Implementing Generalized Audit Software in the Classroom

Volume 4, Number 1
2009
page 27 - 54

Michele Matherly
Xavier University, MatherlyM@xavier.edu, 513-745-3165
Marcia Weidenmier Watson
Mississippi State University, MWatson@cobilan.msstate.edu, 662-325-1638
Susan Ivancevich
University of North Carolina at Wilmington, IvancevichS@uncw.edu, 910-962-3969

ABSTRACT

This paper describes resources designed to help instructors easily integrate the leading generalized audit software (GAS) packages, Audit Command Language (ACL) and Interactive Data Extraction and Analysis (IDEA), into their classes. Specifically, we provide project assignments, exam questions, as well as teaching notes describing implementation issues. The projects require students to: (1) evaluate sales and accounts receivable data, and then create confirmation letters by exporting data from GAS and then importing it into word processing software for a mail merge, (2) use inventory, accounts payable, and payroll data to work with expressions, filters, data extractions, and relations between tables, and (3) import a database and prepare several reports. We also present student self-assessment data on their ability to use GAS as well as their understanding of the general benefits and applications of GAS. The evidence suggests that hands-on projects can improve perceived knowledge of GAS regardless of student background, course format (group versus individual projects), or teaching method (self-study versus lab-time).

Keywords
Computer-Assisted Auditing Techniques (CAATs); Generalized Audit Software (GAS); Audit Command Language (ACL); Interactive Data Extraction and Analysis (IDEA)

A teaching note and electronic files are available for use with this case. If you are member of the AIS Educator Association, please go to http://www.aiseducators.com and follow the links for the AIS Educator Journal. If you are not a member of the Association, please contact the author directly at the address provided above to obtain these materials. Please provide a means for verifying your credentials as a faculty member so that we may protect the integrity of the solutions materials.

INTRODUCTION

To help comply with the sweeping provisions of the Sarbanes-Oxley Act of 2002 (SOX), auditing with the computer has become more commonplace and a necessity for many companies, as evidenced by the increased use of generalized audit software (GAS) (Gray 2006; Jackson 2004). GAS allows auditors to improve the efficiency and effectiveness of audit procedures associated with the large scale operations found in many organizations and the proliferation of enterprise resource planning systems, which limit the traditional paper trail of legacy systems (Gelinas et al. 2001; Chapman 2002; Weidenmier and Herron 2004). The two leading GAS programs, Audit Command Language (ACL) and Interactive Data Extraction and Analysis (IDEA), are used extensively by both internal and external auditors around the world. For instance, ACL (the market leader) is currently used by all of the Big Four accounting firms, 70 percent of the Fortune 500 and more than two-thirds of the Global 500 (ACL 2008).

Due to the popularity of such software and concerns related to traditional accounting curricula,[1] many faculty have begun to incorporate GAS assignments into their courses. These types of projects help address calls for more active learning in accounting (AECC 1990), for learning experiences that foster more technological adeptness (AICPA 1999), and for learning approaches that require higher-order cognitive skills (AICPA 2008). When properly implemented, technologies such as computer-assisted auditing techniques (CAATs) enable students to actively participate (Thompson et al. 1992) in the learning process and to use higher-order analytical skills in reviewing and interpreting data (Bryant and Hunton 2000; Weidenmier and Herron 2004). Furthermore, the use of GAS in the classroom "can enhance understanding of computerized data files, concepts of computer auditing, audit software purposes and functions, internal control, risk assessment and the audit process in general" (McCombs and Sharifi 2004, 38)—all of which are important for SOX compliance.

Prior literature primarily focuses on the development and administration of case studies and assignments requiring the use of GAS (e.g., Gelinas et al. 2001; Nieschwietz et al. 2002; Weidenmier and Herron 2004; McCombs and Sharifi 2004). Gelinas et al. (2001) present a case study incorporating the use of ACL and report student feedback on the case experience. Nieschwietz et al. (2002) develop assorted assignments for use with IDEA. Weidenmier and Herron (2004) perform a comparative analysis of ACL and IDEA, in addition to providing projects related to each program. McCombs and Sharifi (2004) briefly describe their use of ACL and IDEA in the classroom and present summarized student evaluations of the projects. Table 1 summarizes the content coverage of published resources, which include journal articles and textbooks, and provides a comparison to the materials described in this paper.

While resources exist for integrating GAS into courses,[2] our paper represents an extensive supplement to the tutorials that accompany the two leading software packages, ACL and IDEA. To facilitate the classroom use of

[1] Examples of these concerns include the lack of hands-on, real-world educational experiences, courses that are too lecture-oriented, and ones that focus too much on accounting rules rather than on application and strategic simulation (Albrecht and Sack 2001).

[2] In addition to the articles and textbooks summarized in Table 1, Herron maintains a website of GAS teaching resources which can be accessed at http://www.business.umt.edu/faculty/herron/acl_idea/home.htm.

GAS, we provide instructors with additional projects for sales, accounts receivable, inventory, and database applications as well as new projects for accounts payable and payroll. We try to strike a balance between a

Table 1
Comparison of Generalized Audit Software Resources

Panel A: Available Articles

Project Characteristics	Splettstoesser (1999)	Gelinas et al. (2001)	Nieschwietz et al. (2002)	Lehman & Watson (2007)	Our Projects
Audit Software	Not specified.	ACL	ACL	ACL and IDEA	ACL and IDEA
Sales		X	X		
Accounts Receivable		X	X		
Inventory	X	X	X		X
Accounts Payable				X	X
Payroll					X
Database		(minor)	X	X	X
Data Sources	No dataset just case	ACL tutorial + Own dataset	IDEA tutorial + MS Northwind Traders	Own dataset	ACL tutorial + any Access database
Presentation Style	Students describe specific tests using GAS	Step-by-step	Step-by-step	Guidance helps students answer questions	Step-by-step

Panel B: Assignments in Textbooks Accompanied by ACL and IDEA Software

Project Characteristics	Arens (2004)	Hall & Singleton (2005)	Hunton et al. (2004)	Boynton & Johnson (2005)	Rittenberg et al. (2008)	Messier et al. (2008)
Audit Software	ACL	ACL	ACL	IDEA	ACL	ACL
Sales	X	X	Demo only			X
Accounts Receivable	X	X	X	X	X	X
Inventory	X	X	X	X	X	X
Accounts Payable	X	X	X			X
Payroll	X	X	X			X
Database	X	X	X			X
Data Sources	ACL tutorial	ACL tutorial	ACL tutorial	Own datasets	Own dataset	ACL tutorial + Own dataset
Presentation Style	Includes extensive screen shots and step-by-step instructions	Limited results screen shots, some guidance to answer questions, and step-by-step on-line tutorials	Provides specific questions to be answered	Cases with specific questions to answer	Includes screen shots and some instructions. Asks students to write audit programs and develop procedures	Gives students some step-by-step guidance initially

Note: This table is based on one developed by Del DeVries for the 2004 American Accounting Association annual meeting.

step-by-step, screen shot approach versus assignments with little to no guidance by giving students sufficient instructions to complete the projects on their own and still be able to interpret the results and use their judgment to make decisions. We also provide a flexible resource that faculty can use to implement GAS in the classroom or in conjunction with textbooks that provide no guidance for using the software. In addition, each project relates to a different transaction cycle so instructors can implement one, all, or any combination of the projects.

This paper explores the use of GAS projects as a way to enhance student comprehension of key material. Similar to Gelinas et al. (2001) and McCombs and Sharifi (2004), we describe the results from a pre- and post-survey used to evaluate the impact of the projects on students' (self-reported) knowledge of selected audit concepts. The survey data also enables us to assess whether using GAS projects improves student understanding of the benefits and applications of CAATs. Finally, we investigate whether student demographics, teaching approach, and/or course structure affect student self-assessments of GAS learning or project perceptions.

The rest of the paper proceeds as follows. The next section briefly describes the objectives and tasks involved in each project from the perspective of ACL, the most widely-used GAS. The third section details the methodology, while the fourth section reports the student self-assessment results. The last section presents the limitations and conclusion. Appendix A and Appendix B contain the ACL and the IDEA versions of the projects, respectively. Also available through restricted access are Appendix C through Appendix F. Appendix C contains a detailed teaching note, while Appendix D provides a sample of ACL exam questions with solutions. Appendix E and Appendix F present the solutions to the ACL projects in terms of the logs and reports, respectively.

PROJECT DESCRIPTIONS

Introduction to ACL Project

All students start by individually completing the Introduction to ACL Project. This project requires students to work through the ACL tutorial, which is a PDF file packaged with the ACL software. The ACL tutorial covers the basics of ACL and familiarizes students with most of the Analyze menu and Data menu commands. The tutorial also familiarizes students with ACL's report features. As they work through the tutorial, students should begin to understand the value of having a log of the work performed.[3] Our Introduction to ACL Project, provided in Appendix A, extends the basic requirements by asking students to format and print a report, add a comment to the log that includes their name, and print the log of their activity.

Project 1: Revenue Cycle Issues

The first ACL project relates to issues that arise in the revenue cycle. The objectives of this project are for students to: (1) gain confidence in working with ACL by analyzing accounts receivable using some of the features introduced in the tutorial (e.g., stratify, filter, summarize, reports, logs), (2) relate data tables, and (3) exercise judgment to determine which accounts to confirm. Students analyze the accounts receivable transaction data and prepare an aging of accounts receivable using the Sample Project file located on the ACL CD. Students must decide which accounts to confirm and justify their selections in the log. Next, students

[3] The projects are written for ACL Version 9, which is most often packaged with accounting textbooks. However, faculty can contact the authors for the assignments and solutions associated with ACL Version 8 or earlier.

establish the relationship between the accounts receivable data (which contains account balances) and the customer data (where the mailing information resides). The handout Linking Flat Files Using ACL's Index and Relations Commands, contained at the end of Appendix C, explains how to link two flat files. Building the relationship allows fields from both tables to be printed in a report. Students also review the report, comment on any unusual items, and export the necessary fields to a Word mail merge file. In the final phase of the project, students prepare confirmation letters using the data exported from ACL, which helps students become more familiar with the accounts receivable audit process.

