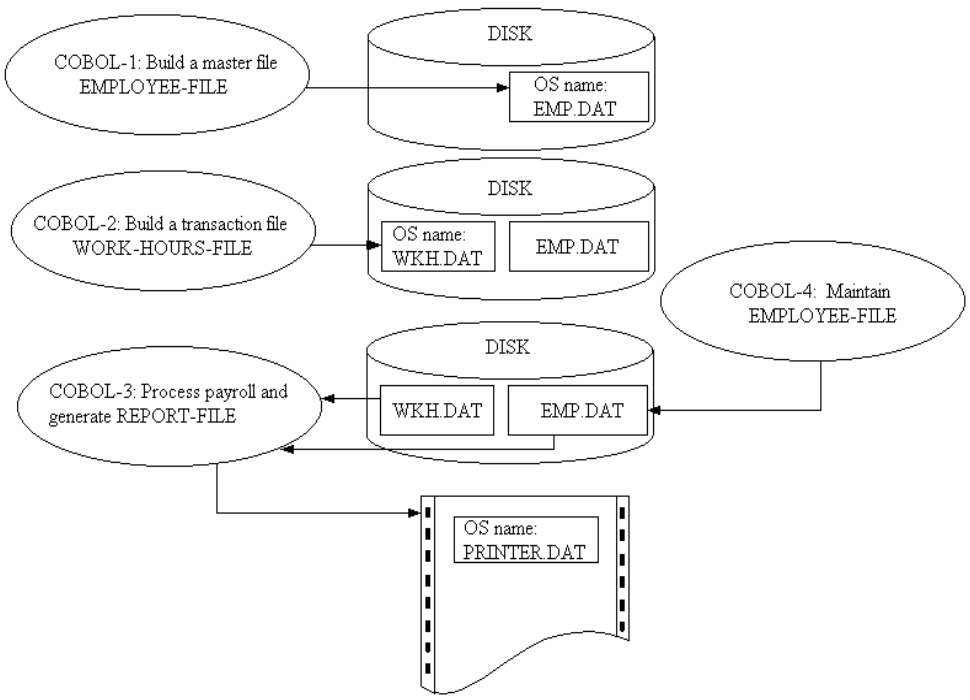
AUTHOR BIOGRAPHY

Shouhong Wang is a Professor of Business Information Systems at University of Massachusetts Dartmouth. He

received his Ph.D. (1990) in Information Systems from McMaster University, Canada. His research interests include systems analysis and design, artificial intelligence in business, and electronic commerce. He has published over 50 papers in academic journals, including *Journal of Management Information Systems, Information & Management, International Journal of Information Management, Human Systems Management, Information Systems Management, IEEE Transactions on Systems, Man, and Cybernetics, Computers & Operations Research, Management Science, European Journal of Operational Research, OMEGA, Decision Sciences*, and others. He has been listed in the *Canadian Who's Who* since 1995, and in the *Who's Who in America* since 2000.

APPENDIX A: CASE STUDY- PAYROLL PROCESSING
(Adopted from the Textbook)

To provide typical COBOL programs, we show four examples of COBOL program for a payroll processing system in Listings 2.3, 2.4, 2.5, and 2.6. After each of the example programs, we provide detailed explanations for the COBOL sentences used in the program. The purpose of the first COBOL program in Listing 2.3 is to build an indexed file for employees. Once the employee master file has been built, it can be used for queries and other data processing. The second COBOL program in Listing 2.4 is to create a sequence data file that records each employee's work hours. These two programs in our examples are actually data entry support programs. The third COBOL program in Listing 2.5 manipulates the data files created by the previous two programs for payroll processing. The fourth COBOL program in Listing 2.6 is a maintenance program that is used to make modifications to the master file. The relationships between the COBOL programs and the data files are depicted in the following figure.



