Computerized JAVA Test For Object - Oriented Programming (OOP) Course

by

Elina Shafinaz Binti Gulam Mohd

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Systems)

DECEMBER 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

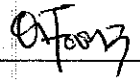
Computerized JAVA Test For Object – Oriented Programming (OOP) Course

By

Elina Shafinaz Binti Gulam Mohd

A project dissertation submitted to the
Information Technology / Information Systems Programme
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION SYSTEMS)

Approved by,

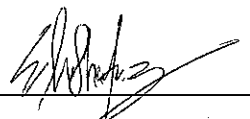


(Mdm. Amy Foong Oi Mean)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
July 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



(ELINA SHAFINAZ BINTI GULAM MOHD)

ABSTRACT

This report covers the scope of studies, the project background and problem definition as well as to define the project objectives. Furthermore, this report also provides literature review and theory on the chosen project, provides result and findings, and explains the methodology and projects works towards its accomplishment.

The objectives of this project, Computerized JAVA Test for Object - Oriented Programming (OOP) Course is to enable the lecturers to save time as the test can be graded online automatically. Besides that, this system too ensures that the lecturers have more time to teach and interact with the students. This indirectly would help to enrich the teaching process.

Conventionally, almost all the testes conducted in the Universiti Teknologi PETRONAS (UTP) are done manually, where it is conducted using a paper-pen format. This creates hassle to the users and irritates them especially to both the UTP students and lecturers. So, a computerized test system is vital, as it enable the students to get their test graded online.

The methodology used for designing and developing this project is the "Waterfall Model." where it begins at the system level and progresses through analysis, design, coding, testing, and maintenance. Based on the system requirements, the author develops main interfaces, forms and views required for the Computerized JAVA Test for OOP Course by using the Microsoft Visual Basic (VB) Version 6.0.

Through the development of this computerized test system, the author hopes that it can help to eliminate the problems that arise in the old manual procedure, so that the department can improve its performance in the future.

ACKNOWLEDGEMENT

The Computerized JAVA Test for Object – Oriented Programming (OOP) Course is the result of the sustained and dedicated efforts, whose unstinted cooperation and abiding interest made the realization of this dream possible.

The author's first thanks go to the Final Year Project (FYP) Committees and Coordinators for giving her a chance to design and develop this Computerized JAVA Test for the OOP Course, which meets her interest and capabilities of completing it.

The author would also like to take this opportunity to express her deepest gratitude to her enthusiastic lecturer, Mdm. Amy Foong Oi Mean, for her guidance and assistance in giving the author a big hand upon completing her final year project successfully. Besides that, the author would also want to extend her special thanks to her parents and her family members for their full support and commitments in making this project a reality.

Last but not least, the author also would like to thank her fellow friends and every person that might not directly support the author towards the completion of the project. All your help, support, and guidance are highly appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS AND NOMENCLATURES

DFD – Data Flow Diagram

UTP – Universiti Teknologi PETRONAS

FYP – Final Year Project

Mdm. – Madam

OOP – Object-Oriented Programming

IT/IS – Information Technology / Information System

VB – Visual Basic

UML – Unified Markup Language

CHAPTER 1
INTRODUCTION

1 INTRODUCTION

As technology evolves, the demand for an effective computerized system also increases. More and more end users are using computer based systems as their primary source of information, and have reassign paper documents as a secondary source. This computerized system can provide quick access to the relevant information anytime and anywhere. In other words, when there is a computer, there is an access to the information.

1.1 Background of Dissertation

The Information Technology / Information System Department of Universiti Teknologi PETRONAS (UTP) is responsible in offering a course known as "Object-Oriented Programming (OOP)" for both Information Technology / Information System (IT/IS) students in order to produce well-rounded students. This course focuses on one programming language called the JAVA Programming language. All the third year students are required to take this course as part of their academic requirements.

Currently, the JAVA test is conducted manually using a paper-pen format. As the number of students has been increasing yearly, it is time consuming for the lecturers to mark the test papers manually. Besides that, the students too who are eager to know their test results couldn't get their results on time. Due to these constraints, a Computerized JAVA Test for OOP Course will be the best solution. This test will enable the students to do their test online in the programming labs and get their testes graded immediately. In addition, the integrated question banks will enable the lecturers to easily add, delete, and edit the test questions. Besides that, this system too provides the privilege for the lecturers to save the test questions according to the chapters as stated in the OOP course syllabus.

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

Conventionally, almost all the testes conducted in UTP are done manually, where it is conducted in a paper-pen format. This creates hassle to the users and irritates them especially to both the students and lecturers. So, a computerized test system is vital, as it enable the students to get their test graded online. After conducting an interview sessions with the targeted users, several problems have been highlighted:

a. *Time consuming*

It consumes time for the lecturers to mark the students' test papers manually as the lecturers have a lot of things to do.

b. *Risk of loss of documents*

The storage of the test grades is done manually. The possibilities of losing the documents are high due to unorganized way of handling the documents.

c. *Cheating and copying during test*

Testes that are conducted manually in a paper-pen format often allow the students to copy each others answer as the questions are arranged accordingly in the question booklet.

d. *Incur cost*

Incur cost in printing the test questions on paper. If there is any mistake in the printed test questions, the lecturers need to re-edit the test question and print them again.

e. *Take time to inform the students on their test results*

As for the manual JAVA test, the lecturers need time to mark the papers manually and could not inform the test results to the students as soon as possible. Students feel very eager and can't wait to see their test result.

1.2.2 Significance of the Project

The project helps the Information Technology / Information System Department to solve the problems that they are facing currently and also helps to easily grade the students once they have completed their testes. By providing this Computerized JAVA Test, it directly saves the lecturers' time and also eliminates the hassle created by marking the test papers manually. This system will enable the lecturers to store, organize, manage and control the test questions more efficiently and effectively. Besides that, the integrated question banks will allow the lecturers to add, delete, edit and save the test questions easily. With the available of technology in the market, it is possible to complete this project within the time frame given. There are several techniques available to quickly identify the possible solutions for the problems discussed above such as fact-finding and researching technique. This can be done by distributing questionnaires, conducting interviews, or by having group discussions.

1.3 OBJECTIVES

The main aim of this project is to overcome the problems encountered by both the lecturers and students of the IT/IS Department, Universiti Teknologi PETRONAS. This project will be conducted by following closely on the Waterfall Model Methodology as described in Chapter 3 of this report. The objectives of this project are listed below :-

a. To enable the lecturers to save time as the test can be graded automatically once the students had completed the test.

b. To ensure that the lecturers have more time to teach and interact with the students.

c. To help deter cheating as there is no advantage of seeing another student's test paper as the questions will be randomized with the integrated question banks.

1.4 SCOPE OF THE PROJECT

1.4.1 The Relevancy of the Project – Scope of Work

This dissertation was conducted by analyzing the current manual procedure of the JAVA test. The fact-finding techniques such as distributing questionnaires and interviewing the targeted users were conducted. A thorough research was carried out on the languages to be used in designing and developing this system. The dissertation does not focus entirely on the research area but also focuses on designing and developing a Computerized JAVA Test for OOP Course which is equipped with an Integrated Question Bank. This dissertation too focuses on the design and implementation of the prototype version of this system. However, testing will be done to gain the feedback from the end users. The feedback will then be used to improve the system in future.

1.4.2 The Feasibility of the Project

1.4.2.1 Technical Feasibility

The decision to develop this Computerized JAVA Test is practical since the author is proficient in designing and developing systems. Universiti Teknologi PETRONAS had exposed their students with many forms of online and computerized systems such as the Online Course Registration System and Online Course/Examination Timetable. The students will be

familiar with this system as it uses a user-friendly software, which is the Microsoft Visual Basic (VB) Version 6.0 software.

1.4.2.2 Operational Feasibility

The proposed system is believed to eliminate the problems faced by the current manual procedure. Thus, the proposed system is assumed that it will fulfill the user's requirement. Besides that, this project too will operate under the project timeline and schedule given and is guaranteed to be fully functioning at the end of this semester.

1.4.2.3 Economic Feasibility

The solution is cost effective because all the processes are computerized. Paper-pen format testes will be reduced and the computerized test system can be easily handled.

1.4.2.4 Schedule Feasibility

The system needs to be developed within the time constraint of 14 weeks including the question banks. Due to this time limit, the author need to narrow down the system scope and produce a simple but applicable Computerized JAVA Test for both the Information Technology / Information System Department.

CHAPTER 2
LITERATURE REVIEW

2.1 COMPUTERIZED TEST

According to Dr. James B. Olsen, Chief Scientist at Alpine Media Corporation in Orem, Utah [1] defined computer-based tests as *"tests or assessments that are administered by computer in either stand-alone or networked configuration or by other technology devices linked to the Internet or the World Wide Web. In the face of the rapid growth of computer-based testing, the ATP sponsored the development of formal, written guidelines to help ensure high measurement quality of computer- and Internet-based tests and to provide direction for the principles and procedures used for developing and administering those tests."* As for this project, Computerized JAVA Test for Object – Oriented Programming (OOP) Course focuses more on a standalone system rather than online test where, for time being, it edge on only one subject matter which is the OOP course. This test will be conducted in the UTP labs that is equip with Microsoft Visual Basic (VB) Version 6.0.

McGraw Hill [2] pointed out that *" easy access to online tests, as well as instant scoring, ensure that lectures have more time to teach and more time to interact with students."* This Computerized JAVA Test is capable of marking and grading the test papers automatically once the students had completed the test. This makes the lecturers' job a bit easier as the lecturers don't have to mark the test papers manually. Thus, the lecturers will have more time to tech and interact with the students and this would help to enrich the teaching process.

2.2 QUESTION BANK

Early work by Ratna, A. A. P. and P. Raymonth, et al. [3] stated that having randomized questions with the same level and structure but with different content help to deter the problem of copying and cheating as there is no advantage of seeing another student's exam paper. The problem of copying and cheating had become the major drawback with the manual pen-and-paper format as the test questions were not arranged in a randomized order.

This Computerized JAVA Test is furnished with question bank. These question banks stores large number of questions in one subject area. Any particular test questions can be selected from the banks at random as reported by Thelwall [4]. This could help the lecturers to organize the test questions easily. Besides that, the question banks also allow the lecturers to add, delete, or edit the test questions easily.

R. Neill Johnson, Diane M. Enerson, and Kathryn M. Plank, from The Pennsylvania State University, [5] stated that *"each student is presented the same questions but in random order. Students are not allowed to bring paper into the facility or take paper out, but they are given scratch paper to write on during the exam. Participants asked what would prevent students from smuggling out questions and answers. Presenters responded that while students may manage to copy or memorize a few questions and the answer choices, most find this difficult and not worth the effort. Because the questions are presented in random order, some students are initially convinced that their peers took a different exam. Test scores have been highly consistent over the years, suggesting that leakage does not occur. Because no paper copies of the exam exist, computerized testing practically eliminates fraternity test files."* For this Computerized JAVA Test, each student is presented with the same questions but in random orders. This is done to prevent the students from peeping other students answer and to deter copying and cheating problems during the test operation.

2.3 TEST SCORES

Arnow, D. and O. Barshay [6] distinguished that *"computerized test system successfully addresses many of the issues raised by recent work in on-line exams by providing reliable automatic checking of exam questions and a secure, restricted programming environment useable by the students during the exam."* This Computerized JAVA Test for OOP Course is also capable of grading the test papers automatically once the students had completed the test. This provides benefits to the students especially to those who feel eager and can't wait to know their test results.

The manual pen-and-paper format test was both time-consuming and laborious, as it takes several weeks or even months to mark the test papers manually and grade them. This could tie up the lecturers' time with marking rather than providing supports to the students as reported by Catherine Dhanjal [7]. So, this Computerized JAVA Test is definitely the most appropriate path to take. By having the automatic assessment, it could help to free up the lecturers' time and the lecturers can spend more time with the students.

The Computerized JAVA Test for Object – Oriented Programming (OOP) Course gives the students an option of printing their test results the moment they have completed the test. This, results in a comprehensive assessment of student achievement levels, as it enable the lecturers to easily respond to individual needs more quickly than before as noted by Hutchin [8]. Rigby and Blaine [8] noted that the timelines of the test data is also equally important as it gives the lectures the ability to monitor the students' progress at the beginning and near the end of the semester. The ability to get information back immediately once the student had finished the test enable the lecturers to quickly look at the scores, see whether the students are on track in meeting their expectations or their grade and then re-access the teaching process to address those results.

CHAPTER 3
METHODOLOGY/ PROJECT WORK

3.1 PROCEDURE IDENTIFICATION

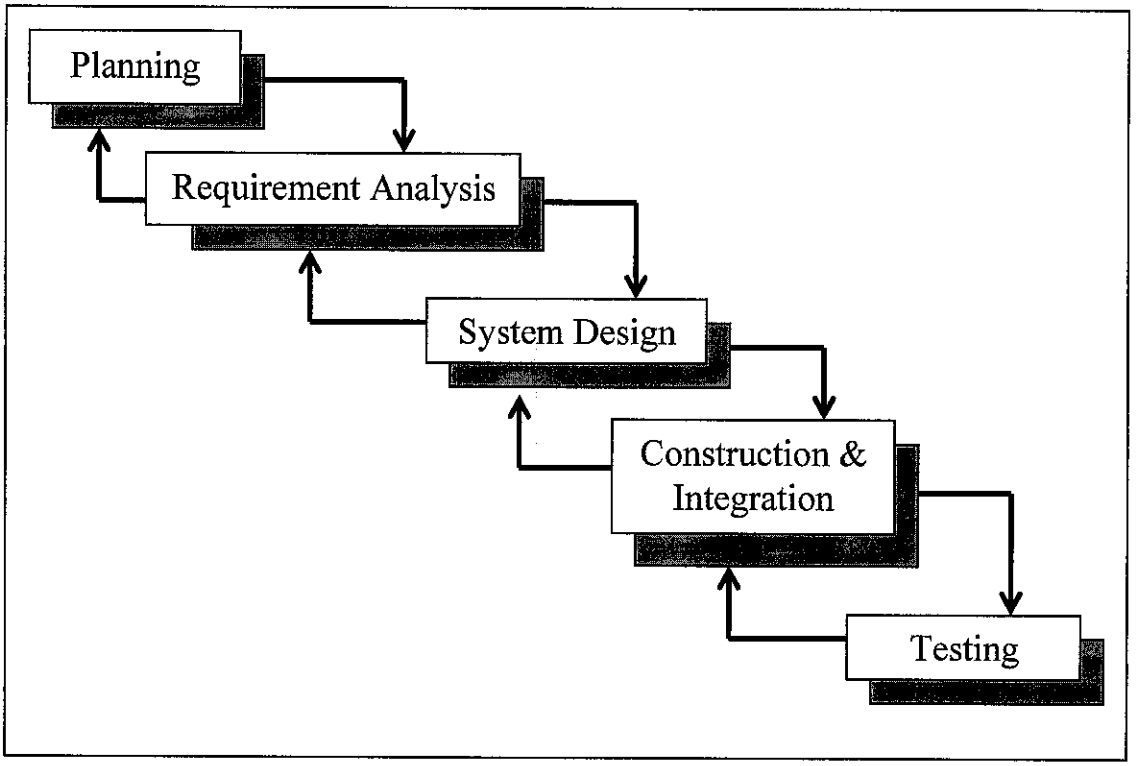


Figure 3.1 : Waterfall Model

Figure 3.1 illustrates the life-cycle paradigm for the software engineering. Sometimes called the "Waterfall Model", the life-cycle paradigm demands a systematic, sequential approach to software development that begins at the planning stage and progresses through requirement analysis, system design, construction & integration and finally, testing. This model is known as the waterfall model because each of every phase is cascading from one another. The core main phases of Waterfall Model is explained in detail below :-

1. *Planning*

In the planning phase, the system concept was developed to describe how the system will operate once it is implemented. Furthermore, this phase is also used to assess how the system will give an impact to end users while performing their daily activities. During this phase, the author had clearly defined the problems, opportunities, and directives towards the development of this Computerized System. The author too had developed a Gantt Chart *(refer Appendix 5.0)* which provides a standard format for displaying project schedule information by listing project activities and their corresponding start and finish dates in a calendar format. The tasks, duration estimates, and dependencies were also identified. The outcome of this task are problem statements, objectives as well as significant of the system being developed.

2. *Requirement Analysis*

Requirement analysis is the most important phase of the methodology. The system's goals, constraints, and services are established by consultation with the system user. Essentially, the purpose of requirements analysis is to identify data, process, and interface requirements for the users of the proposed system. The groundwork for this task was established in the problem analysis phase during identification of system objectives. The system was defined in more detail with regard to system inputs, processes, outputs, and interfaces. This definition process occurs at the functional level. The system shall be described in terms of the functions that need to be performed, not in terms of computer programs, files, and data streams. The emphasis in this phase is on determining what functions must be performed rather than how to perform those functions.

Fact finding is the formal process of using research, interviews, questionnaires, sampling, and other techniques to collect information about problems, requirements, and preferences. Tools, such as data, process, and object models were used to documents facts and conclusions from documented facts. There are few fact

finding methods are available for data gathering processes. They include existing documentations, forms, and databases, research and site visits, questionnaires, prototyping, Joint Requirements Planning (JRP), and many more. The author had used two types of data gathering procedures upon developing this Computerized Test for OOP Course which were the interview session *(refer Appendix 1.0)* and the questionnaires *(refer Appendix 3.0)*. This data gathering technique was conducted between two types of targeted users which are the UTP lecturers and the UTP students. The main reason of conducting this data gathering technique is to seek opinion and feelings from the users in designing anf enhancing this system in future.

3. System Design

The purpose of system design phase is to transform the requirement statements from the requirement analysis and definition phase into design specifications for construction. It involves identifying and describing the basic software system abstractions to ensure completeness, usability, reliability, performance, and quality. Hardware or software requirements will be determined in this phase. The system architecture will be produced on the whole.