Project 2: Expenditure Cycle Issues

The second project uses the inventory, vendor, and accounts payable data. The primary objectives of this project are for students to: (1) use filters, computed fields, and extractions, (2) relate data tables, and (3) identify unusual inventory items. Specifically, students explore the inventory purchases, note unusual items discovered, and prepare several reports. For example, students create reports where the unit cost is higher than the sales price and vendor invoices are greater than $5,000. Students must also extract a portion of the inventory data into a new, smaller file. Students add comments to the log describing the purpose of the batch totals created during the extraction process and a situation where extracting data may be useful to an auditor.

Project 3: Payroll Cycle Issues

This project involves auditing payroll records to identify situations or transactions that may require further investigation. The primary objectives for this project are for students to: (1) use filters, expressions, and computed fields, (2) relate data tables, (3) examine payroll for accuracy, segregation of duties, and fraud, and (4) use judgment to identify anomalies which require further investigation. Specifically, students must build appropriate relationships between the payroll transaction file and the employee master file to prepare several reports. Students also use the Duplicates command to search for related employees as well as the possibility of duplicate paychecks. Students write brief explanations of any issues identified and describe what further analysis is necessary given the evidence collected.

Project 4: Importing Data

The objective of the final project is for students to learn how to import a client's database into ACL. While this project is the most tedious, it clarifies for students the importance of making sure the database arrives in ACL intact. Students create several views of the imported files and print related reports. Students also have an opportunity to review some of the additional documentation features available in ACL.

METHODOLOGY

Implementation

The ACL projects were administered in Master's-level auditing courses at two regional state universities located in the southeastern United States. One university, University FT, has a full-time Master of Accountancy program while the other, University PT, has a part-time program. Students at University FT completed the ACL projects in a 7-week, 1.5 semester-hour elective course, Systems Assurance, which focused on various IT auditing issues. Prerequisites for Systems Assurance included a traditional undergraduate auditing and a graduate risk-based auditing course. The instructor lectured briefly on CAATs and ACL (approximately 2 lecture hours) and then distributed the projects as an out-of-class assignment for students to complete without

further instruction. Students worked in self-formed groups of two or three students. Altogether, the projects were worth 30 percent of each student's course grade.

Students at University PT completed the ACL projects as part of the required graduate auditing course, a 15-week, 3 semester-hour course. This course was the only auditing course available in the Master of Accountancy program. The prerequisite for the course was an accounting degree that included one undergraduate auditing class. The instructor followed lectures on CAATs (4.5 lecture hours) with computer lab time demonstrating ACL (3 lab hours). Students, working individually, started the ACL projects during the labs and then completed the projects on their own. Combined, the ACL projects accounted for 20 percent of each student's grade. In addition, ACL problems were 35 percent of the mid-term exam grade, which was worth another 15 percent of the course grade.

Data Collection

We first pilot tested the materials for two semesters making necessary adjustments. Then during the third semester of use, we evaluated the perceived learning benefits of these projects by collecting demographic, skill assessment, and perception data from the 31 students completing the two courses. We captured demographic and skill assessment data through an identical pre- and post-survey. A pre-survey was administered just prior to students starting the Introduction to ACL Project and a post-survey immediately after students submitted the final ACL project approximately six weeks later. Upon completion of the post-survey, we collected student perceptions regarding the ACL projects in a separate survey.

RESULTS

Student Demographics

Table 2 presents student demographics. Over 90 percent of the students in the two classes were between the ages of 20 and 27. Many of the students had accounting work experience (1-2 years = 51 percent, 3-5 years = 10 percent), although only a few students had any auditing experience (None = 77 percent). The sample is divided almost evenly between males and females and between the two universities. Students at both universities reported fairly similar technology comfort levels (approximately 7.0 out of 10).

Skill Assessment

To test for differences between the two student groups, we use t-tests to compare the pre-test means for the skill assessment and CAAT perception data. The (relatively) high pre-test means seen in Table 3 are not surprising given that the graduate students had already completed an undergraduate auditing course as well as an accounting information systems class. Only four variables yield significant differences in pre-test means between the two student groups.[4] Students at University FT have a higher self assessment of skills related to creating an accounts receivable aging (p-value = 0.0409). In contrast, students at University PT profess more interest in using CAATs (p-value = 0.0061), more agreement that knowledge of CAATs is important (p-value = 0.0285), and more agreement with the use of CAATs to document audit work (p-value = 0.0018). These pre-test differences appear to be consistent with each instructor's emphasis during coursework prior to beginning

[4] Because some of the variables were skewed and failed to satisfy the normality assumption of a standard t-test, all comparisons were also made using the Wilcoxon rank sum test. The Wilcoxon rank sum test is the nonparametric equivalent of the t-test and does not assume normality of data. No notable differences were evident in our results.

the ACL projects.

TABLE 2
Student Demographics

Panel A: Age		**Ranges**			
	N	**20-23**	**24-27**	**28-31**	**32-40**
University FT	16	44%	50%	6%	---
University PT	15	86%	7%	---	7%
Overall	31	65%	29%	3%	3%

Panel B: Work Experience (in years)		**Ranges**		
	N	**None**	**1-2 yrs.**	**3-5 yrs.**
Accounting – University FT	16	50%	50%	---
Accounting – University PT	15	27%	53%	20%
Overall	31	39%	51%	10%
Auditing – University FT	16	94%	6%	---
Auditing – University PT	15	60%	40%	---
Overall	31	77%	23%	---

Panel C: Gender	**N**	**Males**	**Females**
University FT	16	10	6
University PT	15	7	8
Overall	31	17	14

Panel D: Technology Comfort Level				**(Scale: 1=Low, 10 = High)**	
	N	**Mean**	**Std Dev**	**Min**	**Max**
University FT	16	7.31	1.14	5	9
University PT	15	7.07	1.94	3	10
Overall	26	7.19	1.56	3	10

We also evaluate whether students' (self-reported) ability to use GAS improved using one-tailed t-tests that compare the pre-test and post-test results for the skill assessment and CAAT perception data. We separated the survey into three sections: (1) Accounts Receivable, covered in the revenue cycle project (Project 1); (2) Command Log, covered in all projects; and (3) Database Fundamentals, covered in the expenditure cycle, payroll cycle, and importing projects (Projects 2, 3, and 4). As shown in Table 3 Panel A, students believe that their skills improved between the pre-test and post-test for all items included in the survey related to the accounts receivable, command log, and database fundamentals (all p-values < 0.01). Table 3 Panel B shows that student perceptions regarding CAATs also improve in almost every area. The most significant improvements occur in the items related to confidence using ACL, the student's skill level using CAATs, and understanding how CAATs are used by auditors (all p-values < 0.0001). Marginal results are found regarding the importance of CAATs in an auditing career (p-value = 0.0732) and improving audit efficiency (p-value = 0.0961). The only area that does not show significant improvement is the assessment of student interest in using CAATs (p-value = 0.2318). Overall, however, the results suggest that students perceived that completing the ACL projects helped them acquire significant skills as well as a better understanding of how to use CAATs.

TABLE 3
One-Tailed t-Test Results for Skill Assessments and Perceptions of CAATs

Panel A: Skill Assessments (Scale: 1 = totally uncomfortable to 7 = extremely comfortable)

Description	Pre-test Mean (sd)	Post-test Mean (sd)	p-value
Accounts Receivable			
I can identify the content and format of an account receivable confirmation	4.54 (1.61)	6.07 (0.81)	< 0.0001
I know how to create an accounts receivable aging	4.23 (1.86)	6.43 (0.79)	< 0.0001
I know how to develop an account receivable confirmation	3.88 (1.80)	5.82 (0.86)	< 0.0001
I can appropriately identify specific accounts that should be selected for confirmation	3.50 (1.39)	5.43 (1.10)	< 0.0001
Command Logs			
I understand the importance of command logs as evidence trails	5.69 (1.67)	6.64 (0.87)	0.0067
I understand the purpose of command logs	5.50 (1.82)	6.86 (0.36)	0.0004
Command logs allow me to document the audit tests performed	5.42 (1.70)	6.64 (0.68)	0.0009
Database Fundamentals			
I am confident in my ability to download client data files	3.81 (1.86)	5.46 (1.20)	0.0002
I understand data relationships	4.31 (1.29)	5.54 (1.10)	0.0002
Filters allow me to analyze specific data that fits my audit needs	4.58 (1.56)	6.50 (0.64)	< 0.0001

Panel B: Perceptions of CAATs (Scale: 1 = strongly disagree to 7 = strongly agree)

Description	Pre-test Mean (sd)	Post-test Mean (sd)	p-value
I am confident in my skills related to using CAATs such as ACL	3.69 (1.49)	5.29 (1.08)	< 0.0001
I am interested in using CAATs	5.58 (1.41)	5.82 (0.94)	0.2318
I have strong skills related to using CAATs	3.12 (1.24)	4.89 (1.26)	< 0.0001
I understand how CAATs are used in the audit profession	4.69 (1.19)	6.07 (0.81)	< 0.0001
Knowledge of CAATs is important for a successful career in auditing	5.69 (1.41)	6.18 (0.94)	0.0732
CAATs can help improve the efficiency of an audit	6.23 (1.03)	6.54 (0.58)	0.0961
CAATs enable auditors to use/analyze client data files	6.08 (0.98)	6.61 (0.57)	0.0102
CAATs are useful in analyzing accounts receivable	5.81 (1.17)	6.39 (0.69)	0.0159
CAATs improve accounts receivable analysis	5.81 (1.10)	6.43 (0.63)	0.0079
CAATs can document audit tests performed	5.77 (1.02)	6.33 (0.59)	0.0186

We also analyze post-test skill assessment data by school (table not shown). An interesting finding is that all pre-test differences between student groups disappear in the post-test, except for the variable dealing with skills needed to develop accounts receivable confirmations, where University FT students rated themselves higher than University PT students (p-value = 0.0130). The similarity of post-test results suggests that ACL projects can improve perceived knowledge of GAS regardless of student background, course format (group versus individual projects), or teaching method (self-study versus lab-time). Hence, we believe that the results indicate that ACL projects can be used successfully to enhance learning of auditing and CAATs and that there is great flexibility in how the projects can be implemented. The benefit of these projects appears to come from students' hands-on interaction with the software rather than through course format or method of instructional delivery.