APPENDIX B: SAMPLE EXAM

1. Read the following COBOL program and complete it by filling the blanks. [10 marks in total, the numbers in the blanks indicate the mark for each of them.]

```
_____________(0.2) DIVISION.
PROGRAM-ID.        COMMISSION.

_____________(0.2) DIVISION.
CONFIGURATION      SECTION.
SOURCE-COMPUTER.   ALHPA.
_____________(0.2) SECTION.
FILE-CONTROL.
     SELECT        ____________________(1)
        ASSIGN TO "TEAM.DAT"
        ORGANIZATION IS INDEXED
        ACCESS MODE IS DYNAMIC
        RECORD KEY IS SALESMAN-ID-A.

     SELECT   MONTHLY-SALES-FILE
        ASSIGN TO "SALES.DAT"
        ORGANIZATION IS SEQUENTIAL.

     SELECT   COMMISSION-REPORT
        ASSIGN TO PRINTER.

______(0.2)        DIVISION.
FILE               SECTION.
FD  SALES-PERSON-FILE
    LABEL RECORDS ARE STANDARD.
01 ____________________(1).
    02  SALESMAN-ID-A              PIC X(2).
    02  SALESMAN-NAME              PIC X(30).
    02  SALESMAN-ADDRESS           PIC X(30).
    02  SALESMAN-COMMISSION-RATE   PIC 99.

FD  MONTHLY-SALES-FILE
    ____________________________(0.3).
01  MONTHLY-SALES-RECORD.
    02  SALESMAN-ID-B              PIC X(2).
    02  MONTHLY-SALES              PIC 99.
FD ____________________(1)
    LABEL RECORDS ARE OMITTED.
01  COMMISSION-RECORD              PIC X(80).

WORKING-STORAGE  SECTION.
01  COMMISSION-BEFORE-TAX          PIC 9(4)V99.
01  COMMISSION-AFTER-TAX           PIC 9(4)V99.
01  TAX-DEDUCTION                  PIC 9(4)V99.
01  EOF-SIGN                       PIC XXX      VALUE SPACES.

01  TITLE-LINE.
    02   FILLER   PIC X(10)  VALUE "SLM-ID".
    02   FILLER   PIC X(20)  VALUE "SALESMAN NAME".
    02   FILLER   PIC X(12)  VALUE "SALES".
    02   FILLER   PIC X(12)  VALUE " CMS+TAX".
    02   FILLER   PIC X(12)  VALUE " TAX DED".
    02   FILLER   PIC X(12)  VALUE "   CMS-PAY".

01  MONTHLY-COMMISSION-RECORD.
    02   SALESMAN-ID-R              PIC X(2).
    02   FILLER                     PIC X(8) VALUE SPACE.
    02   SALESMAN-NAME-R            PIC X(20).
    02   FILLER                     PIC X(8) VALUE SPACE.
    02   MONTHLY-SALES-R            PIC 99.
    02   FILLER                     PIC X(3) VALUE SPACE.
    02   COMMISSION-BEFORE-TAX-R    PIC $$$$9.99.
    02   FILLER                     PIC X(3) VALUE SPACE.
    02   TAX-DEDUCTION-R            PIC $$$$9.99.
    02   FILLER                     PIC X(3) VALUE SPACE.
```

```
    02   COMMISSION-AFTER-TAX-R      PIC $$$$9.99.

_____________(0.2)       DIVISION.

_____________(0.3).
    PERFORM START-PROCESS.
    PERFORM 01-COMMISSION-PROCESS
            UNTIL EOF-SIGN = _________(0.5).
    PERFORM _____________(0.3).
    STOP    RUN.

_____________(1).
    OPEN  INPUT SALES-PERSON-FILE
          INPUT MONTHLY-SALES-FILE
          ___________(0.5) COMMISSION-REPORT.
    MOVE  TITLE-LINE   TO  COMMISSION-RECORD.
    WRITE COMMISSION-RECORD.
    READ  MONTHLY-SALES-FILE
          AT END      MOVE "FIN" TO EOF-SIGN.
01-COMMISSION-PROCESS.
    MOVE  SALESMAN-ID-B  TO  SALESMAN-ID-A.
    READ  SALES-PERSON-FILE
          KEY IS _____________(1)
          INVALID KEY DISPLAY "THE SALES PERSON IS MISSING!".

    PERFORM  02-CALCULATE-COMMISSION.
    PERFORM  02-PRINT-A-COMMISSION-RECORD.
    READ  MONTHLY-SALES-FILE
          AT END      MOVE ________(0.5) TO EOF-SIGN.

02-CALCULATE-COMMISSION.
    COMPUTE   COMMISSION-BEFORE-TAX =
              SALESMAN-COMMISSION-RATE * MONTHLY-SALES.
    COMPUTE   TAX-DEDUCTION = COMMISSION-BEFORE-TAX * 0.2.
    COMPUTE   COMMISSION-AFTER-TAX =
              COMMISSION-BEFORE-TAX - TAX-DEDUCTION.

02-PRINT-A-COMMISSION-RECORD.
    MOVE SALESMAN-ID-B    TO  SALESMAN-ID-R.
    MOVE SALESMAN-NAME    TO  SALESMAN-NAME-R.
    MOVE MONTHLY-SALES    TO  MONTHLY-SALES-R.
    MOVE COMMISSION-BEFORE-TAX   TO   COMMISSION-BEFORE-TAX-R.
    MOVE TAX-DEDUCTION               TO   TAX-DEDUCTION-R.
    MOVE COMMISSION-AFTER-TAX        TO   COMMISSION-AFTER-TAX-R.
    MOVE MONTHLY-COMMISSION-RECORD   TO   COMMISSION-RECORD.
    WRITE _____________________(1)
           AFTER ADVANCING   2   LINES.

CLOSE-PROCESS.
    CLOSE _____________________(0.2)
          _____________________(0.2)
          _____________________(0.2).
    DISPLAY "COMMISSION REPORT IS GENERATED IN PRINTER.DAT!".
```

2. Draw a structure diagram for the above COBOL program. [1 mark]
3. Answer the following questions.
(1) How many files are used in the above COBOL program? [0.5 mark]
(2) Explain specifically why the organizations of the disk files are so designed. [1.5 mark]
(3) Suppose you apply the operating system command $TYPE and find the sample data of the disk files as follows:

TEAM.DAT

12	Jame	Garden	30
03	Ann	Park	40
05	Tim	Lake	50

SALES.DAT

0510
0310

If you run the above COBOL program, how many times does the program perform the following sentences or paragraphs?

START-PROCESS [0.3 mark]
STOP RUN [0.2 mark]
01-COMMISSION-PROCESS [1 mark]
02-PRINT-A-COMMISSION-RECORD [0.5 mark]

(4) Given the sample data in Question (3), write the expected **print result** generated by the above COBOL program. [5 marks]

APPENDIX C: PROJECT ASSIGNMENTS
(A Portion of the Course Syllabus)

The minimal requirements of four project assignments are:

(1) **COBOL**: create a master file and a transaction file, and manipulate these two data files and generate a well organized print report.
(2) **C++**: one main program and two classes for a business application.
(3) **Web page**: a business oriented Web page with HTML, JavaScript (e.g., checking FORM), Java applet (animation).
(4) **Visual Basic**: GUI with 2 forms, at least 3 command buttons, several label boxes and text boxes, 1 combo box, 1 module, and 1 message box. One of the command buttons is to print a report.
The guidelines of the four reports can be found in the textbook (Appendices 2-3, 3-5, 5-3, and 6-1).

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096