In this phase, the data models and process models that were initially created during requirements and definition phase were analyze in detail. System users were involved in this activity to help address data and process issues. The key inputs to this task were the facts, recommendations, and opinions that were solicited form various sources. The author is working closely with the system users to develop input, output, and dialogue specifications. For designing the user interface, the author has considered such factors as terminal familiarity, possible errors, and misunderstandings that the end users may encounter. Furthermore, the author was trying to make it easy for the end users to understand what the screen is displaying at any given time.

The author had decided to use Microsoft Visual Basic (VB) Version 6.0 in designing this whole computerized system as it is an object-oriented and event-

driven programming language. VB emphasizes on a program that includes objects in the interface as well as events that occur on those objects. Event-driven language needs full control from the user. This means, they will wait for user to take action before they execute. (*Refer Appendix 2.0 for the User Interface Design of this Computerized JAVA Test for OOP Course)*

4. *Construction & Integration*

The objective of the construction phase is to convert the deliverables of the system design phase into a complete system. The construction phase contains activities for building the system, testing the system, to ensure the system functional processes satisfy the user requirements. At the end of this phase, the system will be ready for the activities of the integration and testing.

Programming is generally recognized as a major aspect of the construction phase. The primary inputs for this activity are the technical design statement, plan for programming, and test data develop during system design. In integration phase, individual program units or programs are integrated as a complete system to ensure that the software requirements have been met.

5. *Testing*

In system testing, individual program units or programs that have been integrated into a complete system will be tested to avoid possibilities of system failure. Testing will be conducted during this phase. Testing must not be deferred until after the entire program has been written. There are three levels of testing to be performed namely stub testing, unit testing, and system testing. Stub testing is the test performed on individual events or modules of a program. While unit testing is a test whereby all events and modules that have been coded and stub tested for a program are tested as an integrated unit. Whereas system testing ensures that application

programs written and tested in isolation will work properly when they are integrated into the total system.

The author also tested this system in various Windows platforms such as Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000 and Windows XP to ensure the compatibility with these platforms. Once the system test is complete and determined to be successful, the system will be put into use.

3.2 DATA MODELING DIAGRAM

3.2.1 Workflow Diagram

Based on the meeting conducted with the author's supervisor and the interview session with the targeted users, the author had developed a simple workflow for this Computerized JAVA Test for OOP Course to picture what the author understands on how the system should work. This workflow had been divided into two; one for the students and one for the lecturers. *Figure 3.2* and *figure 3.3* below shows the workflow diagram for both the students and the lecturers.

FIGURE 3.2 : WORKFLOW DIAGRAM - STUDENTS

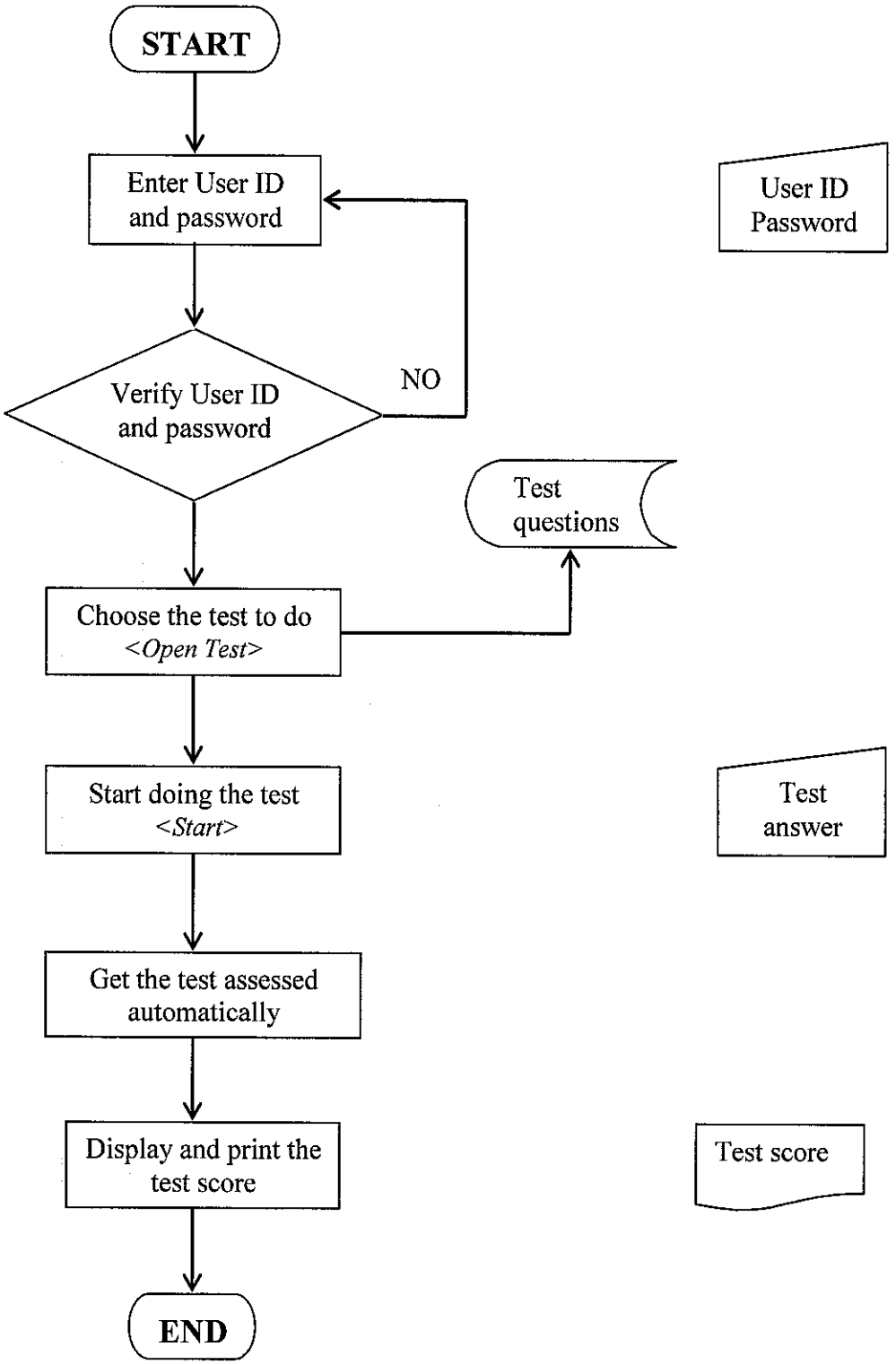
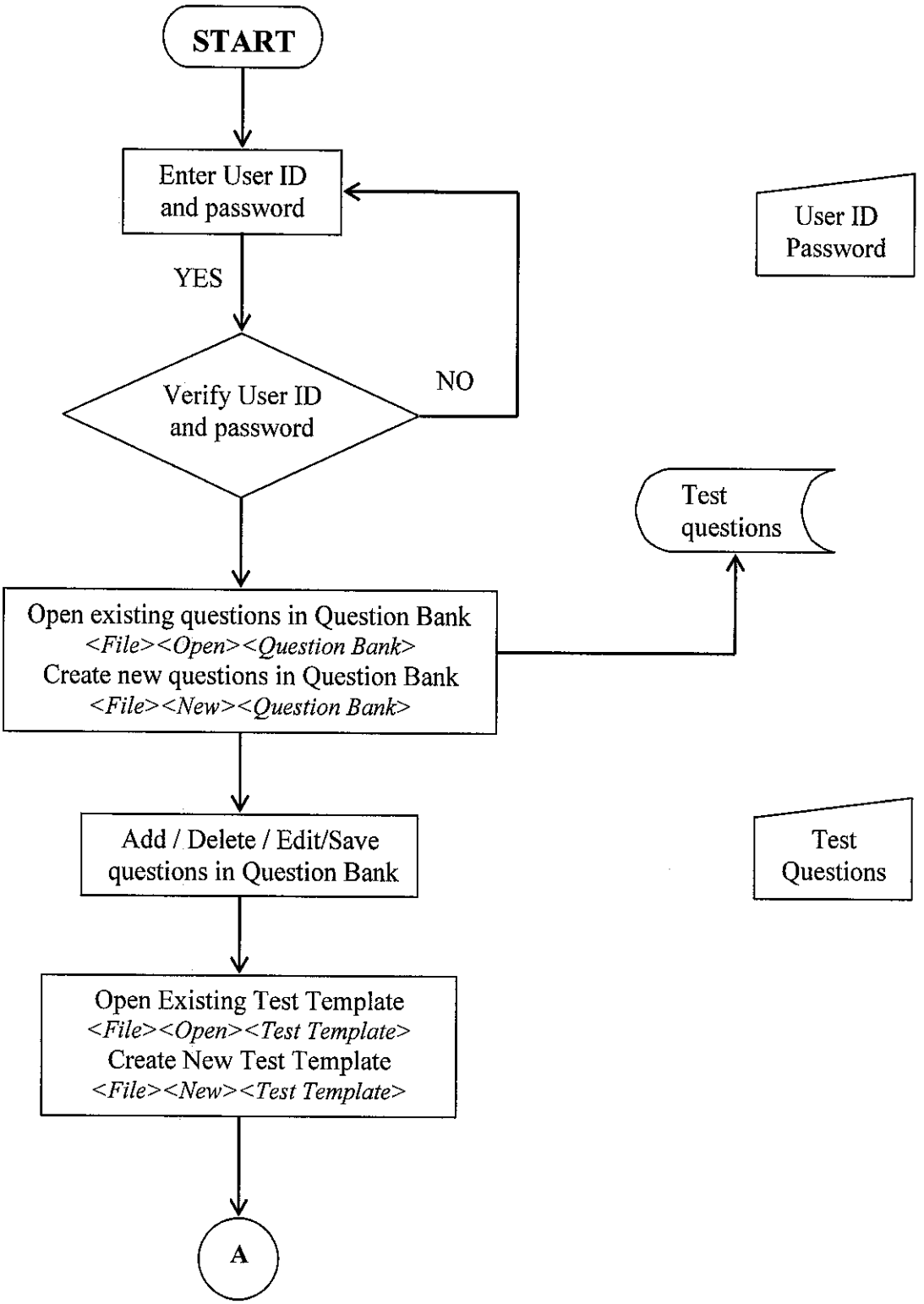
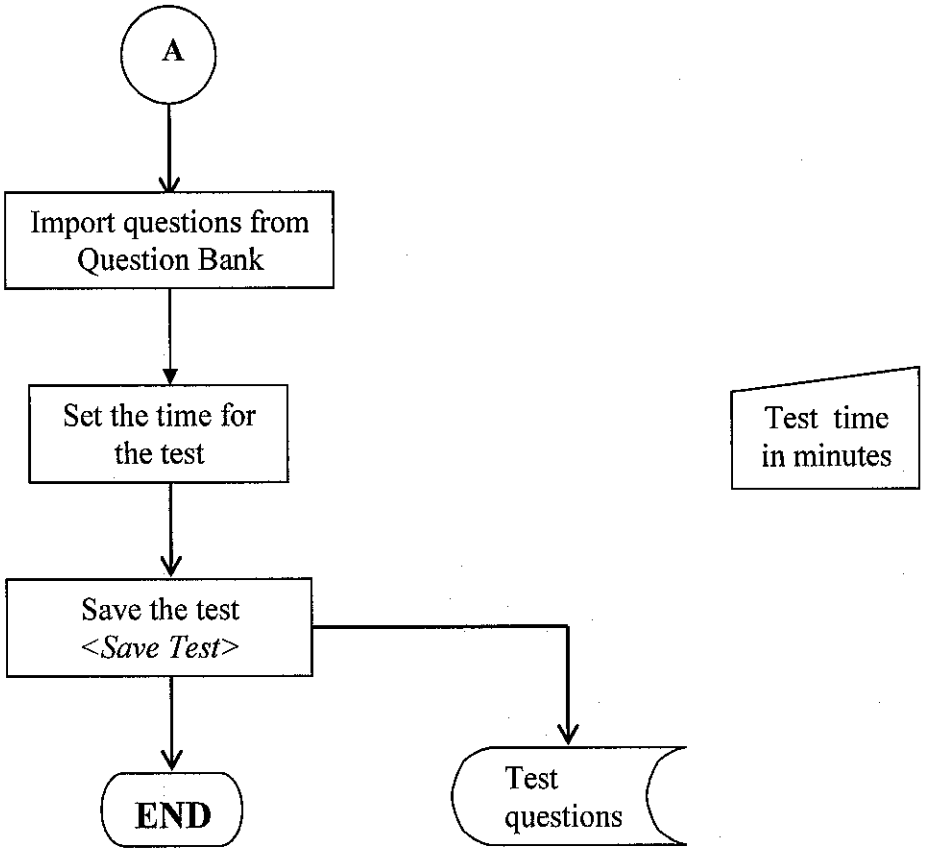


FIGURE 3.3 : WORKFLOW DIAGRAM - LECTURERS

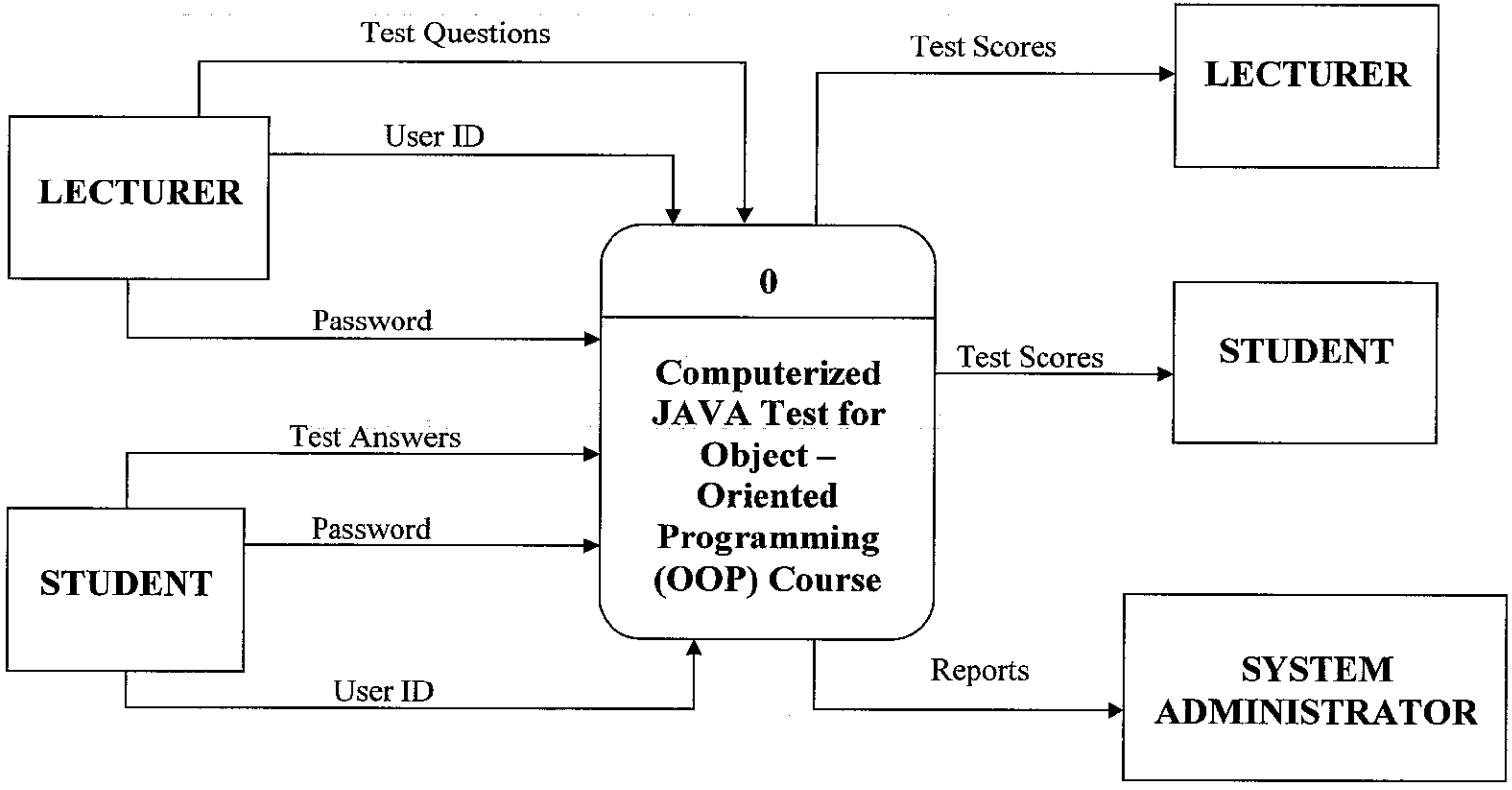




3.2.2 Data Flow Diagram (DFD)

In addition, the author too had developed Data Flow Diagram (DFD) both Level 0 and Level 1 *(refer to Figure 3.4, 3.5 & 3.6 below)* which could help the users to get a closer picture of the processes involve in this computerized test . But, this workflow is not a final workflow to be followed. There may be some changes needed based on the comments made by the users.