Student Perceptions Concerning the Projects

Students also responded to several questions regarding their perceptions about the project experience using an 11-point Likert scale, with 0 = strongly disagree, 5 = indifferent, and 10 = strongly agree. Table 4 summarizes these responses and shows that students tended to agree or strongly agree with most statements. Two highly rated items relate to the benefits of using ACL to better understand CAATs and GAS, with mean scores of 8.19 and 8.25, respectively. The students also indicated that the projects were quite challenging (mean = 8.86). However, further analysis reveals that students in groups did not find the projects as challenging as students working individually. This result is not surprising given the benefit of collaboration in problem solving. Students at University FT rated group dynamics quite high (mean = 8.50). All other statements received high agreement (mean > 7.00), except for the statement indicating student enjoyment of the ACL projects (mean = 5.11). These results suggest that while students perceive benefit from working with ACL and like working in groups, they do not necessarily enjoy the experience of completing the ACL projects. Despite this lukewarm "enjoy" rating, students found the projects challenging, interesting, and knowledge enhancing.

TABLE 4
Student Perceptions of ACL Projects

Students rated the following items on a scale from 0 = strongly disagree to 5 = indifferent to 10 = strongly agree.

Perception	N	Mean	Std Dev	Min	Max
The ACL Projects were beneficial to my auditing education	28	7.64	1.75	3	10
The ACL Projects helped me to better understand CAATs	27	8.19	1.84	2	10
The ACL Projects helped me to better visualize specific audit procedures for the revenue and expenditure cycles	28	7.82	1.52	5	10
I learned more working on the ACL Projects than I would have in an in-class lecture environment	28	7.46	2.32	1	10
The ACL Projects increased my knowledge of generalized audit software tools	28	8.25	2.03	0	10
The ACL Projects were interesting	28	7.21	1.81	4	10
The ACL Projects were challenging	28	8.86	1.46	5	10
I enjoyed completing the ACL Projects	28	5.11	2.67	0	10
My group worked well together on the ACL Projects*	14	8.93	1.38	5	10
My group shared the work fairly equally*	14	8.50	2.03	3	10
I learned more by working in a group than I would have working alone*	14	7.79	2.01	5	10

*Only students at University FT, who completed the ACL projects in groups, responded to this item.

Best and Worst Aspects of the Projects

Students were also asked a series of open-ended questions regarding the best and worst aspects of the projects. Interestingly, the items mentioned were quite similar across both classes despite differences in class structure and teaching style. The most commonly cited "best aspects" of the projects included learning more about how to use ACL software, learning how to audit with the computer using CAATs, and gaining hands-on experience.

The most commonly cited "frustrating aspects" of the assignments included time pressures, lack of sufficient instructions, downloading/technical problems, and lack of a training manual. When asked what aspects of the projects students would change, more detailed instruction was clearly the most common answer

(for both instructors),[5] followed by improved timing of the projects (both in terms of time required and when and how the projects are administered). At least one student working alone suggested that the projects become group projects and at least one person working in a group suggested that the projects require more individual work.

Impact of Gender and Other Factors on Results

We also partition the data along a variety of measures, including gender, age, and technology comfort level, and detected no significant differences. These findings suggest that the results of the study are fairly robust across student groups.

CONCLUSION

In this paper, we present four projects designed to develop students' (1) knowledge of auditing and (2) use of the two dominant GAS programs. The projects are written in a modular format, which allows them to be decomposed into shorter assignments. As a result, the projects are flexible allowing instructors to adapt them to meet specific course objectives.

We also collect student self-assessment data on the projects. Overall, our results suggest that GAS projects enhance student learning. Individual student skill assessments increase with respect to all of the skill assessments. In most cases, students' perceptions of CAATs also improve. Students believe the projects are both challenging and of higher value than a traditional lecture teaching approach, suggesting that these hands-on, active learning GAS projects enhance the classroom experience. In addition, student background, teaching approach, and course format do not appear to influence students' perceptions of knowledge gains.

The data in this study relate to two instructors at two regional public universities. The sample sizes are quite small, the analyses are restricted to only one type of GAS (ACL), and the projects are limited to selected concepts. As such, the results may not be generalizable to other instructors, students, universities, types of software, or audit concepts. However, given the promising results of this study, we encourage researchers to extend this analysis to other universities, instructors, types of GAS, audit concepts, and audit techniques. For instance, future studies might include sampling approaches or an application of Benford's Law. ACL and IDEA can also be integrated into a non-audit course, such as an existing database course, so that students can design databases and then audit them. Anecdotal evidence suggests that ACL and IDEA skills are valued by accounting firms when selecting new hires, but this contention has not been tested empirically. Another question researchers could investigate is how experience with one GAS affects the learning process with similar software packages.

REFERENCES

Accounting Education Change Commission (AECC). 1990. *Objectives of Education for Accountants*. Position Statement No. 1. Torrance, CA: AECC.

ACL Services Ltd. (ACL). 2008. *Company*. March 10. Available at http://www.acl.com/company.

Albrecht, W. S., and R. J. Sack. 2001. The Perilous Future of Accounting Education. *The CPA Journal* (March): 16-23.

American Institute of Certified Public Accountants (AICPA). 1999. *Focus on the Horizon, the CPA Profession in 2011*. New York: AICPA.

[5] The assignments include some instructions, but do not contain screen shots or repeated details describing how to execute the ACL or IDEA commands. Instead, we designed the projects so that students had to think about the appropriate steps needed to accomplish a task. Hence, the desire for further detailed instructions was not surprising.

____. 2008. *Uniform CPA Examination: Examination Content Specification Outlines with References Updated October19, 2005*. March 10. Available at http://www.cpa-exam.org/cpa/how_prepare.html.

Arens, A. A. 2004. *Computerized Auditing using ACL Software: Instructions & Assignments*. 1st ed. Okemos, MI: Armond Dalton Publishers, Inc.

Boynton, W. C., and R.N. Johnson. 2005. *Modern Auditing*: *Assurance Services and the Integrity of Financial Reporting*. 8th ed. Hoboken, NJ: John Wiley & Sons, Inc.

Bryant, S. M., and J. E. Hunton. 2000. The Use of Technology in the Delivery of Instruction: Implications for Accounting Educators and Education Researchers. *Issues in Accounting Education* 15(1): 129-162.

Chapman, C. 2002. Power Tools: 2002 Audit Software Usage Survey. *Internal Auditor* (August): 28-38.

Gelinas, U. J., E. S. Levy, and J. C. Thibodeau. 2001. Norwood Office Supplies, Inc.: A Teaching Case to Integrate Computer-Assisted Auditing Techniques into the Auditing Course. *Issues in Accounting Education* 16(4): 637-662.

Gray, G. L. 2006. An Array of Technology Tools. *Internal Auditor* (August): 56-62.

Hall, J. A., and T. Singleton. 2005. *Information Technology Auditing and Assurance*. 2nd ed. Mason, OH: South-Western.

Hunton, J. E., S. M. Bryant, and N. A. Bagranoff. 2004. *Core Concepts of Information Technology Auditing*. Hoboken, NJ: John Wiley & Sons, Inc.

Jackson, R. A. 2004. Get the Most out of Audit Tools. *Internal Auditor* (August): 36-47.

Kroenke, D. M. 2000. *Database Processing: Fundamentals, Design & Implementation*. 7th ed. Upper Saddle River, NJ: Prentice-Hall, Inc.

Lehman, M. W., and M. Weidenmier Watson. 2007. Case Study: Detecting Shell Companies with Dynamic Text Comparisons. *Journal of Forensic Accounting* 8(1/2): 347-368.

McCombs, G. B., and M. Sharifi. 2004. Utilization of Generalized Audit Software in an Information Systems Auditing Course. *Information Systems Control Journal* 6: 37-38.

Messier, W. F., S. M. Glover, and D. F. Prawitt. 2008. *Auditing & Assurance Services: A Systematic Approach*. 6th ed. New York: McGraw-Hill.

Nieschwietz, R., K. Pany, and J. Zhang. 2002. Auditing with Technology: Using Generalized Audit Software in the Classroom. *Journal of Accounting Education* 20: 307-329.

Perry, J. T., and G. P. Schneider. 2001. *Building Accounting Systems Using Access 2000*. Cincinnati: South-Western College Publishing.

Rittenberg, L. E., B. J. Schwieger, and K. M. Johnstone. 2008. *Auditing: A Business Risk Approach*. 6th ed. Mason, OH: South-Western.

Splettstoesser, I. B. 1999. Information Systems Controls and Auditing: Mathra, Inc. *Issues in Accounting Education* 14(2): 285-303.

Thompson, A., R. Simonson, and C. Hargrave. 1992. *Educational Technology: A Review of the Research*. Washington, DC: Association for Educational Communications and Technology.

Watson, R. T. 1999. *Data Management: Databases and Organizations*. 2nd ed. New York: John Wiley & Sons, Inc.

Weidenmier, M. L., and T. L. Herron. 2004. Selecting an Audit Software Package for Classroom Use. *Journal of Information Systems* 18(1): 95-110.

APPENDIX A

INTRODUCTION TO ACL PROJECT

This project introduces you to the leading generalized audit software package, Audit Command Language (ACL). Internal and external auditors use ACL to improve the efficiency and effectiveness of their audits. The skills you learn in this tutorial will be used in subsequent projects assigned throughout the course.

First, install ACL using the CD included in your course materials. Next, open the PDF file named ACL in Practice or ACL Tutor, which is available in the ACL folder available on the All Programs menu. Your assignment is to complete this tutorial, which means reading the entire tutorial and completing all of the steps that you are asked to perform (except for the Exercises located at the end of each chapter). The tutorial covers the basics of ACL and walks you through many of the commands found in the Analyze menu and the Data menu. The tutorial also introduces ACL's report features.

Begin the tutorial by saving the necessary data files to a USB drive. This step allows you to exit ACL during the tutorial and continue at a later time. First, create a folder on your USB drive called ACL Tutorial. Next, access the ACL data files located at C:\ACL Data\Sample Data Files. Copy the files listed below to the ACL Tutorial folder on your USB drive.