FIGURE 3.4 : CONTEXT LEVEL DIAGRAM (LEVEL 0)



Description:

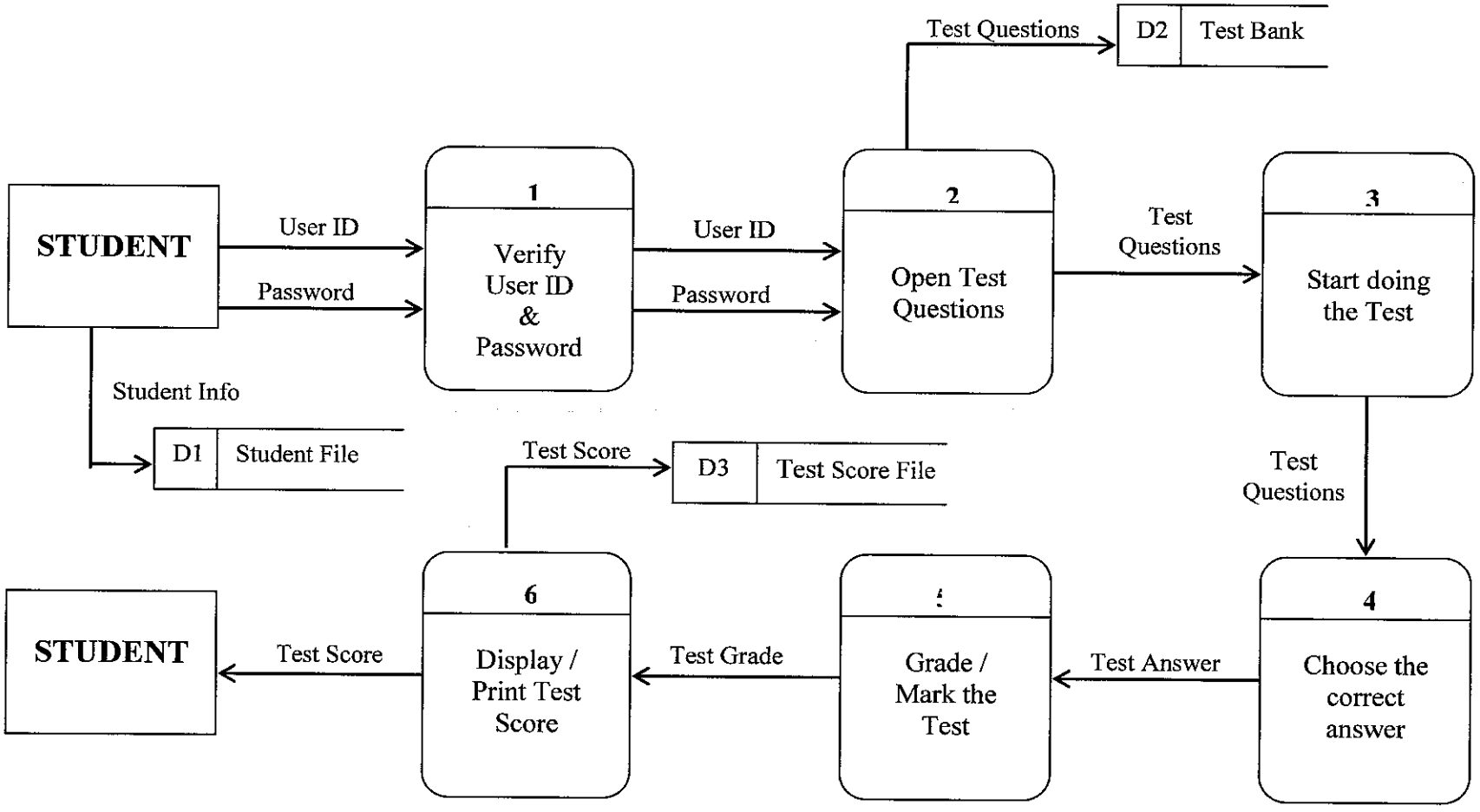
Figure 3.4 above shows the Context Level Diagram (Level 0) of the Computerized JAVA Test for Object – Oriented Programming (OOP) Course. There are three entities involve which are the user, the lecturers and the system administrators.

The students have to enter their user ID and password in order to log-in and use the system. During the test session, they have to click on the correct answers based on the questions given. The system will then responds to their answers and grade them once they had completed the test. The test scores will then be displayed to the entire student.

The lecturers too have to enter their own user ID and password in order to log-in and use this computerized system. Once log-in, the lecturers can create a few sets of test question, edit, delete or save them in the question banks according to the course chapters. In addition, the lecturers are also allowed to view or print the students' test scores for further references.

On the other hand, this Computerized JAVA Test will also be capable of sending reports to the system administrator who is responsible for maintaining and updating the system.

FIGURE 3.5 : DATA FLOW DIAGRAM (DFD) LEVEL 1 - STUDENTS



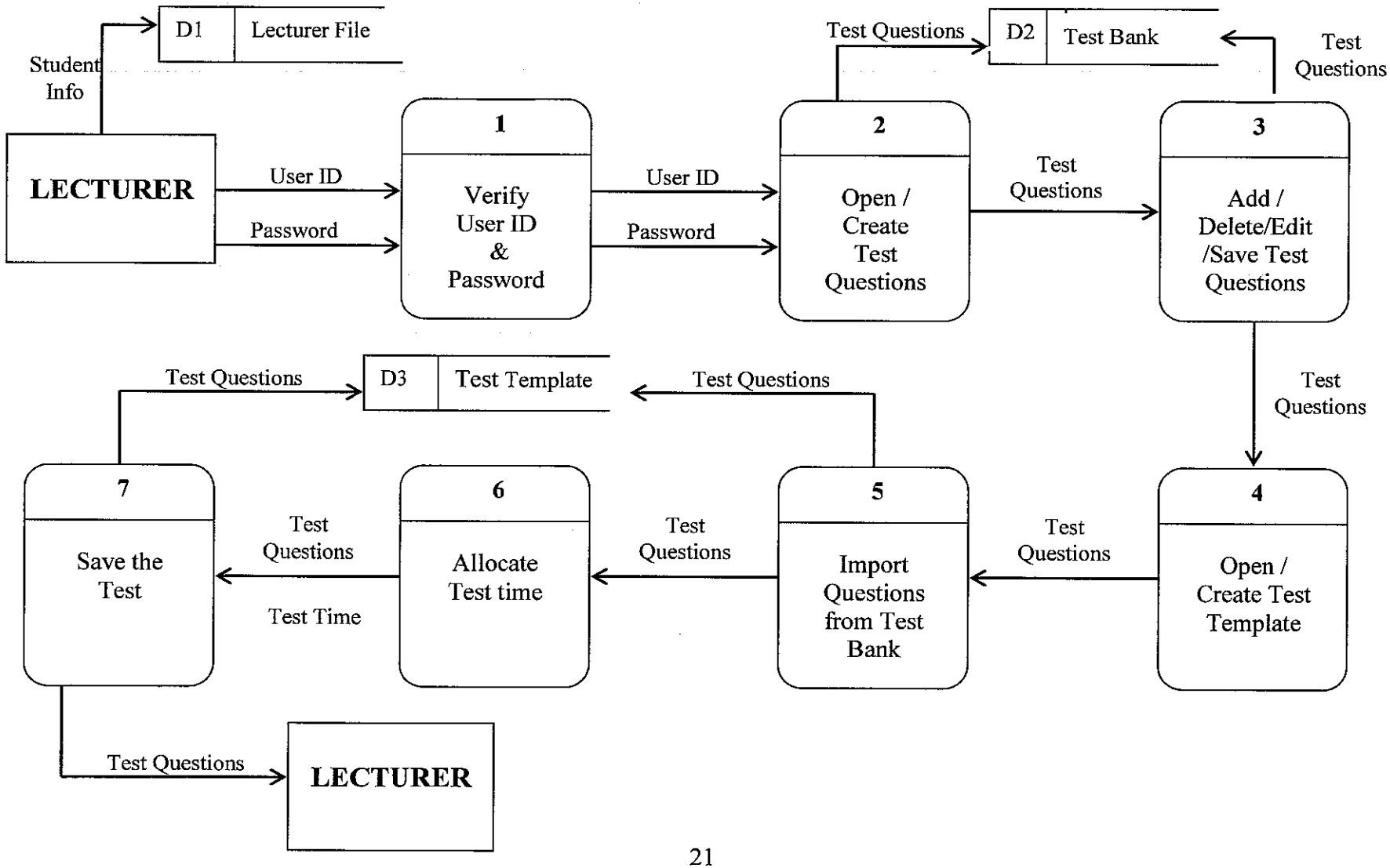
Description :-

Figure 3.5 above shows the Data Flow Diagram (DFD) Level 1 of the Computerized JAVA Test for Object – Oriented Programming (OOP) Course. There are two entities, three databases and six processes involve in this DFD Diagram.

Firstly, the students will enter their user ID and password in order to log-in to the system. The user ID and password will be kept in the student file database. The system will then verify the user ID and password entered. If both the user ID and password does not match the data in the student file, an error message will be displayed. Once log-in, the students will open the suitable test questions which is saved in the test file.

Then, the students will be allowed to start the test. They will have to read the test questions vigilantly and click on the best answer choice provided. Once they have completed the whole test questions, they can have their test graded automatically. The system will then display the test scores to the students. The students will be allowed to print a copy of the test result for their references. A copy of the test scores will be kept in the test score file for future references.

FIGURE 3.6 : DATA FLOW DIAGRAM (DFD) LEVEL 1 -LECTURERS



Description :-

Figure 3.6 above shows the Data Flow Diagram (DFD) Level 1 of the Computerized JAVA Test for Object – Oriented Programming (OOP) Course. There are two entities, three databases and seven processes involve in this DFD Diagram.

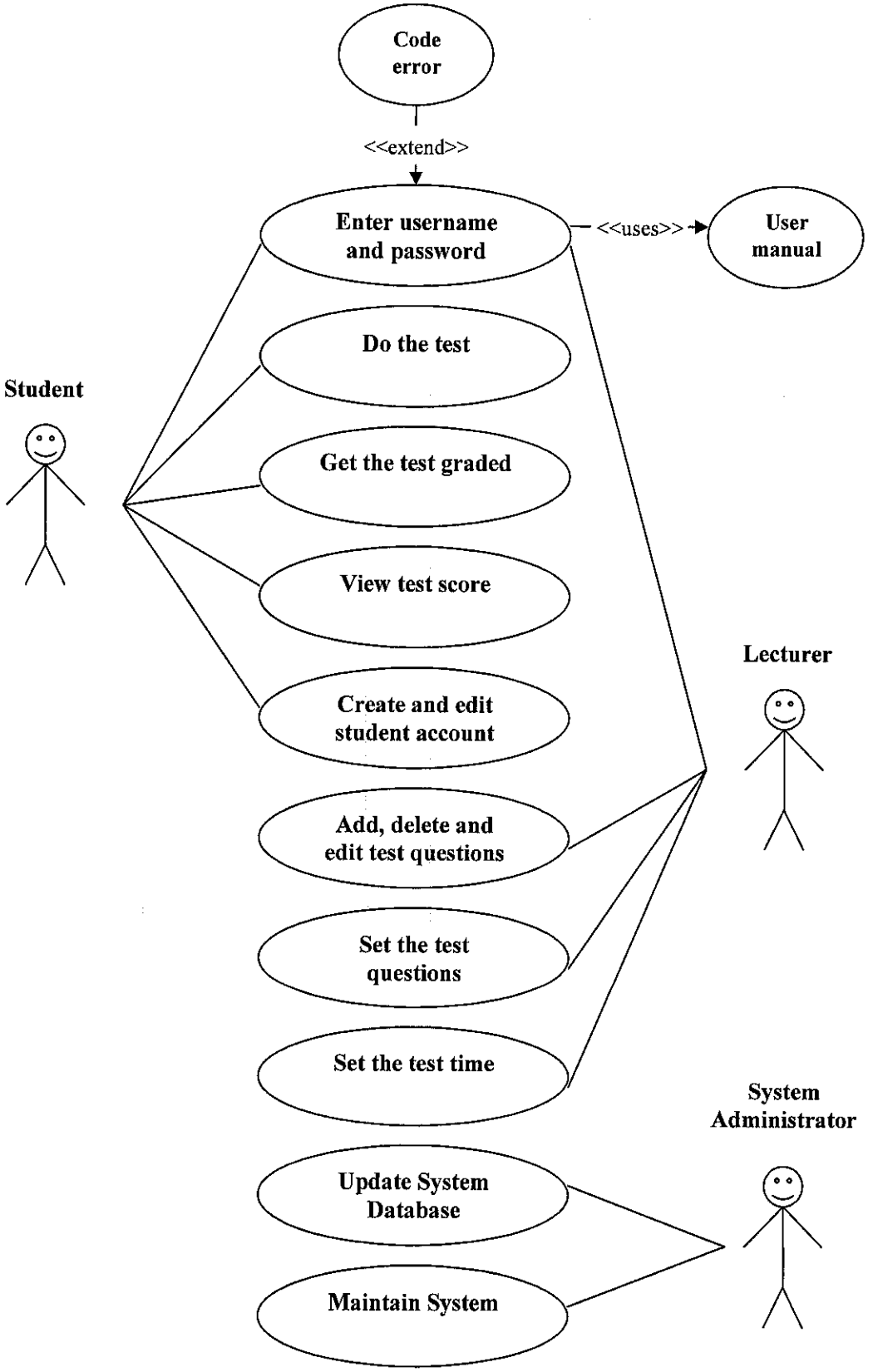
Firstly, the lecturers will enter their user ID and password in order to log-in to the system. The user ID and password will be kept in the lecturer's file database. The system will then verify the user ID and password entered. If both the user ID and password does not match the data in the student file, an error message will be displayed. Once log-in, the lecturers will create or open the test questions which is kept in the question banks. The lecturers too are allowed to add, edit, delete or save the created test questions in the question bank. All these questions will be classified and saved according to the relevant OOP chapters for easy viewing purposes.

Then, the lecturers will create or open a Test Template. This Test Template is actually used to set up a test where the test questions will be imported from the Question Bank. Once the test had been set, the lecturers have to allocate the appropriate test time for the test. Once completed, the test will be saved in the Test Template. The lecturers can even print the test questions for future reference.

3.2.3 Use Case Diagram

The author too had created a Use Case Diagram *(refer Figure 3.7)* which describes the behavior of the system as a whole. This diagram is used mainly during the analysis phase of a project where it is used to identify and partition system functionality. They separate this system into *actors* and *use cases.*

FIGURE 3.7 : USE CASE DIAGRAM



Description :-

Figure 3.7 above shows the Use Case Diagram of the Computerized JAVA Test For OOP Course. There are two items involve that are the actors and the use cases (process).

Actors represent roles that can are played by users of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems. The only criterion is that they must be external to the part of the system being partitioned into use cases. They must supply stimuli to that part of the system, and the must receive outputs from it. For this Computerized JAVA Test, there are a total of three actors involved that are the student, the lecturer and the system administrator.

Use cases describe the behavior of the system when one of these actors sends one particular stimulus. This behavior is described textually. It describes the nature of the stimulus that triggers the use case; the inputs from and outputs to other actors, and the behaviors that convert the inputs to the outputs. The text of the use case also usually describes everything that can go wrong during the course of the specified behavior, and what remedial action the system will take.

3.3 TOOLS REQUIRED

The tools required in designing and developing this system are :-

3.3.1 Hardware Requirements :-

- *Computer: an IBM® or compatible PC with at least an Intel Pentium processor or equivalent*
- *Minimum of 32 MB RAM memory; 64 MB RAM or higher recommended; a minimum of 64 MB required for Windows 2000 and XP, and 128 MB RAM is recommended for XP.*
- *Minimum of 25 MB of available disk space, not including pagefile memory; 40-45 MB recommended*
- *Monitor: 14" or larger color monitor compatible*

3.3.2 Software Requirements :-

- *Microsoft Windows 95/98/2000/Me/XP or NT 4.0*
- *Microsoft Visual Basic (VB) Version 6.0*

CHAPTER 4
RESULT AND DISCUSSION

From the informal interview conducted between the author, the author's supervisor, UTP lecturers and students on August 11, 2004, some essential information have been gathered.

The current manual JAVA test involves two main group of users; UTP lecturers and UTP students. The lecturers need to set up a maximum of two sets of test questions manually and print them according to the number of students sitting for the JAVA test each semester. Once, the student had taken the test, the lecturers need to spent some time in marking and grading the test one by one. After complete marking, the lecturers will post the test grade in the UTP e-learning. The student will then access the e-learning in order to know their test results.

The main problem faced by the lecturers with the current manual JAVA test is that it consumes time and energy for the lecturers to mark the test papers manually one-by-one as compared to this Computerized JAVA Test where it is capable of grading the test automatically. Besides that, there is a risk of loss of documents with the current manual test procedure. The storage of the test grades is done manually where it is kept in a file. So, the possibilities of losing the documents are high due to unorganized way of handling the documents.

In addition, the lecturers noticed that with the manual JAVA test procedure, there is a tendency of copying and cheating during the test session. The JAVA test that are conducted manually in a paper-pen format often allow the students to copy each others answer easily as the questions are arranged accordingly in the question booklet. But, with the Computerized JAVA Test, this problem could be eliminated as the test questions will be randomized and saved in the questions bank.

The author conducted this project by analyzing the current manual procedure of the JAVA test. The author uses the fact-finding techniques such as distributing questionnaires and interviewing the targeted users in finding relevant information or argument faced by the users. Before designing the system, the author had interviewed some lecturers and students of Universiti Teknologi PETRONAS. The main reason of conducting this interview session is to get some information on the requirements of this Computerized JAVA Test. The author had prepared a set of interview questions *(refer Appendix 1.0)* which was used to interview the targeted users. Based the interview conducted, the author had analyzed and produced a pie-chart *(refer to Figure 4.1)* that shows the statistics of users who agrees with this Computerized JAVA Test for OOP Course.

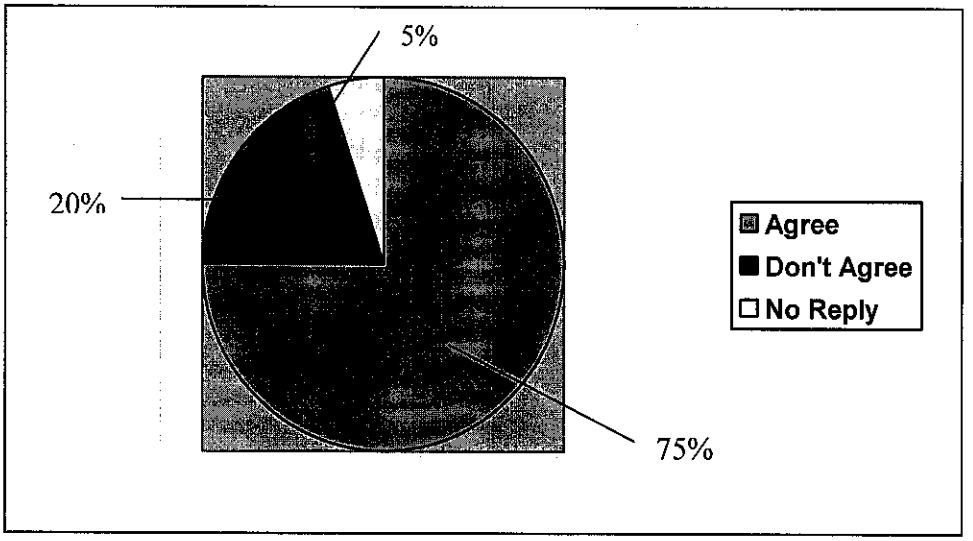


Figure 4.1 : Pie Chart that shows the statistics of users who agree and don't agree with the Computerized JAVA Test for Object – Oriented Programming (OOP) Course

The pie chart above shows the percentage of users who agree and disagree in designing this Computerized JAVA Test which is equip with an Integrated Question Bank. Based on the interview session conducted between the UTP students and lecturers,

about 75% of users agree with this Computerized Test system, while the other 20% of them disagree with this system. These 20% users want to remain with the manual paper-and-pen format test. The remaining 5% of the users did not reply to my questions directly. As for them, they don't mind having the test computerized or remain it as present.

In addition, the author too had prepared a set of questionnaires which will be used in analyzing the efficiency of this system once the system is being tested. Sample of the questionnaires were prepared separately for two types of targeted users, who are the UTP lecturers and the UTP students. The sample of questionnaires for both the students and the lecturers were placed in the Appendices section of this report. *(refer to Appendix 3.0 and Appendix 4.0).*

Since time constraint is one of the major challenges in completing this project (as discussed in Chapter 1), the author has planned her activities very carefully. It is because, without proper plan, the author may not finish the development of this Computerized Test System within the given time frame prepared by UTP Final Year Project (FYP) Committee. The author has prepared a timeline or also known as a Gantt Chart *(refer Appendix 5.0)* as a guideline in developing this system.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

As an overall, this Computerized JAVA Test for Object – Oriented Programming (OOP) Course help the lectures to plan and modify the coming semester's lesson plans to address the strengths and weaknesses of students before the new semester begins. The manual way of managing the system previously has incurred a few problems, which has become the interest of the author to solve it by introducing technology to computerize the test. The author will be conducting the preliminary data gathering, in order to be able to find out the problems faced by the current users and the solutions the users are expecting in order to solve the problem. An initial research will be done to gather information concerning this project including the security issues, software development methodologies, design using the UML approach and the development of the whole version of the system.

Besides that, this system will also be equipped with an integrated question bank where it will allow the lecturers to add, delete, edit and save the test questions easily. On the other hand, the students will be able to sit for the test according to the test question allocated in the question bank. This could help to reduce the problem of cheating and copying each other's answer as the test question will be randomized with the integrated question bank. Once the students had completed the test according to the time allocated, they can have their test graded automatically.

Generally, this system is an effective and user-friendly system because the users are satisfied with the overall performance of the system and managed to reduce the problems faced by the current users.

The advantages of using this system are:

- Helps to deter cheating as the test questions will be randomized with the integrated question bank
- Able to save the lecturers time as the test papers can be graded automatically
- Able to view the students performance easily and give more concentration on weak students
- By automating this system, the risk of losing the documents can also be reduced or prevented by creating a paperless working environment

5.2 RECOMMENDATION

This section discusses the recommendations for future enhancement of this system. There are three (3) recommendations suggested as the future enhancement of this system. They are as follows:

5.3.1 Automatically trigger an email notification to notify the students on the exact time and venue where the test will be conducted

An automatic prompt email is recommended to automatically trigger an email notification to the students involved with the JAVA test. This is to notify and remind the students on the exact time and venue where the test will be conducted. Besides that, if there is a change in the test schedule, it will be easier for the lecturers to inform the students before hand.

5.3.2 Enhance the current interface

Since this Computerized JAVA Test focuses only on objectives and true - and-false questions, it is recommended that in future, this system can be

enhanced to include short answers and essay questions. This will make the test be more challenging and in line meeting the UTP motto which is to produce a well-rounded students.

5.3.3 A modified web-based system

This Computerized JAVA Test for OOP Course is a standalone system that was built by using Microsoft Visual Basic (Version 6.0). It is suggested that, in future this system will be converted from a standalone system to a web-based system where it will be designed using Microsoft Visual Basic.NET and can be accessed in any labs in UTP.

REFERENCES

1. Dr. James B. Olsen, Chief Scientist at Alpine Media Corporation in Orem, Utah *"Guidelines for Computer-Based Testing"*, May/June 2000 <http://www.isoc.org/oti/articles/0500/olsen.html>

2. CTB/McGraw-Hill. *"I-know Online Assessment System"*. University of Technology, Sydney (UTS), 2004 <http://www.ctb.com/product/>

3. Ratna, A. A. P. and P. Raymonth, et al. *"Distance E-learning implementation and analysis on Jarkom-Online evaluation system"*. World Conference on Educational Multimedia, Hypermedia and Telecommunications, 2003

4. Trigwell K. 6 August 2004. *"Online Assessment"*. University of Technology, Sydney (UTS) < http://www.iml.uts.edu.au/assessment/online/>

5. R. Neill Johnson, Diane M. Enerson, and Kathryn M. Plank April 24, 1997. *"Computerized Testing Roundtable"* The Pennsylvania State University <http://www.psu.edu/celt/largeclass/comptest.html>

6. Arnow, D. and O. Barshay. *"On-line programming examinations using WebToTeach"*. Conference on integrating technology into computer science education (ITiCSE), Crocow, Poland, 1999

7. Catherine Dhanjal. 2 April 2004. *"Online Student Assessment"*. Pembrokershire College, United Kingdom, 1999-2004 <http://www.pressbox.co.uk>

8. Kay Woodfield, Rigby School District, Idaho. 2004. *"Getting On Board With Online Testing"*. ETC Group LLC <http://www.thejournal.com/magazine/vault/A4297.cfm>

9. Cannon, R.A. and Newble, D. 1983 "A *Handbook for Clinical Teachers, Lancaster"*. MTP Boston : p 97-105.

10. Kenneth E. Kendall and Julie E. Kendall. *"System Analysis and Design 5th Edition"*. Prentice-Hall International Inc, 2003.

APPENDICES

APPENDIX 1.0
SAMPLE OF INTERVIEW QUESTIONS

UNIVERSITI TEKNOLOGI PETRONAS

SAMPLE OF INTERVIEW QUESTIONS

1. Which year are you in now?

Year 1 : ☐ Year 2 : ☐ Year 3 : ☐

Year 4 : ☐ Year 5 : ☐ Others : _____

2. Have to taken Object-Oriented Programming (OOP) course before?

YES ☐ **NO** ☐

3. When did you take the OOP course?

Year 1 : ☐ Year 2 : ☐ Year 3 : ☐

Year 4 : ☐ Year 5 : ☐ Others : _____

4. What programming skill did you learned during the OOP course?

_____

5. How was the OOP test conducted?

_____

_____

6. How long do the lecturers take to mark your test papers manually?

_____

_____

7. Do you prefer to do the OOP test manually in a paper-pen format or computerized? Why?

_____

_____

_____

8. When using the computerized JAVA Test, do you prefer to get your test result on the spot once completing the test or otherwise? Why?

_____

_____

_____

9. If you are given a choice, what type of questions will you choose for your test? Is it MCQ, true/false, essays or short answers?

_____

_____

APPENDIX 2.0
USER INTERFACE DESIGN

1. Lecturers

1. Authentication Verification by Lecturers

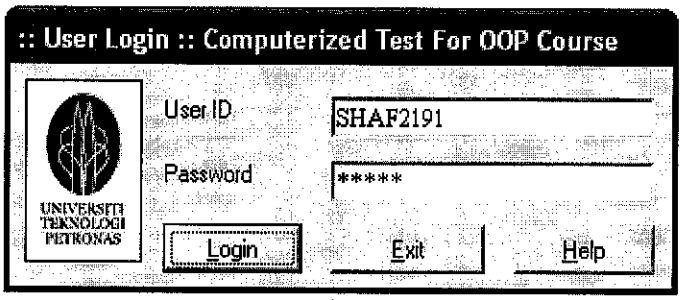


Figure 1.1 : Lecturer Login Form

- Lecturer has to enter their User ID and Password in order to login to the system.
- The "Exit" button is used to exit from the system.
- The "Help" button is used to help a user if he/she do encounter login problem.

2. Open Question Bank to add new sets of test question

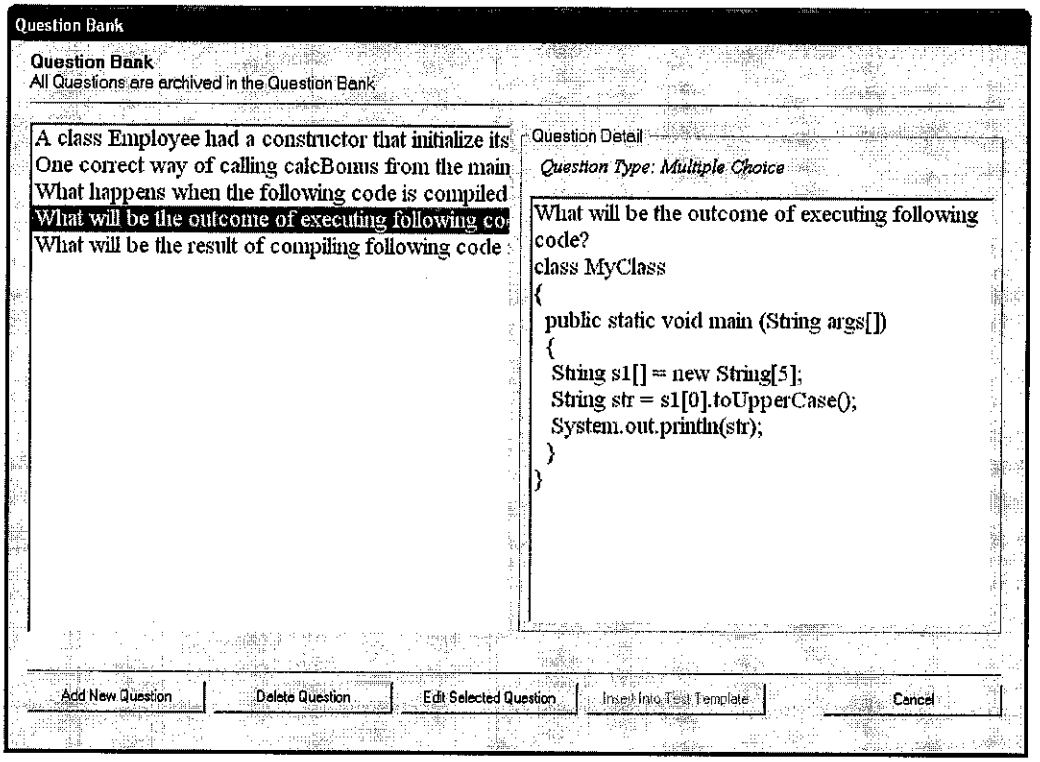


Figure 1.2 : Question Bank Form

- Question Bank is used to add, edit, delete and save the test questions according to their relevant chapters which will be included in the Test Template later. The lecturers can view the saved questions from the question bank itself or add new chapters into the question bank easily.

- To add new chapter into the question bank, the lecturer have to click *<File>*, *<New>*, *<Question Bank>*. The Question Bank template as shown above will appear. The lecturer need to enter the file name, example : Chapter1 – Principles of OOP.tst, and click on the "Save" button in order to add a new chapter into the question bank..

- To open existing chapter from the question bank, the lecturer need the click *<File>*, *<Open>*, *<Question Bank>*. The same question bank template as shown above will appear. The lecturer can choose any chapter they wish and click on the Open button in order to open and view the test questions saved on each chapter.

3. Add or Edit Questions in the Question Bank

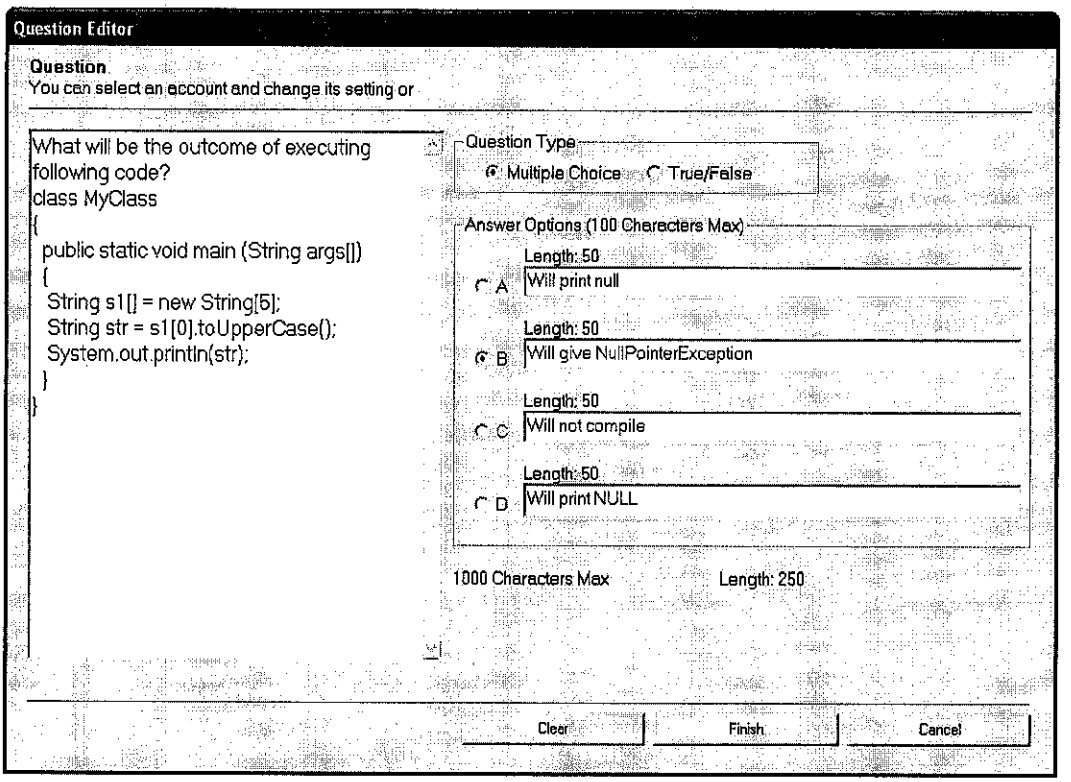


Figure 1.3 : Question Editor Form

- The Question Editor Form as shown above is used add or edit test questions from the question bank. A maximum of 1000 characters can be displayed in the question text box while a maximum of 100 characters can be displayed in the answer option test box.
- The lecturer has an option to choose the type of question to be created for a particular test. (multiple choice or true/false).

- Once the question and the answer option had been created, the lecturer need to select the correct answer for that picky question.
- The "Clear" button is used to clear the question from the Question Editor Form.
- The "Finish" button is used to complete the operation.
- The "Cancel" button is used to cancel or leave the Question Editor Form.

4. Set the OOP test paper

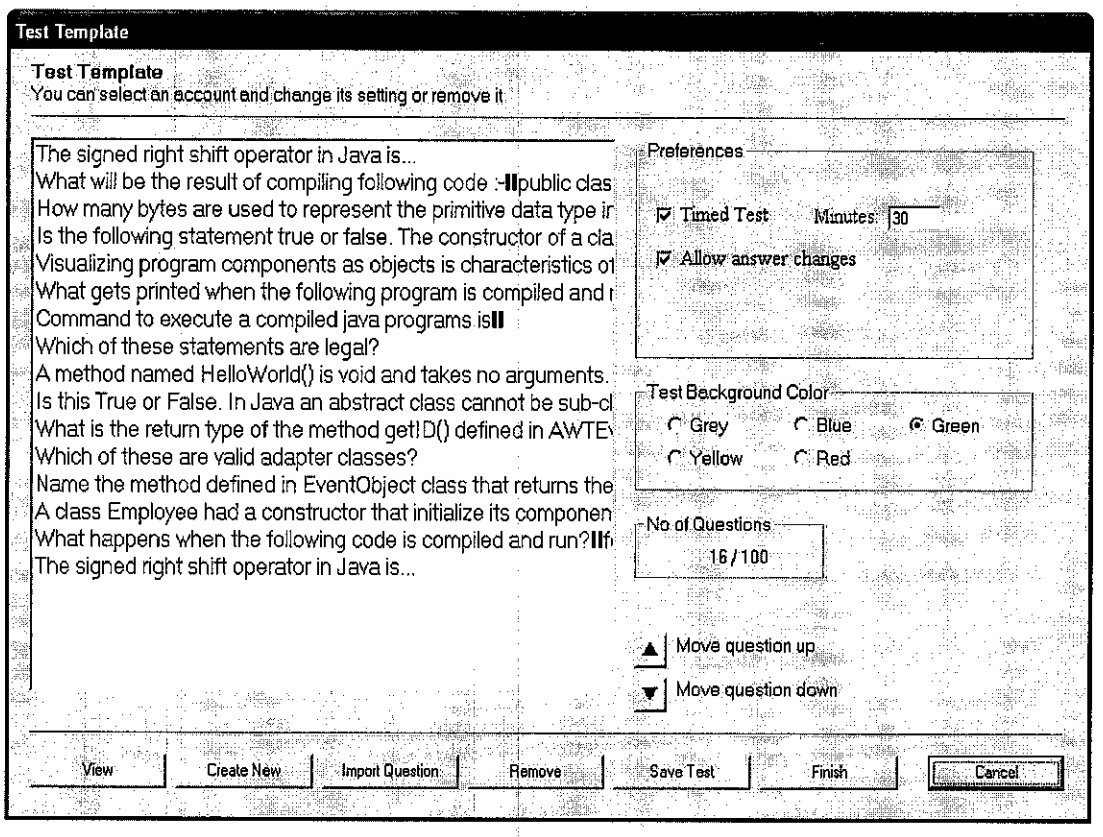


Figure 1.4 : Test Template Form

- The Test Template Form above allows the lecturer to set the OOP Test paper for a particular test. The lecturer will have to click the 'Import Question' button to import questions from the Question Bank which were arranged according to the OOP chapters.