- Agents_Metaphor.fil
- Employee_List.fil
- Metaphor_Employee_Data

When you open the ACL program, you can access the Metaphor_Employee_Data file from the main screen by selecting Open an Existing Project and browsing for the file on your USB drive. When the tutorial asks you to import additional files, remember to access these files at C:\ACL Data\Sample Data Files and to save your work on your USB drive.

As you work through the tutorial, ACL creates a log of your activities. ACL records all of the activity until the project is closed. To view the log, select the Log tab at the bottom of the Project Navigator window. Some versions of ACL also include XML scripting in the log. Turn off this feature by selecting ***Tools→Options***. On the Command tab, make sure the box is checked next to Suppress XML Script for Command Results.

Additional requirements relating to the tutorial:

1. Somewhere during the tutorial (any place at random), add a comment to the log. To add a comment, select ***Tools→Add Comments*** or simply select the ***Comment*** button on the toolbar. Be creative in your comment, and make sure to mention your name.
2. During the tutorial when you prepare a report, print your results. Modify the report's header so that it includes your name. Use the ***Change Font*** button on the toolbar to set the header's font to size 16, Bold Italic. Print the report using a landscape orientation.
3. At the end of the tutorial, print the contents of the log. Before you can print the log, it must be open. To print the log, select ***File→Print Project Contents*** and select the log.

Required: You must turn in the log for grading, as well as any items that you were asked to print during the tutorial.

ACL PROJECT 1: REVENUE CYCLE ISSUES

The primary goal of this project is to export selected accounts receivable records for confirmation. Confirmation is an important tool auditors use to gather evidence regarding the existence and valuation of accounts receivable as reported on the balance sheet. In this project, confirmation preparation will help you become more familiar with ACL and the accounts receivable audit process. To prepare the confirmation letters, you will need to transfer the customers' mailing address information and their outstanding balance as of your client's year-end to word processing software.

Step 1: Get Started

1. Begin by saving the necessary files to a USB drive. This step allows you to exit ACL before you complete the assignment and continue at a later time.
2. Create a folder on your USB drive using your name and the word Revenue (i.e., SmithS-Revenue).
3. Navigate to the following folder C:\ACL Data\Sample Data Files.
4. Copy the files listed below to the folder you just created.
    - AR.fil
    - Customer.fil
    - Sample Project
    - Trans.fil
5. Start ACL and open the Sample Project file located on your USB drive.
6. Clear the command log (Hint: Right click inside the Project Navigator window and select ***Delete Entire Log***). The log is a record of audit findings. Therefore, when you are asked to comment on or describe your results, record your answers as a comment in the log. Insert a comment in the log that includes your name (Hint: ***Tools→Add Comment***). Place all required comments in the log.

Step 2: Examine Sales and Accounts Receivable Data

1. Open the 'Trans' table, which is located in the Accounts_Receivable_Audit folder. This table shows all of the sales transactions with customers. Review the table layout for the 'Trans' table. (Hint: ***Edit→Table Layout***). Notice that the actual field names often differ from the captions visible in the table view.
2. Generate statistics on the Invoice Amount and Quantity fields (Hint: ***Analyze→Statistical→Statistics***). Comment on your findings.
3. Stratify the 'Trans' table into 10 strata on the Invoice Amount field. Review the strata. Comment in the log on your findings. Which invoices would you want to investigate further?
4. Create a filter that screens for zero or negative dollar amounts in the Invoice Amount field. Name and save the filter. What do the results say about the company's controls over invoice entry? Add a comment to the log.
5. Open the 'AR' table, which shows the dollar amounts for each transaction that affects accounts receivable. Generate statistics about the data. Comment in the log on these results.
6. The Trans Type field consists of codes describing the type of transaction. The following codes have been used: CN = Credit note (a.k.a., a credit memo), IN = Invoice, PM = Payment made by a customer and TR = Transfer (a.k.a., a write-off). Since invoices increase the balance in accounts receivable, they should be positive. All the remaining types of transactions decrease accounts receivable and should be negative. Based on this coding scheme, create a filter that screens for inappropriate customer payments. Name and save the filter. Count the number of inappropriate payments (Hint: ***Analyze→Count***). Comment on your findings.
7. Use the ***Summarize*** command to determine the total amount owed by each customer. This command allows you to total a numeric field for each unique value in a character field. In this case, you want to compute the total amount owed (Amount is a numeric field) for each customer number (Cust Number is a text field). However, the ***Summarize*** command only works with sorted data. For both of the following steps, make sure that the "Use Output Table" box is checked and that you name the output table.
    a. Sort the 'AR' table using the Cust Number field. (Hint: ***Data→Sort Records).***
    b. Summarize the sorted 'AR' table on the Cust Number field and accumulate on the Trans Amount field. As part of the audit, auditors will reconcile these computed balances to the client's accounts receivable records before proceeding with the confirmation process.
    c. Determine the maximum and minimum values of the Amount field. (Hint: ***Analyze→Profile***).
    d. Stratify the total amount owed by customers.
8. Use the ***Age*** function to create an aging of the invoices outstanding as of 12/31/00. Assume that you want to age the invoices according to the following categories: 0-30 days, 30-45 days, 45-60 days, 60-90 days, 90-120 days, 120-180 days, and over 180 days. Briefly describe your results and explain the benefit of this tool on an audit engagement.
9. Given the evidence you have collected, write a memo in the log describing which accounts receivable you would confirm along with an explanation for your choice.

Step 3: Collect the Customers' Information

The data needed for confirmation letters resides in several different tables. A view that combines fields from multiple tables can be created by establishing a relationship between the tables. To build a relationship, the tables must first be sorted on a common field (i.e., primary key and foreign key combination). You can use ACL's ***Index*** feature to logically sort a table on a selected field (e.g., primary key). Then, use the ***Relation*** feature to identify the rows where the fields in the two tables match.

1. Review the fields found in the 'AR' table and the 'Customer' table. Identify the fields from these tables that are necessary to create confirmation letters. The confirmation letters require data about customers and their outstanding balances. Note: In the prior analysis, you saved several variations of the accounts receivable table. To create the confirmation letters, you must decide which one to combine with the customer data.
2. Identify one table as the parent and the other as the child. (Hint: This decision depends on the cardinality of the relationship. Refer to the handout Linking Flat Files Using ACL's Index and Relation Commands.)
3. The first step to creating the relationship between tables is to index the child table.
   a. Open the child table and select ***Data→Create Index***.
   b. Highlight the field to use for the index, in this case the table's primary key.
   c. In the area labeled "To…", name the index using your first name and "custno." Make sure the "Use Output Table" box is checked.
4. Once an index is created, you can establish the relationship between tables.
   a. Open the parent table, then select the ***Relations*** button from the tool bar (or ***Data→Relate Tables***).
   b. If the Getting Started Window appears, click OK.
   c. If the tables you want to relate are not visible, select Add Table in the Relations window. Identify the child table you want to build a relationship with and click OK.
   d. Highlight the key field in the parent table and drag the cursor to the same key field in the child table. A line pointing from the parent table to the child table should appear. Click Finish.

Step 4: Working with Related Tables

Using a table that is related to other tables is simple. Whenever the "Available Fields" dialog box appears, you can access fields from any of the related tables. First, select a table from the "From Table" drop-down list. Then all the fields it contains appear in the "Available Fields" dialog box. Note, however, that only records in the parent table will be processed. There may be records in the indexed table that do not appear in your view, as they do not match records in the parent table.

1. With the parent table open, create a new view by selecting ***File→Save As*** and entering a name for the new table that includes your name and 'confirmdata'.
2. Create a confirmation view of the data by adding any additional fields from the child table into the new view. Examine your new view, which should be limited to only the fields used in the confirmation letters.
3. Edit your table according to the following instructions. Make sure the field format for the Trans Amount field reports negative values as (1,000.00). Format the fields, titles, subtotals, and footer to Times New Roman, size 10 font. The header should be formatted as Times New Roman, size 14 font, Bold Italic. Make sure the full field contents are visible.
4. Print a report of the confirmation data with a landscape orientation. The report should include a heading that describes the content of the report along with your name. Review the printout. If you note any unusual item(s), write a brief comment in the log describing each.
5. Export the confirmation view as a Word merge file to your USB drive. Only include the fields needed for the confirmation letters. Name the output as 'confirm.'

Step 5: Create the Confirmation Letters

1. Write a confirmation letter template in Word using the reference(s) provided by your instructor. You will create a mail merge between this document and the confirmation data you exported from ACL.
2. Begin by opening the confirmation letter template. Under the ***Mailings*** tab click on the ***Start Mail Merge*** icon and choose the Step by Step Mail Merge Wizard.
   a. Select Letters as the document type and use the current document as the starting document.
   b. Identify the confirm.doc file exported from ACL as the existing list of recipients.
   c. Edit the form letter to include the fields needed from the ACL confirmation data. Use the address block feature (match the address fields if needed) and insert the Amount field in the appropriate place in the confirmation letter template.
   d. Complete the mail merge and select Edit Individual Letters. Save the new document that contains all the merged confirmation letters.
3. Print the third confirmation letter and sign it as if you are the client requesting the confirmation on behalf of the external auditors.

***Required*:** **Turn in the confirmation data report, confirmation letter, your log with comments and a zip file of your work.**

ACL PROJECT 2: EXPENDITURE CYCLE ISSUES

In this project, you will work with data in the expenditure cycle, including accounts payable transactions, the vendor master data, and the inventory master data. The goal is to work with expressions and filters, establish relationships between flat files, and extract data from one table into another smaller one. A flat file is simply a database table without any of its relationships. After importing data into ACL, the relationships will be restored.

Step 1: Get Started

1. Begin by saving the necessary files to a USB drive. This step allows you to exit ACL before you complete the assignment and continue at a later time.
2. Create a folder on your USB drive using your name and the word Expenditure (i.e., SmithS-Expenditure).
3. Navigate to the following folder C:\ACL Data\Sample Data Files.
4. Copy the files listed below to the folder you just created.
    - Ap_Trans.fil
    - Inventory.fil
    - Sample Project
    - Vendor.fil
5. Start ACL and open the Sample Project file located on your USB drive.
6. Clear the command log (Hint: Right click inside the Project Navigator window and select ***Delete Entire Log***). The log is a record of audit findings. Therefore, when you are asked to comment on or describe your results, record your answers as a comment in the log. Insert a comment in the log that includes your name (Hint: ***Tools→Add Comment***). Place all required comments in the log.