- The "Create New" button from this template gives the privilege to the lecturers to add new test questions directly into this test template without using the Question Bank.
- The "View" button allows the lecturer to view the whole contents of the test questions including the answer option. I also allow the lecturer to do any edition on that particular test question.
- The "Remove" button is used to remove or delete questions from the Test Template.

4. About My Application

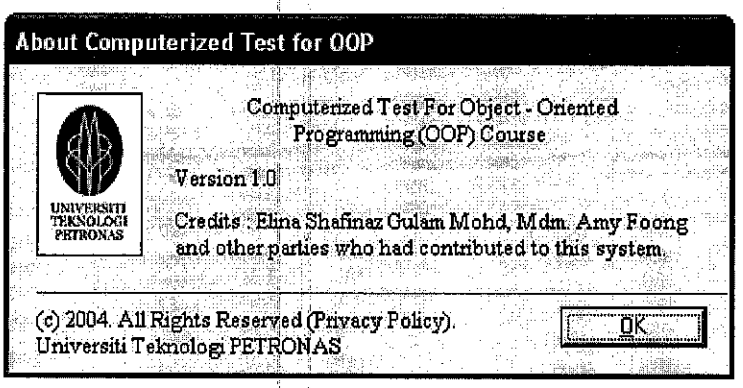


Figure 1.5 : About Computerized Test Form

- The form shown above provides some relevant information regarding this Computerized JAVA Test with Integrated Question Bank.

2. Students

1. Authentication Verification by Students

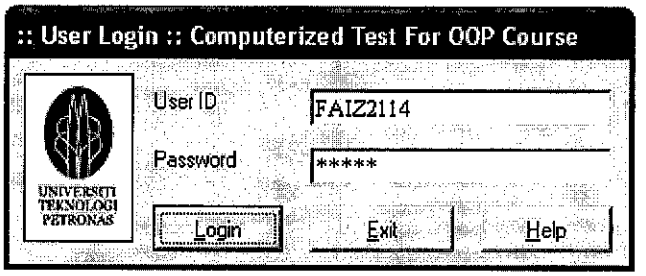


Figure 2.1 : Student Login Form

- Student has to enter their User ID and Password in order to login into the system.
- The "Exit" button is used to exit from the system.
- The "Help" button is used to help a user if he/she do encounter login problem.

2. Open Test File

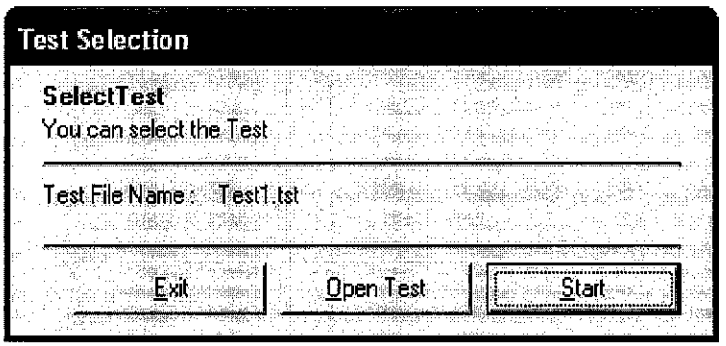


Figure 2.2 : Open Test Form

- Once login, this open test from will be displayed.
- The "Open Test" button allows the student to select the test from the Test File. Once selected, the name of the test will appear in the Test File Name column as shown above. Example ; Test File Name : Test1.tst
- Then, the student will have to click the "Start" button in order to start answering the test.

3. During the Test Operation

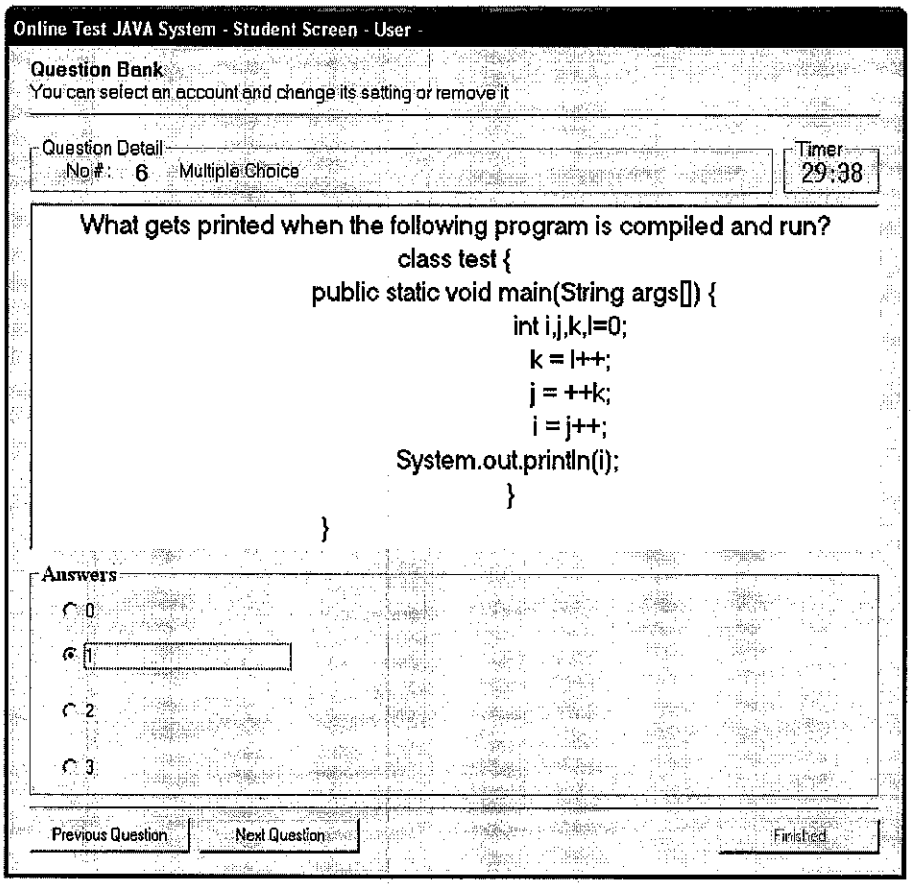


Figure 2.3 : Test Template Form

- The form above shows the test template form which will be used by the student in answering the JAVA test paper.
- During the test operation, the student will click on their preferred answer option as shown in the answer dialog box above.
- The "Next Question" button will display the next test question as arranged in the Test Template. (Students are only allowed to press the "Next Question" button, once they have answered the current test question).

- The "Previous Question" button will display the previous test questions that had already been answered by the student. This button can be used by the student to recheck their test paper before clicking the "Finished" button.

4. Display Student's Test Result

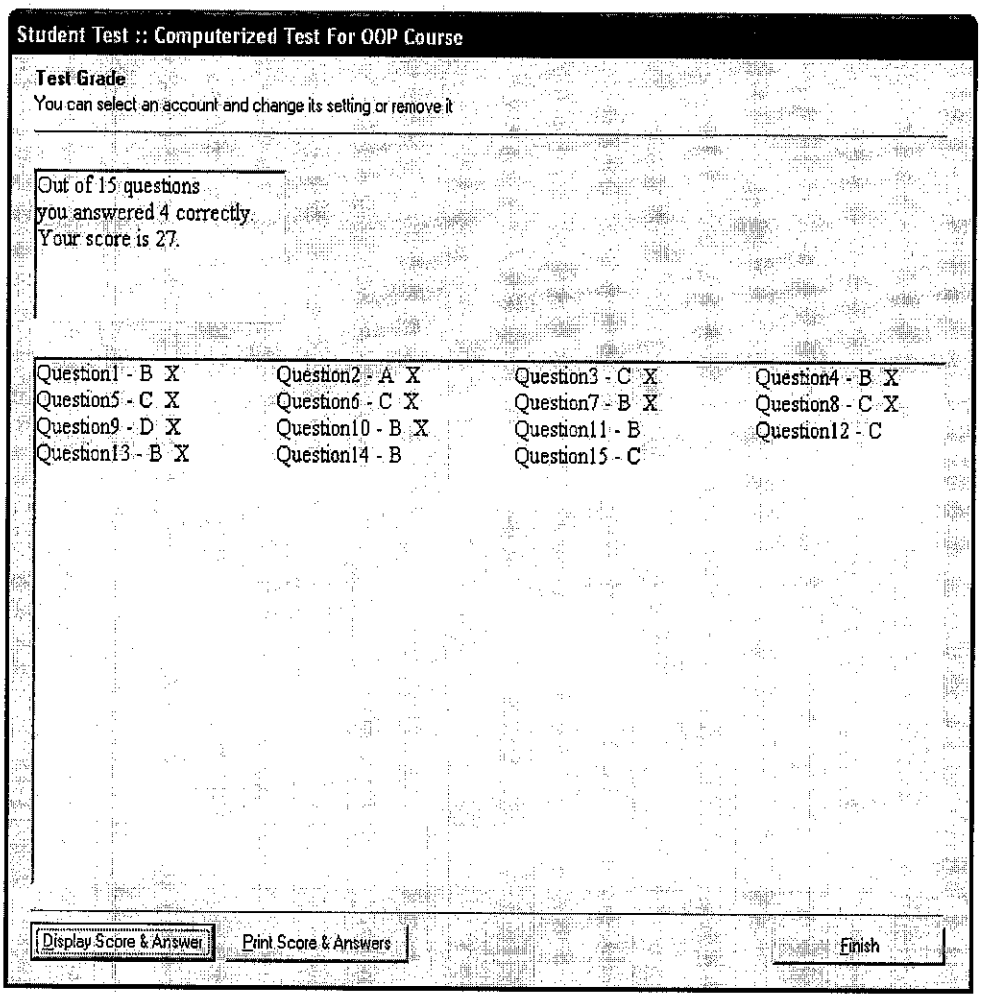


Figure 2.4 : Test Result Template

- The screenshot above shows the test result that the student had taken.
- The student has to click the "Display Scores and Answers" button to display the result of his/her test.

- The "Print Score and Answers" button is used to print the test result for future references.

5. Student's Test Scores throughout the Semester

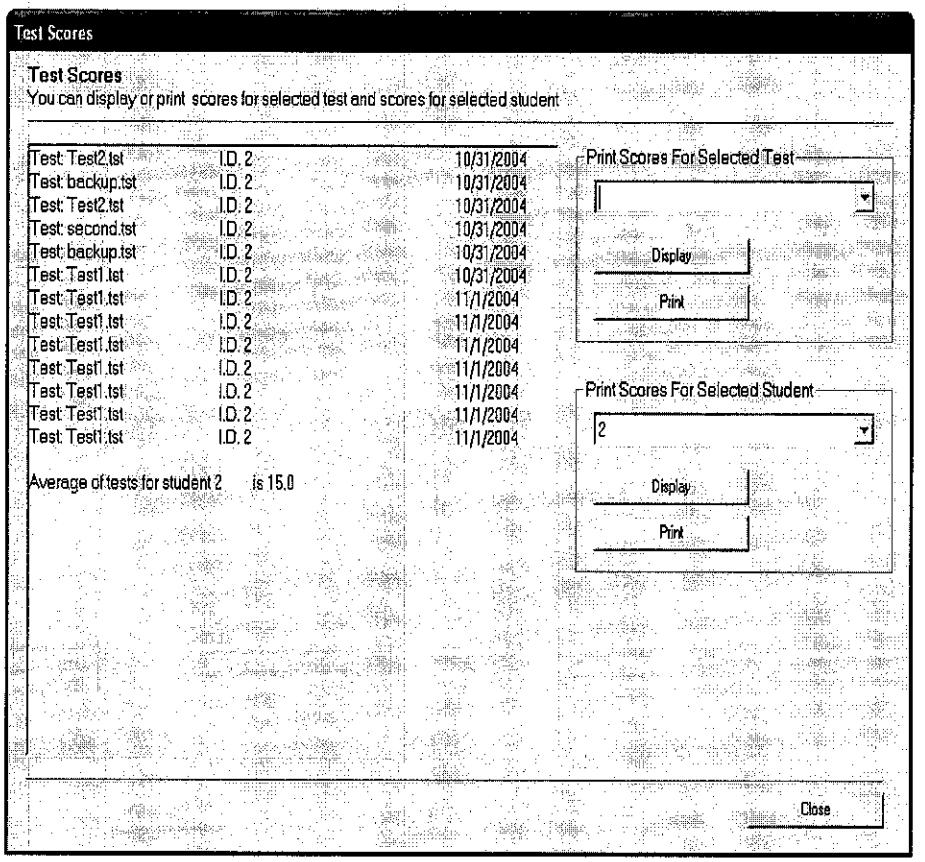


Figure 2.5 : Test Score Template

- The screen shot above is used to display all the test scores taken by a student throughout the whole semester.

APPENDIX 3.0
SAMPLE OF QUESTIONNAIRES - STUDENT

UNIVERSITI TEKNOLOGI PETRONAS

SAMPLE OF QUESTIONNAIRES - STUDENT

Course : _____ **Year :**_____

1. Have u sited and used the Computerized JAVA Test in the labs before?

   YES ☐ **NO** ☐

2. When was the last time you sited for the Computerized JAVA Test?
   State your Year and Semester : _____

3. What is your opinion with the Computerized JAVA Test?

   _____

   _____

   _____

4. Is the system user-friendly?

YES ☐ **NO** ☐

If your answer in **NO**, state your reason :-

_____

_____

5. Do you encounter any difficulty/ problem in doing that JAVA Test?

YES ☐ **NO** ☐

If your answer if YES, state your reason :-

_____

_____

6. Do you print your test results?

YES ☐ **NO** ☐

7. To help with specification, what extra features would you like to see on the Computerized JAVA Test that the manual test don't have?

_____

_____

_____

8. What are your suggestion / opinion in improving the system in future?

_____

_____

_____

9. Any further comments ?

_____

_____

_____

Thank you very much for taking the time to fill in this questionnaire. Your comments will be of value in helping me to complete my research.

APPENDIX 4.0
SAMPLE OF QUESTIONNAIRES - LECTURER

UNIVERSITI
TEKNOLOGI
PETRONAS

SAMPLE OF QUESTIONNAIRES - LECTURER

Position : _____ **Year :** _____

1. What is your opinion with the Computerized JAVA Test?

   _____

   _____

   _____

2. Is the system user-friendly?

   YES ☐ **NO** ☐

3. Does the system save your time and energy in creating questions as compared to the manual test procedure?

   YES ☐ **NO** ☐

4. Does this Computerized JAVA Test able to eliminate the copying and cheating problem?

YES ☐ **NO** ☐

5. Do you encounter any difficulty / problems in using the system?

YES ☐ **NO** ☐

If your answer is **YES**, state your reason :-

_____

_____

6. Do the question banks help you in adding, deleting or editing the test questions?

YES ☐ **NO** ☐

7. How does this computerized system affect the overall performance of the students as compared to the paper-and-pen format test?

_____

_____

_____

8. What are your suggestions / opinions in improving this system in future?

_____

_____

_____

9. To help with specification, what extra features would you like to see on this system that the manual test don't have?

_____

_____

_____

10. Any further comments?

_____

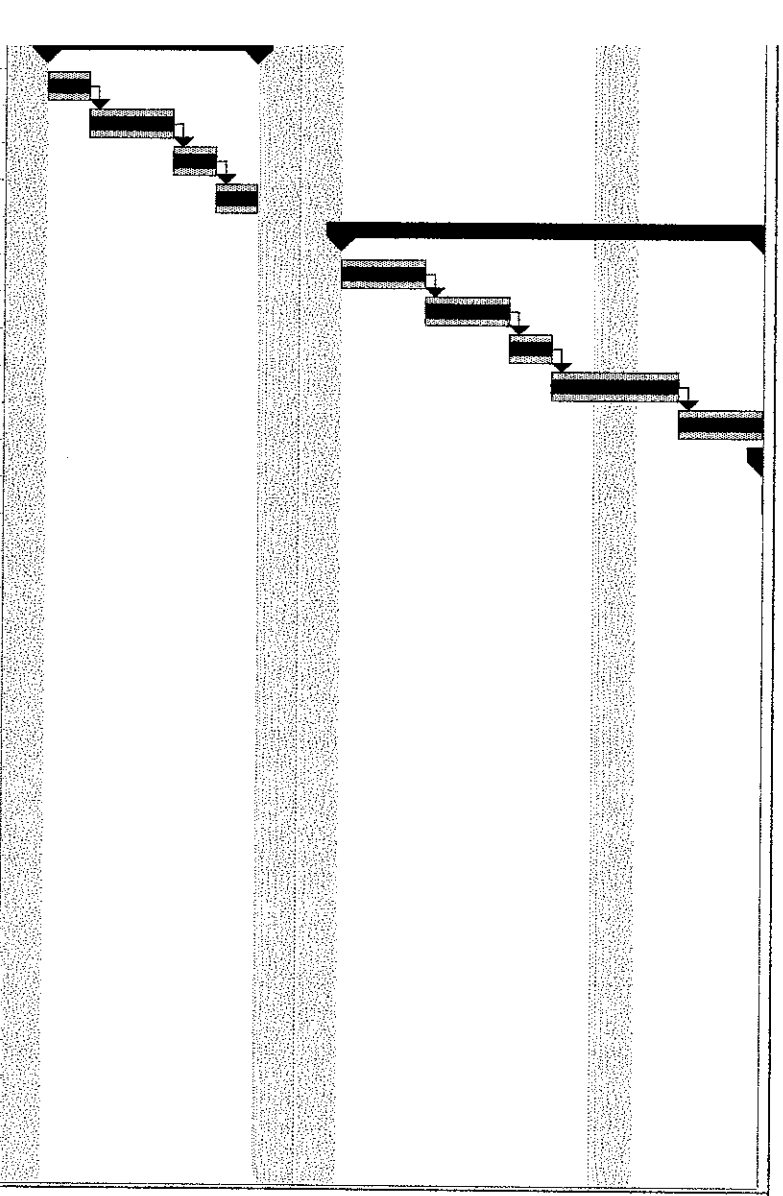
_____

_____

Thank you very much for taking the time to fill in this questionnaire. Your comments will be of value in helping me to complete my research.

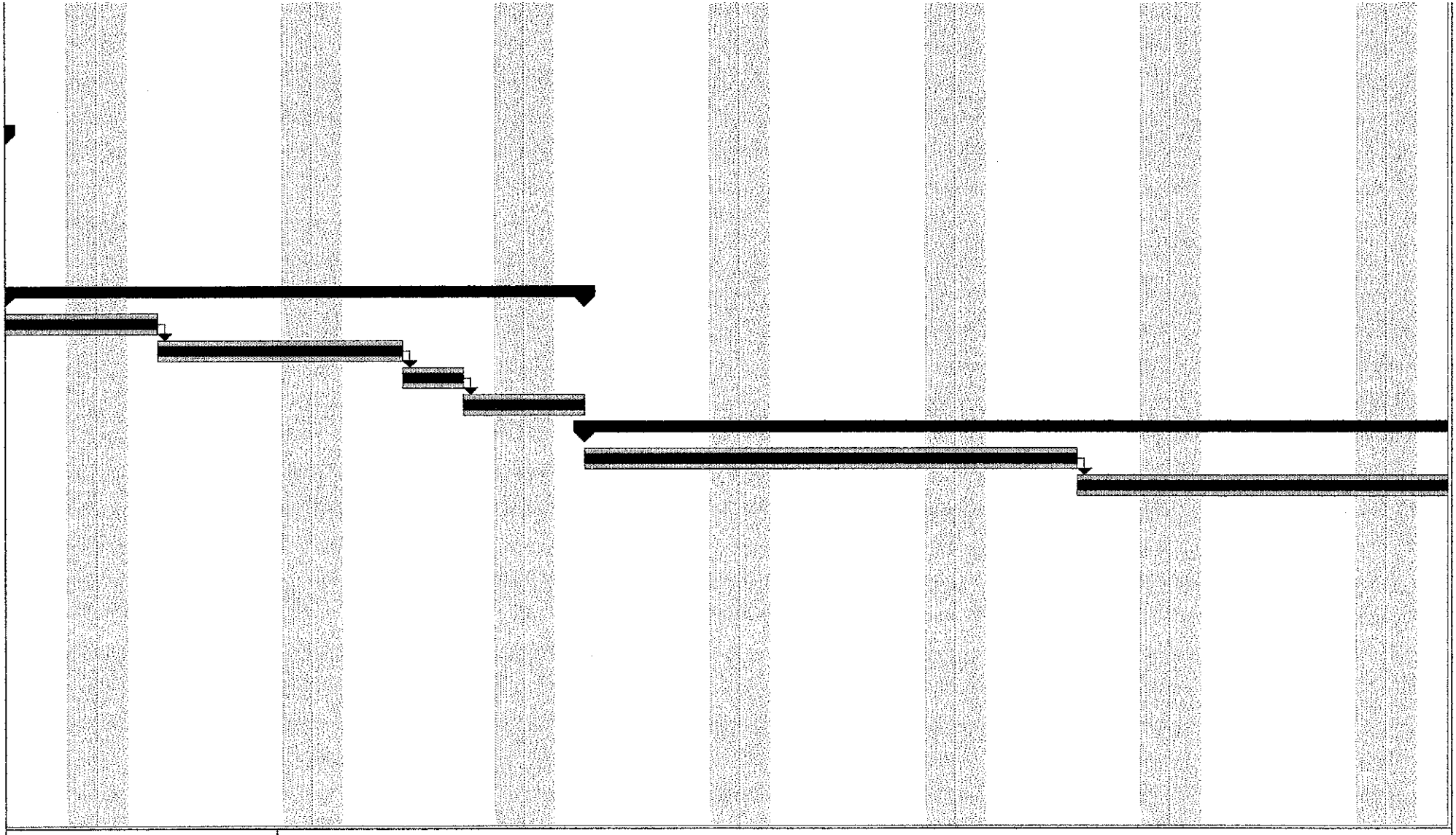
APPENDIX 5.0
GHANTT CHART

2	✓	1.1 Meet Supervisor	1 day	Mon 7/19/04	Mon 7/19/04
3	✓	1.2 Identify Objectives, Problems, Opportunities	2 days	Tue 7/20/04	Wed 7/21/04
4	✓	1.3 Gather User Requirement	1 day	Thu 7/22/04	Thu 7/22/04
5	✓	1.4 Submit Preliminary Report	1 day	Fri 7/23/04	Fri 7/23/04
6	✓	**2. System Analysis**	**9 days**	**Mon 7/26/04**	**Wed 8/4/04**
7	✓	2.1 Determine task duration, resources and dependencies	2 days	Mon 7/26/04	Tue 7/27/04
8	✓	2.2 Project Scheduling - Prepare Gantt Chart	2 days	Wed 7/28/04	Thu 7/29/04
9	✓	2.3 Consultation with Users	1 day	Fri 7/30/04	Fri 7/30/04
10	✓	2.4 Develop Detail User Requirement	2 days	Sat 7/31/04	Mon 8/2/04
11	✓	2.5 Define System Inputs, Processes, Outputs, Interfaces	2 days	Tue 8/3/04	Wed 8/4/04
12	✓	**3.System & Software Design**	**13 days**	**Thu 8/5/04**	**Mon 8/23/04**
13	✓	3.1 Data Modelling	3 days	Thu 8/5/04	Mon 8/9/04
14	✓	3.2 Application Architecture	6 days	Tue 8/10/04	Tue 8/17/04
15	✓	3.3 Design User Interfaces	2 days	Wed 8/18/04	Thu 8/19/04
16	✓	3.4 Submit Progress Report	2 days	Fri 8/20/04	Mon 8/23/04
17	✓	**4. Coding & Development**	**22 days**	**Tue 8/24/04**	**Wed 9/22/04**
18	✓	4.1 Build Database	12 days	Tue 8/24/04	Wed 9/8/04
19	✓	4.2 Create Coding	10 days	Thu 9/9/04	Wed 9/22/04
20	✓	**5. System Testing**	**12 days**	**Thu 9/23/04**	**Sun 10/10/04**
21	✓	5.1 Integration Test	7 days	Thu 9/23/04	Fri 10/1/04
22	✓	5.2 User Acceptance Test	5 days	Mon 10/4/04	Sun 10/10/04
23	✓	**6. Implementation Phase**	**52 days**	**Mon 10/11/04**	**Mon 12/20/04**
24	✓	6.1 Prepare User Manual & Execute Training	4 days	Mon 10/11/04	Thu 10/14/04
25	✓	6.2 Demo my system to Mdm. Amy	1 day	Fri 10/15/04	Fri 10/15/04
26	✓	6.3 IT/IS Exhibition	2 days	Tue 10/19/04	Wed 10/20/04
27	✓	6.3 First Final Draft Submission	1 day	Thu 10/21/04	Thu 10/21/04
28	✓	6.4 EDX Exhibition	1 day	Mon 10/25/04	Mon 10/25/04
29	✓	6.5 First Final Draft Submission - 2 copies	1 day	Tue 11/2/04	Tue 11/2/04
30	✓	6.5 Oral Presentation	3 days	Mon 12/6/04	Wed 12/8/04
31	✓	6.6 Project Dissertation Submission	1 day	Mon 12/20/04	Mon 12/20/04



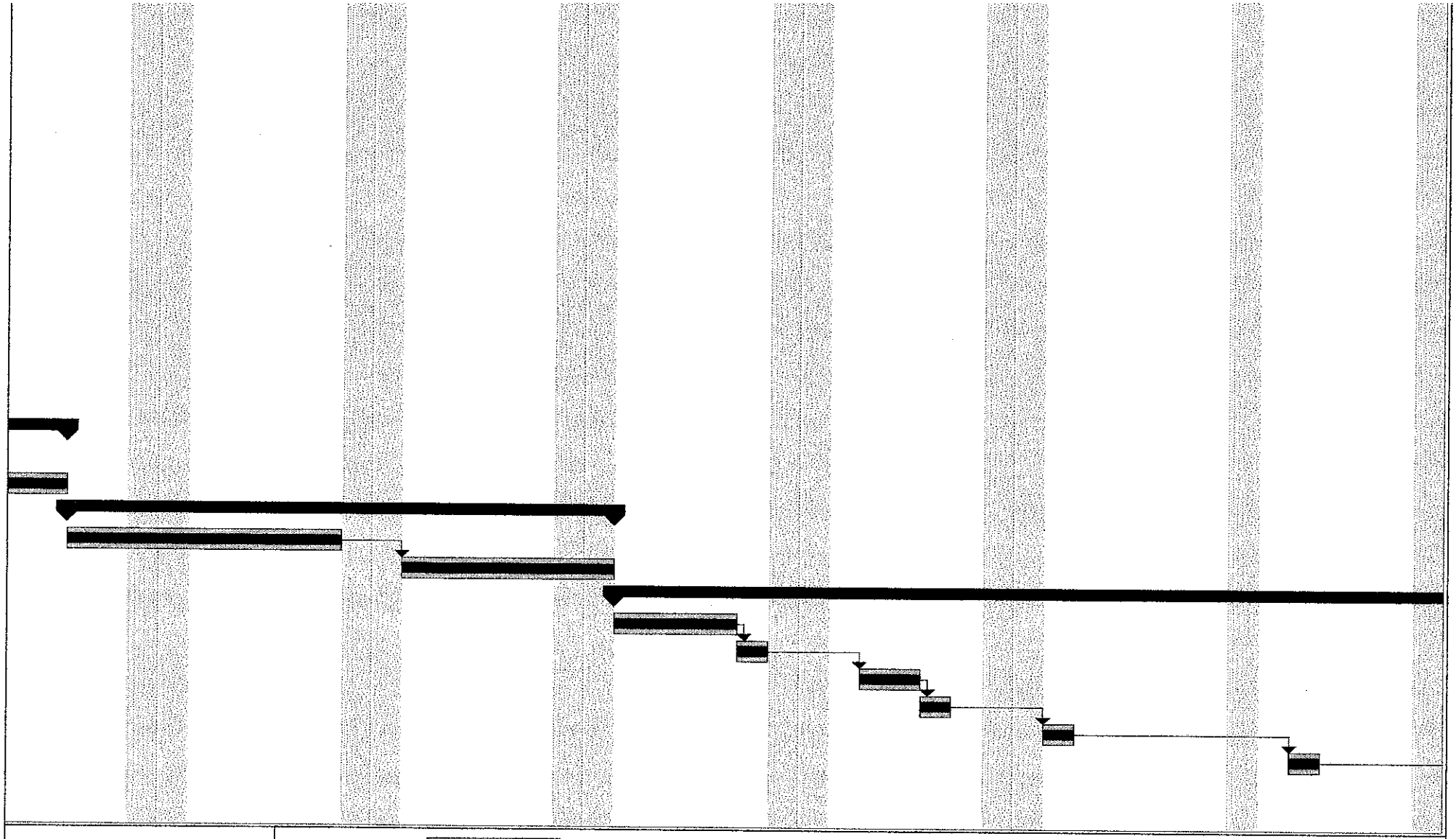
Project: Figure 4.8 - Gantt Chart
Date: Thu 12/9/04

Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline



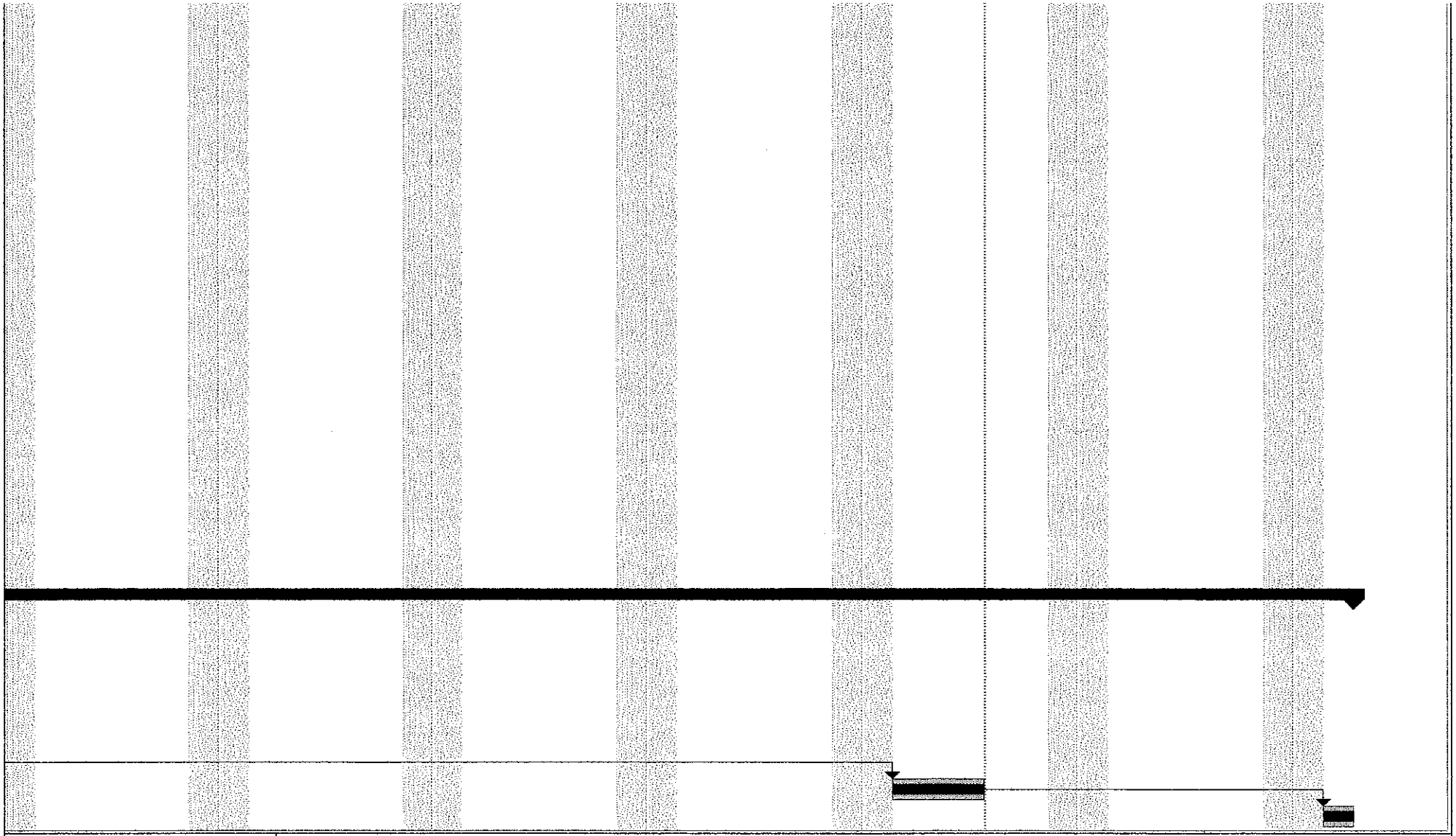
Project: Figure 4.8 - Gantt Chart
Date: Thu 12/9/04

Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline



Project: Figure 4.8 - Gantt Chart
Date: Thu 12/9/04

Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline



Project: Figure 4.8 - Gantt Chart
Date: Thu 12/9/04

Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline

APPENDIX 6.0
SOURCE CODE

1. Lecturer and Student Login Form

```
Private Sub Form_Load()
   'done'
   With datLogin
      .DatabaseName = App.Path & "\login.mdb"
      .RecordSource = "Login"
      .Refresh
   End With
End Sub
Private Sub cmdExit_Click()
   'done'
   End
End Sub

Private Sub cmdHelp_Click()
   'done'
   MsgBox ("Please consult the administrator if you have any problems for login")
End Sub

Private Sub cmdLogin_Click()

   Dim found As Boolean
   Dim Instructor As Boolean

   'reset found flag to false
   found = False

   'call procedures
   If CheckBlankLogin() = False Then Exit Sub
   If UserExist() Then
      If CheckPassword() = True Then
         StartMain
         Exit Sub
      Else
         MsgBox "Password is incorrect!", , "Warning!!"
         Exit Sub
      End If
   End If
   MsgBox "User ID was not found, try again.", , "Warning!"
End Sub
Private Function CheckBlankLogin() As Boolean
   'check for blank login fields
   If txtPassword.Text = "" Or txtUserID.Text = "" Then
      MsgBox "One of your login fields is blank, pleae try again.", , "Attention"
      CheckBlankLogin = False
      Exit Function
   End If
   CheckBlankLogin = True
End Function
Private Function UserExist() As Boolean
   'search userID to see if it exists
   datLogin.Recordset.MoveFirst
   userCode = datLogin.Recordset.Fields("UserID").Value
   Do Until found Or datLogin.Recordset.EOF
      userCode = datLogin.Recordset.Fields("UserID").Value
      If UCase(Trim(userCode)) = UCase(Trim(txtUserID.Text)) Then
         UserExist = True
         Exit Function
      Else
         datLogin.Recordset.MoveNext
      End If
   Loop
   UserExist = False
End Function
Private Function CheckPassword() As Boolean
      'only check password if user was found
      Password = Trim(datLogin.Recordset.Fields("Password").Value)
      Instructor = datLogin.Recordset.Fields("Instructor").Value
```

```
    If UCase(Trim(Password)) = UCase(Trim(txtPassword.Text)) Then
      With currentUser
        .userID = Trim(datLogin.Recordset.Fields("userID").Value)
        .Firstname = Trim(datLogin.Recordset.Fields("Firstname").Value)
        .Lastname = Trim(datLogin.Recordset.Fields("Lastname").Value)
        .ID = Trim(datLogin.Recordset.Fields("SSN").Value)
        .Password = Trim(datLogin.Recordset.Fields("Password").Value)
        .Instructor = Trim(datLogin.Recordset.Fields("Instructor").Value)
      End With
      CheckPassword = True
    Else
      CheckPassword = False
    End If
End Function
Private Sub StartMain()
  If currentUser.Instructor Then
    Load frmMain
    frmMain.Show
    Unload Me
  Else
    Load frmMain
    frmMain.Show
    frmChooseTest.Show
    Unload Me
  End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
  'close login database
  datLogin.Recordset.Close
End Sub
```