Step 2: Using Expressions and Filters

Expressions can be used to define new fields, or as part of a filter to limit the records presented in a view. They can be a combination of fields (both data and/or computed), functions, and operators.

Add a Computed Field to a Table

1. Open the default view of the 'Inventory', which is located in the Inventory Review folder.
    a. Select the ***Add Columns*** button.
    b. Click on the 'Expr …' button and write an expression to compute the unit markup percentage. Name the expression Markup.
    c. Add the Markup field to your 'Inventory' table view.

Working with Command Filters

A filter is a special type of expression used to identify records that meet certain criteria. A command filter is applied when a single command is executed for one occurrence only. A global filter is applied to all views of a table and stays in effect until it is removed.

2. With the 'Inventory' table open, select ***Analyze→Count Records***.
    a. Select "If. …" to open the expression builder.
    b. Write a filter that identifies inventory items with a negative total cost.
    c. Name the filter so that it can be used again and click ***OK*** to apply the filter.
3. Use a filter to count the inventory items with a negative or zero quantity on hand. Save the filter.
4. Use a filter to count the inventory items that have a sale price lower than unit cost. Save the filter.

Working with Global Filters

5. To examine the specific inventory items that have a unit cost higher than their sales price, apply your saved filter as a global filter by simply selecting the function button at the top of the view. In the Fields box, select the appropriate filter and then ***OK***.
6. Prepare a report of these items. Give the report a descriptive title that includes your name.
    a. Delete these fields from the view: Location, Product Class, and Market Value.
    b. Adjust the remaining columns so that each field's entire contents are visible.
    c. Print and save the view.

Step 3: Recreating Relationships from Flat Files

To analyze inventory transactions with vendors requires a comparison of data in the 'AP_Trans', 'Inventory', and 'Vendor' tables. A view that combines fields from multiple tables can be created by establishing a relationship between the tables. To build a relationship, the tables must first be sorted on a common field (i.e., primary key and foreign key combination). You can use ACL's ***Index*** feature to logically sort a table on a selected field (e.g., primary key). Then, use the ***Relation*** feature to identify the rows where the fields in the two tables match.

1. Review the tables 'AP_Trans', 'Inventory', and 'Vendor'. Decide which one is the parent table and which are the children tables. (Hint: This decision depends on the cardinality of the relationship. Refer to the handout Linking Flat Files Using ACL's Index and Relation Commands).
2. The first step to creating the relationships between tables is to index one of the child tables.

a. Open one of the child tables and select ***Data→Create Index***.
b. Highlight the field to use for the index, in this case the table's primary key.
c. In the area labeled "To…", give the index a name. Make sure the "Use Output Table" box is checked.

3. Once an index is created, you can establish the relationship between the parent table and the child table.
   a. Open the parent table, then select the ***Relations*** button from the tool bar (or ***Data→Relate Tables***).
      1. If the Getting Started Window appears, click ***OK***.
      2. If the tables you want to relate are not visible, select ***Add Table*** in the Relations window. Identify the child table you want to build a relation with and click ***OK***.
      3. Highlight the key field in the parent table and drag the cursor to the same key field in the child table. A line pointing from the parent table to child table should appear. Click ***Finish***.
4. Now, build the second relationship needed to complete the connection between the three tables.

Step 4: Working with Related Tables

Using a table that is related to other tables is simple. Whenever the "Available Fields" dialog box appears, you can access fields from any of the related tables. First, select a table from the "From Table" drop-down list. Then all the fields it contains appear in the "Available Fields" dialog box. Note, however, that only records in the parent table will be processed. There may be records in the indexed tables that do not appear in your view, as they do not match records in the parent table.

1. With the parent table open, create a new view by selecting ***File→Save As*** and entering a name for the new table.
2. Select the following fields to include in your new view: Vendor Number, Vendor Name, Product Description, Invoice Date, and Invoice Amount.
3. Prepare a report of this new view. Give the report a descriptive title that includes your name.
   a. Limit the report's contents to vendor invoices in excess of $5,000.
   b. Format the entire table using Tahoma with at least a 10 font, double-spaced.
   c. Make sure the field format for the Invoice Amount field reports negative values as (1,000.00).
   d. Print and save the view. Add a comment to the log describing the purpose of this report.

Step 5: Extracting Selected Data

ACL's ***Extract Data*** command allows you to save specific records or fields from a large table into a smaller table. The ***Extract Data*** feature copies a subset of the records or fields and creates a table of the extracted records. Records in the original table remain unchanged. The basic steps for extracting data are as follows: (1) create a filter that identifies the subset of data you want to extract, (2) compute control totals, (3) extract the records to a new table, and (4) compute new control totals and compare them to the original ones.

1. With the 'Inventory' table open, select ***Analyze→Count Records***. The ***Count*** feature returns the total number of records in a table.
   a. Select the "If…" button and limit the count to only those items stored at Location #4.
   b. Save the filter and execute the ***Count*** command.
2. For inventory stored at Location #4, compute control totals for all the numeric fields by selecting the ***Total*** button on the tool bar (or ***Analyze→Total Fields***).
3. Select the ***Data→Extract Data***.
   a. If you want to extract a limited number of fields, select the ***Fields*** radio button. However, if you want all the fields to be extracted, select the ***Record*** radio button.
   b. Limit the fields extracted to the Product Number, Product Description, Quantity on Hand, and the Inventory Value at Cost.
   c. In the area labeled "If …" identify the saved filter for Location #4.
   d. In the area labeled "To…" give the extracted table a name. The "Use Output Table" box should be checked.
4. Verify the control totals by selecting the ***Total*** button on the tool bar (or ***Analyze→Total Fields***).
5. Add a comment to the log to document the results of your comparison. Briefly explain the purpose of this comparison.
6. In the log, briefly describe one situation where extracting data could be useful in an audit.
7. Prepare a report of the extracted table. Give the report a descriptive title that includes your name.
   a. Format the entire table using Tahoma with at least a 10 font.
   b. Make sure the field format for the Quantity on Hand and Inventory Value at Cost fields report negative values as (1,000.00). Remove totals on the Quantity on Hand field. Make sure each field's entire contents are visible.
   c. Print and save the view.

***Required*:** **Turn in any printouts, your log with comments and a zip file of your work.**

ACL PROJECT 3: PAYROLL CYCLE ISSUES

This project focuses on auditing payroll. In particular, you will be looking at payroll records to identify situations or transactions that may require further investigation.

Step 1: Get Started

1. Begin by saving the necessary files to a USB drive. This step allows you to exit ACL before you complete the assignment and continue at a later time.
2. Create a folder on your USB drive using your name and the word Payroll (i.e., SmithS-Payroll).
3. Navigate to the following folder C:\ACL Data\Sample Data Files.
4. Copy the files listed below to the folder you just created.
    - Empmast.fil
    - Payroll.fil
    - Sample Project
    - Work_Depts.fil
5. Start ACL and open the Sample Project file located on your USB drive.
6. Clear the command log (Hint: Right click inside the Project Navigator window and select Delete Entire Log). The log is a record of audit findings. Therefore, when you are asked to comment on or describe your results, record your answers as a comment in the log. Insert a comment in the log that includes your name (Hint: ***Tools→Add Comment***). Place all required comments in the log.

Step 2: Substantive Test of Accuracy

As auditors, we want to ensure the accuracy of the payroll calculations. As a substantive test of accuracy, you will recompute net pay.

1. Open the 'Payroll' table, which is located in the Payroll Analysis folder.
2. Add a new field that recomputes net pay.
    a. Select the ***Add Columns*** button.
    a. Click on the 'Expr …' button and write an expression to recompute net pay. Name the expression.
    b. Add the new field to your 'Payroll' table view.
3. Write a filter to test whether the original net pay field equals your recalculated amount.
4. Add a comment to the log describing the results of this test.

Step 3: Related Employees

As auditors, we are concerned with segregation of duties. To test for segregation of duties, we need to review the organizational chart and job descriptions for conflicting duties. We also should observe employees to determine whether the segregation policy is followed in practice. As an additional test, we may search the employee records for evidence of related employees. To test for related employees, you will isolate employees with the same last name or same address.

1. Open the 'Empmast' table.
2. Create an index using the Last Name field (Hint: ***Data→Create Index)***. Make sure the "Use Output Table" box is not checked. Name and save the index. (NOTE: The data must be sorted before executing the ***Duplicates*** or ***Sequence*** command. An index logically sorts the data, while the ***Sort*** command builds a physical sort of the data, which results in a new table.)
3. Select ***Analyze→Look for Duplicates***. Sequence on the employee's last name. Select the following fields to list on: First Name, Employee Number, Street Address, Job Description, and Gender. Direct your output to the screen and add a header describing the test.
4. Create an index using the employee's address. Make sure the "Use Output Table" box is not checked. Name and save the index.
5. Select ***Analyze→Look for Duplicates***. Sequence on the employee's address. Select the following fields to list on: Employee Number, First Name, Last Name, Job Description, and Gender. Direct your output to the screen and add a header describing the test.
6. Briefly describe in the log what your results suggest and the steps you would take to follow-up on them.

Step 4: Fraud Detection

As auditors, we are concerned about the possibility of fraud. Overpayment of employees and payment to nonexistent employees are two common types of payroll fraud. To test for payroll fraud, you will look for multiple paychecks to the same employee and verify that all checks were paid to valid employees.

Test for Multiple Paychecks

1. Open the 'Payroll' table.
2. Create an index using the Employee Number field. Name and save the index.
3. Select ***Analyze→Look for Duplicates***. Sequence on the Employee Number field. Select the following fields to list on: Check Number, Gross Pay, and Pay Date. Direct your output to the screen and add a header describing the test.