2. Main Form

```
Private Sub MDIForm_Load()
  Select Case currentUser.Instructor
  Case False
    'disable menu if user is not an instructur
    mnuNew.Visible = False
    mnuOpen.Visible = False
    mnuAccount.Visible = False
  Case True
  End Select
End Sub

Private Sub mnuAbout_Click()
  Load frmAbout
End Sub

Private Sub mnuAccount_Click()
Load frmAccount
End Sub

Private Sub mnuCreateTB_Click()
  ToggleMenu
  MenuCommand = "CreateQuestionBank"
  SetParentForm Me
  Load frmQuestionBank
End Sub

Private Sub mnuCreateTT_Click()
  currentTest = ""
  FlushQuestTest
  ToggleMenu
```

```
    MenuCommand = "CreateTestTemplate"
    SetParentForm Me
    Load frmtestTemplate
End Sub

Private Sub mnuEditYourAccount_Click()
    Load frmAccountSelfChange
End Sub

Private Sub mnuLogOut_Click()
    Unload frmMain
    Load frmLogin
    frmLogin.Show
End Sub
Private Sub mnuExit_Click()
    End
End Sub
Private Sub mnuOpenTB_Click()
    ToggleMenu
    MenuCommand = "OpenQuestionBank"
    SetParentForm Me
    frmQuestionBank.Show
End Sub

Private Sub mnuOpenTT_Click()
    ToggleMenu
    MenuCommand = "OpenTestTemplate"
    SetParentForm Me
    Load frmtestTemplate
End Sub

Private Sub mnuTestScores_Click()
    ToggleMenu
    MenuCommand = "OpenTestScores"
    Load frmTestScores
End Sub
```

3. Question Banks Form

```
Private Sub cmdAdd_Click()
    SetParentForm Me
    ParentFormCommand = "AddQuestionBank"
    Load frmQuestionEdit
End Sub

Private Sub cmdClose_Click()
    'ReturnToParent
    Select Case ParentForm
    '--------------------------
    Case "frmtestTemplate"
        frmtestTemplate.Show
    '--------------------------
    Case "frmMain"
        ToggleMenu
        FlushGlobalVariables
    '--------------------------
    End Select

    Unload Me
End Sub

Private Sub cmdEdit_Click()
    SetParentForm Me
    ParentFormCommand = "EditQuestionBank"
    If lisTestBank.Text = "" Then
        MsgBox "Click on a question first.", , "Attention"
```

```
      Exit Sub
    End If
    Me.Hide
    frmQuestionEdit.Show
End Sub

Private Sub cmdRemove_Click()
    Dim questRem As String

    'check to see if the question to be removed from the bank
    'has been clicked on
    If lisTestBank.Text = "" Then
      MsgBox "You have not selected a question to remove.", , "Warning!"
    Else

      'remove question from database
      questRem = questHold(lisTestBank.ListIndex + 1).quest
      With datBank.Recordset
      .MoveFirst
      .FindFirst ("question = '" & questRem & "'")
      .Delete
      .MoveFirst
      End With

      'remove question from bank list
      lisTestBank.RemoveItem lisTestBank.ListIndex

    End If
End Sub

Private Sub Form_Load()
    Call CenterThisFormOnScreen(Me)

    Select Case MenuCommand
    '-----------------------------------------------
    Case "CreateQuestionBank"
      CreateQuestionBank
    '-----------------------------------------------
    Case "OpenQuestionBank"
      OpenQuestionBank
    '-----------------------------------------------
    End Select

    Select Case ParentForm
    '---------------------------------
    Case "frmtestTemplate"
      Select Case ParentFormCommand
      Case "Import"
        cmdMoveQuestion.Enabled = True
        OpenQuestionBank
        'cmdAddToBank.Enabled = False
        'cmdRemoveFromBank.Enabled = False
        Exit Sub
      Case Else
      End Select
    '---------------------------------
    Case "frmMain"
      cmdMoveQuestion.Enabled = False
    '---------------------------------
    End Select

    'reset newDb to false
    newDB = False

End Sub
Private Sub CreateQuestionBank()

    Dim NewBank As Database, MyWS As Workspace
    Dim T1 As TableDef
    Dim T1Flds(1 To 7) As Field
```

```
Dim T1Idx As Index
Dim myRec As Recordset
Dim checkDIR As String

On Error GoTo DialogError

'open save dialog
With dlgFile
   .CancelError = True
   .DialogTitle = "Choose A Name For The Test Bank"
   .Flags = cdlOFNOverwritePrompt
   .Filter = "Question Bank Files|*.mdb"
   .ShowSave

   datBank.Database.Close

   'delete the file chosen if it exists
   If Dir(.FileName) <> "" Then
      Kill .FileName
   End If

   'set currentBank with file name
   currentBank = .FileName

   'set test bank label with file name
   'lblTestBank.Caption = "Test Bank: '" & .FileName & "'"

   'set Add to bank, remove from bank, and edit from bank
   'command buttons
   cmdAddToBank.Enabled = True
   cmdRemoveFromBank.Enabled = True
   cmdEditBank.Enabled = True

   'create new question bank database
   Set MyWS = DBEngine.Workspaces(0)
   Set NewBank = MyWS.CreateDatabase(.FileName, dbLangGeneral)

   Set T1 = NewBank.CreateTableDef("bank")

   Set T1Flds(1) = T1.CreateField("question", dbText, 250)
   Set T1Flds(2) = T1.CreateField("type", dbText, 1)
   Set T1Flds(3) = T1.CreateField("opt1", dbText, 50)
   Set T1Flds(4) = T1.CreateField("opt2", dbText, 50)
   Set T1Flds(5) = T1.CreateField("opt3", dbText, 50)
   Set T1Flds(6) = T1.CreateField("opt4", dbText, 50)
   Set T1Flds(7) = T1.CreateField("answer", dbText, 50)

   T1.Fields.Append T1Flds(1)
   T1.Fields.Append T1Flds(2)
   T1.Fields.Append T1Flds(3)
   T1.Fields.Append T1Flds(4)
   T1.Fields.Append T1Flds(5)
   T1.Fields.Append T1Flds(6)
   T1.Fields.Append T1Flds(7)

   Set T1Idx = T1.CreateIndex("question")
   T1Idx.Primary = True
   T1Idx.Unique = True
   T1Idx.Required = True
   Set T1Flds(1) = T1Idx.CreateField("question")
   T1Idx.Fields.Append T1Flds(1)
   T1.Indexes.Append T1Idx
   NewBank.TableDefs.Append T1

   'add a dummy record to prevent any db errors
   Set myRec = T1.OpenRecordset
   myRec.AddNew
   myRec("question") = "dummy"
   myRec("type") = "D"
   myRec("opt1") = "dummy"
```

```
        myRec("opt2") = "dummy"
        myRec("opt3") = "dummy"
        myRec("opt4") = "dummy"
        myRec("answer") = "d"
        myRec.Update
        myRec.Close

        'close new database
        NewBank.Close

        'open new database with data control
        datBank.DatabaseName = .FileName
        datBank.RecordSource = "bank"
        datBank.Refresh

        'reset questHold array to null
        For x = 1 To 200
          questHold(x).answerA = ""
          questHold(x).answerB = ""
          questHold(x).answerC = ""
          questHold(x).answerD = ""
          questHold(x).correctAns = ""
          questHold(x).quest = ""
          questHold(x).theType = ""
        Next x

        'set newDB flag to true
        newDB = True

        'clear test bank list box
        lisTestBank.Clear

        'clear large question display
        txtQuestDisp.Text = ""

        'load form to add questions to bank
        'Load frmAddToBank

    End With

DialogError:
    On Error GoTo 0
    Exit Sub

End Sub
Private Sub OpenQuestionBank()
    Dim i As Integer
    Dim x As Integer

    'On Error GoTo DialogError

    'open the open dialog
    With dlgFile
      .CancelError = True
      .DialogTitle = "Choose The Name Of The Test Bank"
      .Flags = 2
      .Filter = "Database Files|*.mdb"
      .ShowOpen

      'set database,label, and currentBank with file name
      datBank.DatabaseName = .FileName
      'lblTestBank.Caption = "Quiz Bank:  '" & .FileName & "'"
      currentBank = .FileName
    End With

    'enable add to bank, romove from bank, and edit bank
    'command buttons
    cmdAdd.Enabled = True
    cmdRemove.Enabled = True
    cmdEdit.Enabled = True
```

```
    'clear big question display and test bank list box
    txtQuestDisp.Text = ""
    lisTestBank.Clear

    'open database
    datBank.RecordSource = "bank"
    datBank.Refresh
    datBank.Recordset.MoveFirst

    'clear questHold array with null
    For x = 1 To 200
       questHold(x).answerA = ""
       questHold(x).answerB = ""
       questHold(x).answerC = ""
       questHold(x).answerD = ""
       questHold(x).correctAns = ""
       questHold(x).quest = ""
       questHold(x).theType = ""
    Next x

    'load questHold array with test bank database data
    Do Until datBank.Recordset.EOF
       i = i + 1
       With questHold(i)
          .quest = datBank.Recordset.Fields("question").Value
          .theType = datBank.Recordset.Fields("type").Value
          .answerA = datBank.Recordset.Fields("opt1").Value
          .answerB = datBank.Recordset.Fields("opt2").Value
          .answerC = datBank.Recordset.Fields("opt3").Value
          .answerD = datBank.Recordset.Fields("opt4").Value
          .correctAns = datBank.Recordset.Fields("answer").Value
          lisTestBank.AddItem (.quest)
       End With
       datBank.Recordset.MoveNext
    Loop

    'If i = 1 And Trim(datBank.Recordset.Fields("question").Value) = "dummy" Then
    '   newDB = True
    'End If

    'set question type label
    lblQuestType.Caption = "Question Type:"

DialogError:
    On Error GoTo 0
    Exit Sub
End Sub
Private Sub cmdMoveQuestion_Click()
    Dim x As Integer
    x = lisTestBank.ListIndex + 1
    With questTest(frmtestTemplate.lisTestQuests.ListCount + 1)
          .quest = questHold(x).quest
          .answerA = questHold(x).answerA
          .answerB = questHold(x).answerB
          .answerC = questHold(x).answerC
          .answerD = questHold(x).answerD
          .theType = questHold(x).theType
          .correctAns = questHold(x).correctAns
          frmtestTemplate.lisTestQuests.AddItem .quest
    End With

    Select Case ParentForm
    '---------------------------------------
    Case "frmtestTemplate"
       frmtestTemplate.Show
    '---------------------------------------
    Case Else
    '---------------------------------------
    End Select
```

```
    strImport = False
    Unload Me


End Sub

Private Sub lisTestBank_Click()
    'load question from bank list to large question display
    txtQuestDisp.Text = lisTestBank.Text

    'set the label to display correct question type
    Select Case questHold(lisTestBank.ListIndex + 1).theType
        Case "M"
            lblQuestType.Caption = "Question Type: " & "Multiple Choice"
        Case "T"
            lblQuestType.Caption = "Question Type: " & "True/False"
    End Select
End Sub

Private Sub lisTestBank_DblClick()
cmdEdit_Click
End Sub
```

4. Question Editor Form

```
Private Sub cmdCancel_Click()
    ReturnToParent Me
End Sub

Private Sub Form_Load()

    Call CenterThisFormOnScreen(Me)
    Call EnableAllControls
    Call FlushQuestionEditorValues

    Select Case ParentForm
    '--------------------------------
    Case "frmtestTemplate"
        Select Case ParentFormCommand
        Case "ViewQuestion"
            InitializeQuestionEditorValues (frmtestTemplate.lisTestQuests.ListIndex + 1)
            InitializeAllLengthCounters
        Case "CreateNewQuestion"

        End Select
    '--------------------------------
    Case "frmQuestionBank"
        Select Case ParentFormCommand
        Case "AddQuestionBank"
        Case "EditQuestionBank"
            Dim iTemp As Integer
            iTemp = Int(frmQuestionBank.lisTestBank.ListIndex + 1)
            InitQuestionEditorValuesFromQuestionBank frmQuestionEdit, iTemp

        End Select
    '--------------------------------
    End Select
    'ReturnToParent Me
End Sub
Private Sub EnableAllControls()
    optC.Enabled = True
    optD.Enabled = True
    optC.Visible = True
    optD.Visible = True

    txtOptC.Enabled = True
```

```
    txtOptD.Enabled = True
    txtOptC.Visible = True
    txtOptD.Visible = True

    lblOptCLength.Visible = True
    lblOptDLength.Visible = True
End Sub
Private Sub FlushQuestionEditorValues()
    txtQuestTemplate.Text = ""
    optMultiple.Value = False
    optTrueFalse.Value = False

    txtOptA.Text = ""
    txtOptB.Text = ""
    txtOptC.Text = ""
    txtOptD.Text = ""

    optA.Value = False
    optB.Value = False
    optC.Value = False
    optD.Value = False
End Sub
Private Sub InitializeQuestionEditorValues(i As Integer)

    txtOptA.Locked = False
    txtOptB.Locked = False
    With questTest(i)

        txtQuestTemplate.Text = .quest
        txtOptA.Text = .answerA
        txtOptB.Text = .answerB
        txtOptC.Text = .answerC
        txtOptD.Text = .answerD

        "
        Select Case .theType
        Case "M"
            optMultiple.Value = True
        Case Else
            optTrueFalse.Value = True
            optC.Enabled = False
            optD.Enabled = False
            txtOptC.Enabled = False
            txtOptD.Enabled = False

            optC.Visible = False
            optD.Visible = False
            txtOptC.Visible = False
            txtOptD.Visible = False

            lblOptCLength.Visible = False
            lblOptDLength.Visible = False
        End Select

        "
        Select Case .correctAns
        Case "A"
            optA.Value = True
        Case "B"
            optB.Value = True
        Case "C"
            optC.Value = True
        Case "D"
            optD.Value = True
        End Select

    End With

End Sub
Private Sub InitializeAllLengthCounters()
```

```
    lblOptALength.Caption = "Length: " & Len(txtOptA.Text)
    lblOptBLength.Caption = "Length: " & Len(txtOptB.Text)
    lblOptCLength.Caption = "Length: " & Len(txtOptC.Text)
    lblOptDLength.Caption = "Length: " & Len(txtOptD.Text)
    lblMainLength.Caption = "1000 Characters Max                    Length: " & Len(txtQuestTemplate.Text)
End Sub
Private Sub cmdFinish_Click()

    If ThereIsBlankValues() Then Exit Sub

    'for the use of Test Template
    Select Case ParentFormCommand
    '-----------------------------------------------------------------
    Case "ViewQuestion"
        AssignValuesIntoQuestTest
    '-----------------------------------------------------------------
    Case "CreateNewQuestion"
      'add question to question test array
      AddValuesIntoQuestTest Me
      frmtestTemplate.lisTestQuests.AddItem txtQuestTemplate
    Case "AddQuestionBank"
      AddValuesIntoQuestHold
    End Select
    '-----------------------------------------------------------------
    Select Case ParentForm
    Case "frmtestTemplate"
      frmtestTemplate.lblNoOfQuestions = frmtestTemplate.lisTestQuests.ListCount & " / 100 "
      If questTest(1).quest <> "" Then Enable_all_buttons frmtestTemplate
    Case "frmQuestionBank"

    End Select
    '-----------------------------------------------------------------


    ReturnToParent Me
End Sub
Private Function ThereIsBlankValues() As Boolean
    'check template for blank values

    If (optMultiple = False And optTrueFalse = False) Then GoTo BLANKED
    If (txtQuestTemplate.Text = "") Then GoTo BLANKED

    '---multiple type----'
    If (optMultiple) Then
      If (optA = False And optB = False And optC = False And optD = False) Then GoTo BLANKED
      If (txtOptA = "" And txtOptB = "" And txtOptC = "" And txtOptD = "") Then GoTo BLANKED
    End If

    '---true|false type----'
    If (optTrueFalse) Then
      If (optA = False And optB = False) Then GoTo BLANKED
    End If

    ThereIsBlankValues = False
    Exit Function

BLANKED:
    ThereIsBlankValues = True
    MsgBox "One of your fields is blank!", , "Warning!"
End Function
Private Sub AssignValuesIntoQuestTest()
    With questTest(frmtestTemplate.lisTestQuests.ListIndex + 1)
      .quest = txtQuestTemplate.Text
      .answerA = txtOptA
      .answerB = txtOptB
      .answerC = txtOptC
      .answerD = txtOptD

      If optMultiple.Value = True Then
        .theType = "M"
```

```
      Else
         .theType = "T"
      End If

      If optA Then
         .correctAns = "A"
      Else
         If optB Then
            .correctAns = "B"
         Else
            If optC Then
            .correctAns = "C"
            Else
               .correctAns = "D"
            End If
         End If
      End If
   End With
End Sub

Private Sub cmdClear_Click()
   Call FlushQuestionEditorValues
End Sub
Private Sub ManageQuestionType(myValue As Boolean)


   optC.Enabled = myValue
   optD.Enabled = myValue
   txtOptC.Enabled = myValue
   txtOptD.Enabled = myValue

   optC.Visible = myValue
   optD.Visible = myValue
   txtOptC.Visible = myValue
   txtOptD.Visible = myValue

   lblOptCLength.Visible = myValue
   lblOptDLength.Visible = myValue

   txtOptA.Locked = Not myValue
   txtOptB.Locked = Not myValue
   If myValue = False Then
      txtOptA.Text = "True"
      txtOptB.Text = "False"
      optC.Value = False
      optD.Value = False
   End If

   Select Case myValue
   Case True
      'Select Case strTempTestAnswer
      '   Case "A"
      '      optA.Value = True
      '   Case "B"
      '      optB.Value = True
      '   Case "C"
      '      optC.Value = True
      '   Case "D"
      '      optD.Value = True
      'End Select
   Case False
      'If optA = True Then
      '   strTempTestAnswer = "A"
      'ElseIf optB = True Then
      '   strTempTestAnswer = "B"
      'ElseIf optC = True Then
      '   strTempTestAnswer = "C"
      'ElseIf optD = True Then
      '   strTempTestAnswer = "D"
```

```
    'End If
  End Select

End Sub

Private Sub optMultiple_Click()
  ManageQuestionType True
End Sub

Private Sub optTrueFalse_Click()
  ManageQuestionType False
End Sub
Private Sub txtOptA_Change()
  'display number of characters in text box
  lblOptALength.Caption = "Length: " & Len(txtOptA.Text)
End Sub

Private Sub txtOptB_Change()
  'display number of characters in text box
  lblOptBLength.Caption = "Length: " & Len(txtOptB.Text)
End Sub

Private Sub txtOptC_Change()
  'display number of characters in text box
  lblOptCLength.Caption = "Length: " & Len(txtOptC.Text)
End Sub

Private Sub txtOptD_Change()
  'display number of characters in text box
  lblOptDLength.Caption = "Length: " & Len(txtOptD.Text)
End Sub

Private Sub txtQuestTemplate_Change()
  'display number of characters in text box
  lblMainLength.Caption = "1000 Characters Max                    Length: " & Len(txtQuestTemplate.Text)
End Sub
Public Sub AddValuesIntoQuestHold()
  'check for blank fields


  'If txtQuestion = "" Or txtOptA = "" Or txtOptB = "" Then
  '  MsgBox "One of your fields is blank.", , "Warning"
  '  Exit Sub
  'End If

  'delete dummy record if database is new
  If newDB Then
    frmInstructor.datBank.Recordset.MoveFirst
    frmInstructor.datBank.Recordset.Delete
    newDB = False
  End If

  'call sub to add record
  Call addrecord
  Unload Me
End Sub
Private Sub addrecord()

  Dim i As Integer

  i = frmQuestionBank.lisTestBank.ListCount + 1
  frmQuestionBank.lisTestBank.AddItem (txtQuestTemplate)

  'add new question to test bank database, and load question
  'hold array correct answer and question type
  With frmQuestionBank.datBank.Recordset
    .AddNew
    .Fields("question").Value = txtQuestTemplate
    If optMultiple.Value = True Then
      .Fields("type").Value = "M"
```

```
            questHold(i).theType = "M"
        Else
            .Fields("type").Value = "T"
            questHold(i).theType = "T"
        End If
        .Fields("opt1").Value = txtOptA
        .Fields("opt2").Value = txtOptB
        If txtOptC = "" Then
            .Fields("opt3").Value = " "
        Else
            .Fields("opt3").Value = txtOptC
        End If
        If txtOptD = "" Then
            .Fields("opt4").Value = " "
        Else
            .Fields("opt4").Value = txtOptD
        End If
        If optA.Value = True Then
            .Fields("answer") = "A"
            questHold(i).correctAns = "A"
        Else
            If optB.Value = True Then
                .Fields("answer") = "B"
                questHold(i).correctAns = "B"
            Else
                If optC.Value = True Then
                    .Fields("answer") = "C"
                    questHold(i).correctAns = "C"
                Else
                    .Fields("answer") = "D"
                    questHold(i).correctAns = "D"
                End If
            End If
        End If

        .Update
        .MoveFirst

    End With

    'load question and answers to questHold array
    With questHold(i)
        .quest = txtQuestTemplate
        .answerA = txtOptA
        .answerB = txtOptB
        .answerC = txtOptC
        .answerD = txtOptD
    End With

End Sub
```