Test for Paychecks to Only Valid Employees

Evaluating whether all the payroll checks were paid to valid employees requires a comparison of data in the 'Empmast' table and 'Payroll' table. A view that combines fields from multiple tables can be created by establishing a relationship between the tables. To build a relationship, the tables must first be sorted on a common field (i.e., primary key and foreign key combination). You can use ACL's ***Index*** feature to logically sort a table on a selected field (e.g., primary key). Then, use the ***Relation*** feature to identify the rows where the fields in the two tables match.

4. Review the tables 'Empmast' and 'Payroll'. Decide which one is the parent table and which one is the child table. (Hint: This decision depends on the cardinality of the relationship. Refer to the handout Linking Flat Files Using ACL's Index and Relation Commands.)
5. The first step to creating the relationship between the tables is to index the child table.
   a. Open the child table and select ***Data→Create Index***.
   b. Highlight the field to use for the index, in this case the table's primary key.
   c. In the area labeled "To…", give the index a name. Make sure the "Use Output Table" box is checked.
6. Once an index is created, you can establish the relationship between the parent table and the child table.
   a. Open the parent table, then select the ***Relations*** button from the tool bar (or ***Data→Relate Tables***).
   b. If the Getting Started Window appears, click OK.
   c. If the tables you want to relate are not visible, select Add Table in the Relations window. Identify the child table you want to build a relation with and click OK.
   d. Highlight the key field in the parent table and drag the cursor to the same key field in the child table. A line pointing from the parent table to child table should appear. Click Finish.

Using a table that is related to other tables is simple. Whenever the "Available Fields" dialog box appears, you can access fields from any of the related tables. First, select a table from the "From Table" drop-down list. Then all the fields it contains appear in the "Available Fields" dialog box.

7. With the parent table open, create a new view by selecting ***File→Save As*** and entering a name for the new view.
8. Add the Employee Number field from the child table to the parent table.
9. Rearrange the fields so that the two Employee Number fields are adjacent. Write a filter that limits the view to records where the Employee Number fields do not agree.
10. Print the view as a report. The report should include a heading that describes the report's contents and your name. Make sure all field contents are visible. Save the view.
11. Describe your findings and possible causes for them in the log.

***Required*:** **Turn in the Fraud report, your log and a zip file of your work.**

ACL PROJECT 4: IMPORTING DATA

The goal of this project is to import a client's database into ACL. The client has supplied you with a copy of their Microsoft Access database, along with an entity-relationship diagram. To import their database into ACL, each table will be converted into a flat file. A flat file is simply a database table without any of its relationships. The relationships will be restored once the files are accessible in ACL.

Step 1: Import the Database into ACL as Flat Files

1. Download the Access database provided by your instructor to your desktop.
2. Open ACL and create a new project using your name and save it to your USB drive.
3. Follow the Data Definition Wizard to create a new table.
4. Select ***File→New→Table*** to import the next table from the database. Continue until all the tables from the database have been imported into ACL.

Step 2: Modify the Tables in ACL

1. Select ***Edit→Table Layout*** and ensure the each table conforms to the following guidelines.
   a. Check the field names to confirm that they are consistent with the original database.
   b. Format the field names using all caps.
   c. Format captions using both upper and lower case. The field name that displays in the ACL view must be consistent with the caption that appears in the database.
   d. Format currency fields as numeric so that negative amounts appear as (1,000.00).
   e. Remove subtotals from numeric fields, except for fields related to quantity or currency fields.
   f. Primary keys should be formatted as ASCII not Numeric.
   g. Use an input mask to format any date fields.
   h. Limit column widths to the size of the data.

Step 3: Print the Tables

1. Open the each table and review your work.
2. Make any modifications needed.
3. Create a report for each table. Add a heading to the report that describes the table and includes your name.
4. Format each entire report as Times New Roman, 10 font. Change the font for the heading to Times New Roman, 14 Bold Italic font.
5. Print the first page of each table. If possible, force the table to print on a single page.

Step 4: Create Additional Views (to be defined by the instructor as indicated by XXX)

1. Open the table for the XXX data.
2. Create the following views: XXX, XXX.
3. Print the additional views. Make sure you follow the printing instructions in Step 3 above.

Step 5: Wrap Up

1. Add a comment to the log that includes your name (Hint: ***Tools→Add Comment***).
2. Print the documentation for all the table layouts and the log. (Hint: Select ***File→Print Project Contents*** and check the appropriate boxes.) Insert a page break after each category.

***Required*: Turn in all of the view printouts, the documentation, your log and a zip file containing your completed project.**

APPENDIX B
IDEA TEACHING NOTES

This appendix highlights changes that must be made to use the projects with IDEA software. Accordingly, each of the projects has been modified to be compatible with IDEA.

Exporting Files from ACL

The ACL project dataset, Sample Project, can be easily adapted for use in IDEA by exporting it into Excel, reformatting the date, saving the file as an Excel spreadsheet, and then importing it into IDEA. The following files should be exported from ACL into Excel format:

- Project 1: AR, Customer, Trans
- Project 2: AP_Trans, Inventory, Vendor
- Project 3: Empmast, Payroll

Each file with a date must have the date field custom formatted in Excel as YYYYMMDD. Rather than having students export the files from ACL, faculty can provide students with the Excel files that accompany this paper. However, transferring the data through Excel gives students additional practice importing and exporting files. The IDEA version of the projects has been written assuming students will import the files from Excel. Another option is to use a plug-in available at www.caseware-idea.com to directly import ACL files into IDEA.

Noteworthy Differences between ACL and IDEA

If students work the projects in both ACL and IDEA, they should notice several differences. Some of the more obvious differences between the two software packages are identified below.

- The Log: ACL records events in the log for the entire project, while IDEA creates a history for each database. Consequently, students will print a single log in ACL; but using IDEA, they will need to print a History log for each database in the project. Comments are also stored separately in IDEA and need to be printed individually for each database.
- Project 1, Step 2-3: When stratifying data, ACL will automatically create a specified number of intervals of the same size. In IDEA, the interval size must be specified.
- Project 1, Step 2-7: Before summarizing data, ACL requires that the data be sorted, while IDEA does not.
- Project 1, Step 2-8: The first date range in the ACL Age command covers 0-29 days old. IDEA treats zero as a separate line item. To get an answer comparable to ACL, the first two rows of the IDEA output need to be combined.
- Project 1, Step 2-8: ACL excludes the endpoints used in the date ranges from the output. For example, a cutoff of 30 days generates a period of 0-29 days. In IDEA, a cutoff of 30 days results in a period from 1-30 days. The IDEA project in this appendix adjusts the endpoint dates so that the results will match the ACL solution log provided in Appendix E.
- Project 1, Step 3: ACL requires data to be indexed before joining tables, IDEA does not.
- Project 2, Step 2: In ACL, the filters appear in the log and can be graded. IDEA does not show Criteria in the History log. To be able to grade Criteria, instructors can review the individual zip files, ask student's to add a Comment describing the filtered results, or have student's prepare a report of the

filtered result. The Project asks students to add a Comment. Alternatively, an Extraction creates a new database that is recorded in the History log. Faculty can change the instructions to explicitly use Extractions if they want the results recorded in the History log.

- Project 2, Step 2-5 & 2-6: ACL differentiates between command filters and global filters. IDEA's Criteria are similar to command filters, but there is no parallel to global filters. In the IDEA projects, we use Direct Extractions in place of global filters. Given that Direct Extractions also appear in Step 5 of the Project, Direct Extractions are covered twice in the IDEA version.
- Project 2, Step 4: ACL allows reports to be double-spaced, but IDEA does not.
- Project4: ACL imports tables individually, while IDEA imports the entire database at once (assuming all tables are selected).

Introduction to IDEA Project

Several tutorial options exist. Students beginning the projects in IDEA will need a more detailed tutorial experience, such as the *IDEA Version 7 Workbook* (272 pages). The Workbook accompanies the educational version of IDEA. Students that have already completed an ACL tutorial should be able to use one of the abbreviated tutorials, such as the *IDEA Version 7 Quickstart Guide* (41 pages), *IDEA Version 7 Case Study* (43 pages) or the *IDEA 2004 Introduction Guide* (36 pages). All three resources can be downloaded for free at http://www.audimation.com/self-study.cfm.

As in ACL, IDEA creates a History log of all activities. The History log can be viewed by clicking the Database section of the Properties window followed by the History link. An expanded view of the History log can be seen by clicking on the ***Expand All Details*** button. To print the contents of the History log, make sure that the History view is visible and select ***File→Print→Print***. To add a comment in the comments section, select ***Add Comment*** under the ***Comment Heading***. To print comments for a specific database, (1) select the desired file/database in the IDEA Files window, (2) click on Comment in the Comments section of the Properties window, and (3) click the Print button in the Database comments window. Note that comments print separately from the History log.

IDEA PROJECT 1: REVENUE CYCLE ISSUES

The primary goal of this project is to export selected accounts receivable records for confirmation. Confirmation is an important tool auditors use to gather evidence regarding the existence and valuation of accounts receivable as reported on the balance sheet. In this project, confirmation preparation will help you become more familiar with IDEA and the accounts receivable audit process. To prepare the confirmation letters, you will need to transfer the customers' mailing address information and their outstanding balance as of your client's year-end to word processing software.

Step 1: Get Started

1. Open IDEA and select ***File→Set Working Folder*** to specify the USB drive where you want the data files to be stored. Navigate to a folder that contains your name. Click ***Open***. In the Project Properties window, enter the project name as Accounts Receivable – Your Name with a period of Jan. 1, 2000 to Dec. 31, 2000. Click ***OK***.
2. Use the following steps to import the 'AR', 'Customer', and 'Trans' Excel files into IDEA.
   a. Use the ***Import Assistant*** button (or ***File→Import Assistant→Import to IDEA***) to begin importing a file.
   b. In the Import Assistant window, highlight Microsoft Excel. Click Browse (…) by the Filename and find the desired Excel file. Highlight the file and then click ***Open***. Then click ***Next***.
   c. In the Excel window, click the First Row is field names box. Make sure the data is formatted properly. Click ***OK*** to import the Excel worksheet.
3. The History log is a record of audit findings conducted using IDEA. The History log cannot be altered. Additional comments can be documented in a separate comments file. Therefore, when you are asked to comment on or describe your results, record your answers as a comment. Insert a comment that includes your name (Hint: In the Comments section of the Properties window, select ***Add Comment***). Place all required comments in this Comments file.

Step 2: Examine Sales and Accounts Receivable Data

1. Open the 'Trans' database. This database shows all of the sales transactions with customers. View the statistics on the Invoice Amount and Quantity fields (Hint: ***Field Stats***). Comment on your findings.
2. Stratify the 'Trans' database into 10 strata on the Invoice Amount field beginning with a value of ($3,366.30) and using intervals of size $4,126.15. Review the strata. Add a Comment on your findings and possible explanations for the results obtained. Which invoices would you want to investigate further?
3. Extract zero or negative invoice dollar amounts in the Invoice Amount field. What do the results say about the company's controls over invoice entry? Add a Comment.
4. Open the 'AR' database, which shows the dollar amounts for each transaction that affects accounts receivable. View the statistics for the data. Add a Comment on these results.
5. The Trans Type field consists of codes describing the type of transaction. The following codes have been used: CN = Credit note (a.k.a., a credit memo), IN = Invoice, PM = Payment made by a customer, and TR = Transfer (a.k.a., a write-off). Since invoices increase the balance in accounts receivable, they should be positive. All the remaining types of transactions decrease accounts receivable and should be negative. Based on this coding scheme, extract inappropriate customer payments. (Hint: Use the ***Direct Extractions*** button (or ***Data→Extractions→Direct Extraction***)). Add a Comment on your findings.
6. Use ***Quick Summarization*** to determine the total amount owed by each customer. This command allows you to total a numeric field for each unique value of a field. In this case, you want to compute the total amount owed for each customer number (key field). Make sure that the "Create database" box is checked and that you name the output database.
   a. Summarize the 'AR' database on the Cust Number field and accumulate on the Trans Amount field. As part of the audit, auditors will reconcile these computed balances to the client's accounts receivable records before proceeding with the confirmation process.
   b. Stratify the total amount owed by each customer into 10 strata on the summed transaction amount (created in step a). Begin with a value of ($67.91) and use intervals of size $9,595.90. Make sure to total on the summed transaction amount as well to show the total (dollars) for each strata.
7. Use the ***Aging*** function to create an aging of the invoices outstanding as of 12/31/00. Assume that you want to age the invoices according to the following categories: 29, 44, 59, 89, 119, and 179. Add a Comment briefly describing your results and explain the benefit of this tool on an audit engagement.
8. Given the evidence you've collected, write a memo describing which accounts receivable you would confirm along with an explanation for your choice.

Step 3: Collect the Customers' Information

The data needed for confirmation letters resides in several different tables (or databases in IDEA terms). To access the fields from multiple databases, a relationship must be established between the databases. This relationship will allow columns to be added from one database into another. As a result, all of the fields needed for the confirmation letters will exist in a single database.

1. Review the fields found in the 'AR' and 'Customer' databases. Identify the fields from each database that are necessary to create confirmation letters. The confirmation letters require data about customers and their outstanding balances. Note: In the prior analysis, you saved several variations of the accounts receivable data. To create the confirmation letters, you must decide which one to combine with the customer data.
2. To join the databases,
   a. Identify the common key that you will use to join the databases. Verify that the common field has the same format in both databases using ***Field Manipulation*** (***Data→Field Manipulation***).
   b. Open the primary database for the join (transaction file). Click the ***Join Databases*** button. Click the ***Select*** button and choose the secondary database (master file).
   c. For both the primary and secondary databases, select only the fields you want for the confirmation letters.
   d. Name the output database 'confirmdata.' Use the ***Match*** button to identify the key(s) (i.e., the common field) you are joining the databases on using the default Ascending order. Use the "All records in both files" option. Click ***OK***.

Step 4: Working with Joined Databases

Using a database created by combining other databases is simple. Click on the tab with the newly created joined database (here, confirmdata created above). Work with the joined database as you would any IDEA database.

1. Edit your database according to the following instructions. Make sure the field format for the Trans Amount field reports negative values as (1,000.00). Format the fields, column headings, subtotals, and footer to Times New Roman, size 10 font. Makes sure the full field contents are visible.
2. Print a report of the confirmation data with a landscape orientation. The report should include a heading that describes the content of the report along with your name. The title should be formatted as Times New Roman, size 14 font, Bold Italic. Review the printout. If you note any unusual item(s), write a brief Comment describing each.
3. Export the confirmation database as a Word file to your USB drive. Only include the fields needed for the confirmation letters. Name the output as 'confirm.'

Step 5: Create the Confirmation Letters

1. Write a confirmation letter template in Word using the reference(s) provided by your instructor. You will create a mail merge between this document and the confirmation data you exported from IDEA.
2. Begin by opening the confirmation letter template. Under the ***Mailings*** tab click on the ***Start Mail Merge*** icon and choose the Step by Step Mail Merge Wizard.
   a. Select Letters as the document type and use the current document as the starting document.
   b. Identify the confirm.doc file exported from IDEA as the existing list of recipients.
   c. Edit the form letter to include the fields needed from the IDEA confirmation data. Use the address block feature (match the address fields if needed) and insert the Amount field in the appropriate places in the confirmation letter template.
   d. Complete the mail merge and select Edit Individual Letters. Save the new document that contains all the merged confirmation letters.
3. Print the <u>third</u> confirmation letter and sign it as if you are the client requesting the confirmation on behalf of the external auditors.

***Required*:** **Turn in the confirmation data report, confirmation letter, your History log, Comments, and a zip file of your work.**

IDEA PROJECT 2: EXPENDITURE CYCLE ISSUES

In this project, you will work with data in the expenditure cycle, including accounts payable transactions, the vendor master data, and the inventory master data. The goal is to work with field manipulations, criteria, reestablish relationships between flat files, and extract data from one database into another smaller one. A flat file is simply a database table without any of its relationships. After importing data into IDEA, the relationships will be restored.

Step 1: Get Started

1. Open IDEA and select ***File→Set Working Folder*** to specify the USB drive where you want the data files to be stored. Navigate to a folder that contains your name. Click ***Open***. In the Project Properties window, enter the project name as Accounts Payable – Your Name for the period Jan. 1, 2000 – Dec. 31, 2000. Click ***OK***.
2. Use the following steps to import the 'Inventory', 'Vendor', and 'AP_Trans' databases into IDEA.
   a. Use the ***Import Assistant*** button (or ***File→Import Assistant→Import to IDEA***) to begin importing a file.
   b. In the Import Assistant window, highlight Microsoft Excel. Click Browse (…) by the Filename and find the desired Excel file. Highlight the file and then click ***Open***. Then click ***Next***.
   c. In the Excel window, click the First Row is field names box. Make sure the data is formatted properly. Click ***OK*** to import the Excel worksheet.
3. The History log is a record of audit findings conducted using IDEA. The History log cannot be altered. Additional comments can be documented in a separate comments file. Therefore, when you are asked to comment on or describe your results, record your answers as a comment. Insert a comment that includes your name (Hint: In the Comments section of the Properties window, select ***Add Comment***). Place all required comments in this Comments file.

Step 2: Field Manipulations, Extractions, and Criteria

Field manipulations can be used to define new fields. To identify a subset of the records in a database that match given criteria, you can use criteria or extractions. Field manipulations, criteria, and extractions can be a combination of fields (both data and/or computed fields), functions, and operators.

Add a Computed Field to a Database

1. With the original 'Inventory' database open:
   a. Select the ***Field Manipulation*** button (or ***Data→Field Manipulation***).
   b. Add a field using a parameter that computes the unit markup percentage. The field should display two decimal places and be named Markup.
   c. Click ***OK*** to add the Markup field to your inventory database.

NOTE: ***Extraction*** and ***Criteria*** are both used to identify records that meet certain criteria. A ***Criteria***, however, does not create an output database; it displays the appropriate records in the Database window and is not recorded in the History log. On the other hand, an ***Extraction*** creates a new database and is recorded in the History log.

Working with Criteria

2. With the 'Inventory' database open, select the ***Criteria*** link (in the Properties window).
   a. Write an equation that identifies inventory items with a negative total cost (Hint: Use the ***Insert Database Field*** button in the Equation Editor.)
   b. Validate the equation and then exit.
   c. The new Criteria can be reused, changed, or deleted by selecting it from the Properties window. Add a Comment describing your result.
3. Use a ***Criteria*** to count the inventory items with a negative or zero quantity on hand. Add a Comment describing your result.
4. Use a ***Criteria*** to count the inventory items that have a sale price lower than unit cost. Add a Comment describing your result.

Working with Extractions

5. To examine the specific inventory items that have a unit cost higher than their sales price, use the ***Direct Extractions*** button (or ***Data→Extractions→Direct Extraction***). Save the extraction as LossItems with your initials. Notice that this command creates a new database and was written to the History log but the ***Criteria*** created in steps 2-4 were not.
6. Prepare a report of the database LossItems. Give the report a descriptive title that includes your name.
   a. Hide these fields from the view: Location, Product Class, and Market Value. (Hint: Use ***Column Settings***).
   b. Adjust the remaining columns so that each field's entire contents are visible.
   c. Print and save the database.

Step 3: Recreating Relationships from Flat Files

To analyze inventory transactions with vendors requires a comparison of data in the 'AP_Trans', 'Inventory', and 'Vendor' tables (or databases in IDEA terms). To access the fields from multiple databases, a relationship must be established between the databases. This relationship will allow columns to be added from one database into another. As a result, all of the fields needed for the confirmation letters will exist in a single database.

1. Review the 'AP_Trans', 'Inventory', and 'Vendor' databases. Determine how you are going to link these databases.
2. To join two databases,
   a. Identify the common field that you will use to join the databases. Verify that the common field has the same format in both databases using ***Field Manipulation*** (or ***Data→Field Manipulation***).
   b. Open the primary database for the join (transaction file). Click the ***Join Databases*** button. Click the ***Select*** button and choose the secondary database (master file).
   c. For both the primary and secondary databases, select only the fields you want to include in the joined database. NOTE: You can limit the fields added to the primary database using the ***Fields*** button. You must include the field you are joining the databases on-otherwise, the join does not work. Alternatively, you can change the view of the data as described below in Step 4.
   d. Name the output database. Use the ***Match*** button to identify the key(s) (i.e., the common field) you are joining the databases on using the default Ascending order. Use the "All records in both files" option. Click ***OK***.
3. Now, using this joined database as the new primary database, add the information from the remaining database with another join.

Step 4: Working with Related Databases

Using a database created by combining other databases is simple. Click on the tab with the newly created joined database (from Step 3). Work with the joined database as you would any IDEA database. Only virtual or user created fields can be deleted from the database. Therefore, users cannot delete columns for a report. However, you can eliminate columns from a screen or report using the ***View*** menu. With the newly joined database you created in Step 3 open, select the following fields (in order) to include in your view: Vendor Number, Vendor Name, Product Description, Invoice Date, and Invoice Amount.

1. To change the order of the columns, click on the field name to highlight the entire column, hold down the mouse until the drag icon appears, and then drag the column to the desired position.
2. Remove the fields that you do not want by clicking on the field name to highlight the column. Right-mouse click and select Hide Fields.
3. Prepare a report of this new view. Give the report a descriptive title that includes your name.
   a. Limit the report's contents to vendors with invoices in excess of $5,000.
   b. Format the entire database using Tahoma with at least a 10 font.
   c. Format the Invoice Amount field as (1,000.00) and include a subtotal for this field.
   d. Print and save the view. Add a Comment describing the purpose of this report.

Step 5: Extracting Selected Data

IDEA's ***Direct Extraction*** command allows you to save specific records or fields from a large database into a smaller one. A ***Direct Extraction*** copies a subset of the records or fields from a database and creates a new database. Records in the original database remain unchanged. The basic steps for extracting data are as follows: (1) identify the subset of data that you want to extract, (2) compute control totals, (3) extract the records to a new database, and (4) compute new control totals and compare them to the original ones.

1. With the 'Inventory' database open, select ***Analysis→Summarization→Quick***.
   a. Enter the summarization field as Location.
   b. Compute control totals for all numeric fields.
   c. Limit the results to only those items stored at Location #4.
   d. View the control totals and record count by using the Field Statistics link in the Properties window.
2. With the 'Inventory' database open, select the ***Direct Extractions*** button and enter a file name for the output database.
   a. Select the "Insert database fields" button and limit the database to only those items stored at Location #4.
   b. Validate the equation and exit.
   c. Using the ***Fields*** button, limit the fields extracted to the Product Number, Product Description, Quantity on Hand, and the Inventory Value at Cost.
   d. With the new database open, view the new control totals by clicking on the Field Statistics link in the Properties window.
3. Verify the ***Direct Extractions*** results by comparing them with the control totals calculated with the ***Quick Summarization***. Add a Comment to document the results of your comparison. Briefly explain the purpose of this comparison.
4. In the Comments, briefly describe one situation where extracting data could be useful in an audit.
5. Prepare a report of the extracted database. Give the report a descriptive title that includes your name.
   a. Create a control total for Quantity on Hand.
   b. Format the entire database using Tahoma with at least a 10 font.
   c. Format Quantity on Hand and Inventory Value at Cost so that negative amounts appear as (1,000.00). Add a subtotal for Inventory Value at Cost. Make sure each field's entire contents are visible.
   d. Print and save the report.

***Required*:** **Turn in any printouts, your History log, Comments and a zip file of your work.**

IDEA PROJECT 3: PAYROLL CYCLE ISSUES

This project focuses on auditing payroll. In particular, you will be looking at payroll records to identify situations or transactions that may require further investigation.

Step 1: Get Started

1. Open IDEA and select ***File→Set Working Folder*** to specify the USB drive where you want the data files to be stored. Navigate to a folder that contains your name. Click ***Open***. In the Project Properties window, enter the project name as Payroll – Your Name for the period Jan. 1, 2000 – Dec. 31, 2000. Click ***OK***.
2. Use the following steps to import the 'Empmast and 'Payroll' databases into IDEA.
   a. Use the ***Import Assistant*** button (or ***File→Import Assistant→Import to IDEA***) to begin importing a file.
   b. In the Import Assistant window, highlight Microsoft Excel. Click Browse (…) by the Filename and find the desired Excel file. Highlight the file and then click ***Open***. Then click ***Next***.
   c. In the Excel window, click the First Row is field names box. Make sure the data is formatted properly. Click ***OK*** to import the Excel worksheet.
3. The History log is a record of audit findings conducted using IDEA. The History log cannot be altered. Additional comments can be documented in a separate comments file. Therefore, when you are asked to comment on or describe your results, record your answers as a comment. (Hint: In the Comments section of the Properties window, select ***Add Comment***). Place all required comments in this Comments file.

Step 2: Substantive Test of Accuracy

As auditors, we want to ensure the accuracy of the payroll calculations. As a substantive test of accuracy, you will recompute net pay.

1. Open the 'Payroll' database and insert a Comment that contains your name.
2. Add a new field that recomputes net pay.
3. Create an extraction to test whether the original net pay field equals your recalculated amount. (Hint: Use the ***Direct Extractions*** button (or ***Data→Extractions→Direct Extraction***)).
4. Add a Comment describing the results of this test.

Step 3: Related Employees

As auditors, we are concerned with segregation of duties. To test for segregation of duties, we need to review the organizational chart and job descriptions for conflicting duties. We also should observe employees to determine whether the segregation policy is followed in practice. As an additional test, we may search the employee records for evidence of related employees. To test for related employees, you will isolate employees with the same last name or same address.

1. Open the 'Empmast' database.
2. Click on the ***Duplicate Key Detection*** button (or ***Analysis→Duplicate Key→Detection***). Output duplicate records and name the output database. For the Key field select the employee's last name. Select the following fields to list: Last Name, First Name, Employee Number, Street Address, Job Description, and Gender.
3. Now check for duplicate addresses. Select the following fields to list: Street Address, Employee Number, First Name, Last Name, Job Description, and Gender.
4. Briefly describe in the Comments what your results suggest and the steps you would take to follow-up on them.

Step 4: Fraud Detection

As auditors, we are concerned about the possibility of fraud. Overpayment of employees and payment to nonexistent employees are two common types of payroll fraud. To test for payroll fraud, you will look for multiple paychecks to the same employee and verify that all checks were paid to valid employees.

Test for Multiple Paychecks

1. Open the 'Payroll' database.
2. Check for duplicate employee numbers. Select the following fields to include: Employee Number, Check Number, Gross Pay, and Pay Date. Name the database.

Test for Paychecks to Only Valid Employees

Evaluating whether all the payroll checks were paid to valid employees requires a comparison of data in the 'Empmast' table and 'Payroll' table (or databases in IDEA terms). To access the fields from multiple databases, a relationship must be established between the databases. This relationship will allow columns to be added from one database into another. As a result, all of the fields needed for this test will exist in a single database.

3. Review the 'Empmast' and 'Payroll' databases. Determine how you are going to link these databases.
4. To join the two databases,
   a. Identify the common field that you will use to join the databases. Verify that the common field has the same format in both databases using ***Field Manipulation*** (or ***Data→Field Manipulation***).

b. Open the primary database for the join (transaction file). Click the ***Join Databases*** button. Click the ***Select*** button and choose the secondary database (master file).
c. Include only the fields in the primary database plus the Employee Number field in the secondary database.
d. Name the output database. Use the ***Match*** button to identify the key(s) (i.e., the common field) you are joining the databases on using the default Ascending order. Use the "All records in both files" option. Click ***OK***.

5. Create an extraction that limits the database to records where the employee numbers do not agree.
   a. Rearrange the fields so that the two Employee Number fields are adjacent.
   b. Print the extraction as a report. The report should include a heading that describes the report's contents and your name. Make sure all field contents are visible. Save the report.
6. Add a Comment describing your findings and possible causes for them.

Required: **Turn in the Fraud report, your History log, Comments and a zip file of your work.**

IDEA PROJECT 4: IMPORTING DATA

The goal of this project is to import a client's database into IDEA. The client has supplied you with a copy of their Microsoft Access database, along with an entity-relationship diagram. To import their database into IDEA, each table will be converted into a flat file. A flat file is simply a database table without any of its relationships. The relationships will be restored once the files are accessible in IDEA.

Step 1: Import the Database into IDEA as Flat Files

1. Download the Access database from the class website to your desktop.
2. Open IDEA and create a new Working Folder using your name and save it to your USB drive.
3. Select the ***Import Assistant***. Click on Microsoft Access as a pre-defined Format and follow the Import Assistant to import all tables from the database into IDEA.

Step 2: Modify the Databases in IDEA

1. Double click on any record to view the ***Field Manipulation*** window (or click on the ***Field Manipulation*** button). Ensure that each database has field names consistent with the original database. Use the ***Field Manipulation*** window to make any necessary changes.
2. Right click on any column to view the ***Column Settings*** (or click on the ***Column Settings*** button). Filter on numeric fields. Highlight each currency field and format it so that negative amounts appear as (1,000.00).
3. Make sure that primary key fields are formatted as ASCII not Numeric.
4. Adjust the column widths to the size of the data. (Hint: Place cursor at the top of the Data by the field names until double arrow appears. Adjust the column width using the double arrow.)

Step 3: Print the Databases

1. Open each database, review your work and make any modifications needed.
2. Create a report for each database. Add a heading to the report that describes the database and includes your name.
3. Format each entire report as Times New Roman, 10 font. (Hint: Use the ***Column Settings*** window.) Change the font for the heading to Times New Roman, 14 Bold Italic font.
4. Print the first page of each database. If possible, force the database to print on a single page.

Step 4: Create Additional Databases/Views (to be defined by the instructor as indicated by XXX)

1. Open the database for the XXX data.
2. Create the following databases/views: XXX, XXX.
3. Print the additional databases/views. Make sure you follow the printing instructions in Step 3 above.

Step 5: Wrap Up

1. Add a Comment that includes your name. (Hint: Click ***Add Comments***.)
2. Print the History log for each database (be sure to expand to show all before printing).

***Required*:** **Turn in all of the database/view printouts, your History log, Comments, and a zip file containing your completed project.**