5. Test Template Form

```
Dim createCancel As Boolean
Dim questNum As Integer

Private Sub cmdSaveTest_Click()
    If currentTest = "" Then
        If currentTest = "" Then
            Call createTest
        End If
    End If
    If lisTestQuests.ListCount > 0 Then
        Call SaveTest
    End If
End Sub
```

```
Private Sub Form_Load()
   Call CenterThisFormOnScreen(Me)
   Call FlushQuestTest

   Select Case ParentForm
   '-----------------------------------------------------------------
   Case "frmMain"
      Select Case MenuCommand
      Case "CreateTestTemplate"
         UpdateNoOfQuestions
         DisableSomeControls False
         lisTestQuests.Clear
         'set defaults for new test
         chkTimed.Value = False
         txtMinutes.Text = ""
         chkAllowGoBack.Value = False
         optGrey = True
      Case "OpenTestTemplate"
         OpenTestTemplate
         UpdateNoOfQuestions
      End Select
   '-----------------------------------------------------------------
   Case "frmQuestionBank"
   '-----------------------------------------------------------------
   Case Else
   End Select
End Sub

Private Sub cmdImport_Click()
   SetParentForm Me
   ParentFormCommand = "Import"
   Load frmQuestionBank
   Me.Hide
End Sub
Private Sub UpdateNoOfQuestions()
   lblNoOfQuestions.Caption = lisTestQuests.ListCount & " / 100 "
End Sub
Private Sub DisableSomeControls(myValue As Boolean)
   cmdReviewQuestion.Enabled = myValue
   cmdRemoveTestQuest.Enabled = myValue
   cmdUp.Enabled = myValue
   cmdDown.Enabled = myValue
   cmdSaveTest.Enabled = myValue
End Sub
Private Sub cmdReviewQuestion_Click()
   If ListWereNotSelected() Then Exit Sub
   SetParentForm Me
   ParentFormCommand = "ViewQuestion"
   Me.Hide
   Load frmQuestionEdit
End Sub
Private Sub cmdCreateNewQuestion_Click()
   'check for more than 100 questions
   If questNum > 100 Then
      MsgBox "You have already reached the maximum limit of 100.  " & _
           "Remove an existing question if you want to add a new one." _
           , , "Warning!"
      Exit Sub
   End If

   SetParentForm Me
   ParentFormCommand = "CreateNewQuestion"
   Load frmQuestionEdit
   Me.Hide
End Sub
Private Sub cmdRemoveTestQuest_Click()
   Dim i As Integer

   'make sure quesition is selected :: checked
```

```
    If lisTestQuests.Text = "" Then
        MsgBox "Click on a question first.", , "Attention"
    Else
        'delete current question from array
        For i = lisTestQuests.ListIndex + 1 To lisTestQuests.ListCount - 1
            questTest(i) = questTest(i + 1)
        Next i

        'remove question from listbox
        lisTestQuests.RemoveItem lisTestQuests.ListIndex
        UpdateNoOfQuestions
    End If
End Sub
Private Sub OpenTestTemplate()
    Dim i As Integer
    Dim userResponse As Integer
    Dim timed As String
    Dim minutes As Integer
    Dim goBack As String
    Dim theColor As String

    On Error GoTo dlgError

    'if test has changed check to see if user wants to save current
    'test before opening new one
    'If testHasChanged Then
    '   userResponse = MsgBox("The current test has changed since you last saved it. " & _
    '                   "Do you want to save it before you open another one?", _
    '                   vbYesNoCancel, "Warning!")
    '   If userResponse = vbYes Then
    '      Call SaveTest
    '   Else
    '      If userResponse = vbCancel Then
    '         Exit Sub
    '      End If
    '   End If
    'End If

    'select filename to open
    With dlgFile
        .CancelError = True
        .DialogTitle = "Choose Test To Open"
        .Filter = "Test Files|*.tst"
        .Flags = 2
        .ShowOpen

        Open .FileName For Input As #1
        currentTest = .FileName
        'lblCurTest.Caption = "Current Quiz: " & currentTest

        Call Enable_all_buttons(Me)

        lisTestQuests.Clear

        'load test question array from test file
        Do Until EOF(1)
            i = i + 1
            Input #1, questTest(i).quest
            Input #1, questTest(i).answerA
            Input #1, questTest(i).answerB
            Input #1, questTest(i).answerC
            Input #1, questTest(i).answerD
            Input #1, questTest(i).correctAns
            Input #1, questTest(i).theType
            lisTestQuests.AddItem questTest(i).quest
        Loop
        Close #1

        'load test options
        Open Left(.FileName, Len(.FileName) - 3) & "lyt" For Input As #1
```

```
        Input #1, timed
        Input #1, minutes
        Input #1, goBack
        Input #1, theColor
      Close

      If timed = "T" Then
        chkTimed.Value = 1
        txtMinutes = minutes
      Else
        chkTimed.Value = 0
        txtMinutes = ""
      End If
      If goBack = "T" Then
        chkAllowGoBack.Value = 1
      Else
        chkAllowGoBack.Value = 0
      End If
      If theColor = "G" Then
        'optGrey.Value = True
      Else
        If theColor = "R" Then
          'optRed.Value = True
        Else
          If theColor = "B" Then
            'optBlue.Value = True
          Else
            'optGreen.Value = True
          End If
        End If
      End If

    End With

dlgError:
    On Error GoTo 0
    Exit Sub


End Sub
Private Function ListWereNotSelected() As Boolean
    'make sure a question is selected
    If lisTestQuests.Text = "" Then
      NothingIsSelected = True
      MsgBox "Click on a question first.", , "Attention"
      Exit Function
    End If
    NothingIsSelected = False
End Function

Private Sub cmdCancel_Click()
    'FlushGlobalVariables
    'Initialize_Variables

    FlushGlobalVariables
    ToggleMenu
    Unload Me
End Sub

Private Sub lisTestQuests_DblClick()
cmdReviewQuestion_Click
End Sub

Private Sub SaveTest()

    Dim recordNum As Integer

    'make sure all files are closed
    Close
    'double check to see if there is at least 1 question
```

```
    If lisTestQuests.Text = "" Then
        MsgBox "Click on a question first.", , "Attention"
    Else
        'delete current question from array
        For i = lisTestQuests.ListIndex + 1 To lisTestQuests.ListCount - 1
            questTest(i) = questTest(i + 1)
        Next i

        'remove question from listbox
        lisTestQuests.RemoveItem lisTestQuests.ListIndex
        UpdateNoOfQuestions
    End If
End Sub
Private Sub OpenTestTemplate()
    Dim i As Integer
    Dim userResponse As Integer
    Dim timed As String
    Dim minutes As Integer
    Dim goBack As String
    Dim theColor As String

    On Error GoTo dlgError

    'if test has changed check to see if user wants to save current
    'test before opening new one
    'If testHasChanged Then
    '   userResponse = MsgBox("The current test has changed since you last saved it. " & _
    '                   "Do you want to save it before you open another one?", _
    '                   vbYesNoCancel, "Warning!")
    '   If userResponse = vbYes Then
    '      Call SaveTest
    '   Else
    '      If userResponse = vbCancel Then
    '         Exit Sub
    '      End If
    '   End If
    'End If

    'select filename to open
    With dlgFile
        .CancelError = True
        .DialogTitle = "Choose Test To Open"
        .Filter = "Test Files|*.tst"
        .Flags = 2
        .ShowOpen

        Open .FileName For Input As #1
        currentTest = .FileName
        'lblCurTest.Caption = "Current Quiz: " & currentTest

        Call Enable_all_buttons(Me)

        lisTestQuests.Clear

        'load test question array from test file
        Do Until EOF(1)
            i = i + 1
            Input #1, questTest(i).quest
            Input #1, questTest(i).answerA
            Input #1, questTest(i).answerB
            Input #1, questTest(i).answerC
            Input #1, questTest(i).answerD
            Input #1, questTest(i).correctAns
            Input #1, questTest(i).theType
            lisTestQuests.AddItem questTest(i).quest
        Loop
        Close #1

        'load test options
        Open Left(.FileName, Len(.FileName) - 3) & "lyt" For Input As #1
```

```
            If userResponse = vbCancel Then
               Exit Sub
            End If
         End If
      End If

      'get name for new test
      With dlgFile
         .CancelError = True
         .FileName = ""
         .DialogTitle = "Choose A Name For The Test"
         .Flags = 2
         .Filter = "Test Files| *.tst"
         .ShowSave

         If Dir(.FileName) <> "" Then
            Kill .FileName
         End If

         currentTest = .FileName
         'lblCurTest.Caption = "Current Test: " & currentTest
         Enable_all_buttons Me
         testHasChanged = False

      End With

      'add test name to test table in login database
      foundTest = False
      datLogin.DatabaseName = App.Path & "\login.mdb"
      datLogin.RecordSource = "Test"
      datLogin.Refresh
      With datLogin.Recordset
         .MoveFirst
         Do Until .EOF Or foundTest
            If Trim(currentTest) = Trim(.Fields("TestName").Value) Then
               foundTest = True
            End If
            .MoveNext
         Loop
         If Not foundTest Then
            .AddNew
            .Fields("TestName").Value = Trim(currentTest)
            .Update
         End If
      End With


   dlgError:

      On Error GoTo 0
      Exit Sub

   End Sub
```

6. User Account Form

```
Private Sub cmdAdd_Click()
   Me.Hide
   Load frmAccountAdd
End Sub

Private Sub cmdCancel_Click()
   Unload Me
End Sub
```

```
    If lisTestQuests.ListCount > 0 Then
      'select file for output
      Open currentTest For Output As #1
      'write array to file
      For recordNum = 1 To lisTestQuests.ListCount
        Write #1, questTest(recordNum).quest
        Write #1, questTest(recordNum).answerA
        Write #1, questTest(recordNum).answerB
        Write #1, questTest(recordNum).answerC
        Write #1, questTest(recordNum).answerD
        Write #1, questTest(recordNum).correctAns
        Write #1, questTest(recordNum).theType
      Next recordNum
      Close #1

      'open file for test layout
      Open Left(currentTest, Len(currentTest) - 3) & "lyt" For Output As #1
      'write the layout options
      If chkTimed Then
        Write #1, "T"
      Else
        Write #1, "F"
      End If
      If chkTimed And Val(txtMinutes) < 1 Then
        Write #1, 1
      Else
        Write #1, Val(txtMinutes)
      End If
      If chkAllowGoBack Then
        Write #1, "T"
      Else
        Write #1, "F"
      End If
      If optGrey.Value = True Then
        Write #1, "G"
      Else
        If optYellow.Value = True Then
          Write #1, "R"
        Else
          If optBlue.Value = True Then
            Write #1, "B"
          Else
            Write #1, "GN"
          End If
        End If
      End If
      Close #1

      'reset test has changed to false
      testHasChanged = False
    End If

End Sub
Private Sub createTest()

    Dim foundTest As Boolean
    Dim i As Integer
    Dim userResponse As Integer

    'On Error GoTo dlgError

    'if test had changed then ask to save current test before
    'creating a new one
    If testHasChanged Then
      userResponse = MsgBox("The current test has changed since you last saved it. " & _
                    "Do you want to save it before you create a new one?", _
                    vbYesNoCancel, "Warning!")
      If userResponse = vbYes Then
        Call SaveTest
      Else
```

```
Private Sub cmdFinish_Click()
   Unload Me
End Sub

Private Sub cmdModify_Click()
   UserToModify.userID = MSFlexGrid1.TextMatrix(MSFlexGrid1.Row, 0)
   Call SelectAllFromLoginDatabase(Me)
  'find userID in recordset
   datLogin.Recordset.MoveFirst
   userCode = datLogin.Recordset.Fields("UserID").Value
   Do Until datLogin.Recordset.EOF
      userCode = datLogin.Recordset.Fields("UserID").Value
      If userCode = UserToModify.userID Then
         With UserToModify
            .Firstname = datLogin.Recordset.Fields("Firstname").Value
            .Lastname = datLogin.Recordset.Fields("Lastname").Value
            .ID = datLogin.Recordset.Fields("SSN").Value
            .Password = datLogin.Recordset.Fields("Password").Value
            .Instructor = datLogin.Recordset.Fields("Instructor").Value
         End With
         Exit Do
      Else
         datLogin.Recordset.MoveNext
      End If
   Loop

   Me.Hide
   Load frmAccountChange
End Sub

Private Sub cmdRemove_Click()
  'determine userID
   UserToModify.userID = MSFlexGrid1.TextMatrix(MSFlexGrid1.Row, 0)
   'Call SelectAllFromLoginDatabase(Me)

   'find userID in recordset
   datLogin.Recordset.MoveFirst
   userCode = datLogin.Recordset.Fields("UserID").Value
   Do Until datLogin.Recordset.EOF
      userCode = datLogin.Recordset.Fields("UserID").Value
      If userCode = UserToModify.userID Then
         datLogin.Recordset.Delete
         datLogin.Refresh
         datLogin.UpdateRecord
         Exit Do
      Else
         datLogin.Recordset.MoveNext
      End If
   Loop
End Sub

Private Sub Form_Load()
   Call CenterThisFormOnScreen(Me)
   rbtStudent.Value = True
   'set column width
   MSFlexGrid1.ColWidth(0) = 1500
   For i = 1 To 2
      MSFlexGrid1.ColWidth(i) = 1500
   Next i

   Call InitializeLoginDatabase(Me)
End Sub

Private Sub rbtLecturer_Click()
      With datLogin
         .DatabaseName = App.Path & "\login.mdb"
         .RecordSource = "SELECT UserID, FirstName, LastName " & _
                         "FROM Login " & _
                         "WHERE Instructor = True"
```

```
        .Refresh

    End With
End Sub

Private Sub rbtStudent_Click()
    With datLogin
    .DatabaseName = App.Path & "\login.mdb"
    .RecordSource = "SELECT UserID, FirstName, LastName " & _
                    "FROM Login " & _
                    "WHERE Instructor = False"
    .Refresh
    'MSFlexGrid1.
    '.ListField = "UserID"
  End With
End Sub
```

7. Create User Account Form

```
Private Sub cmdCancel_Click()
  'refresh frmAccount grid
  updateLogin
  'call subroutine
  refreshLocation
  Unload Me
  frmAccount.Show

End Sub

Private Sub cmdCreate_Click()
  Dim Password As String
  Dim userID As String
  Dim passchar As Integer
  Dim i As Integer

  'check for any blank fields
  If txtFirst <> "" And txtLast <> "" And txtID <> "" Then
    'generate random password
    Randomize Timer
    For i = 1 To 5
      passchar = Int(26 * Rnd) + 65
      Password = Password + Chr$(passchar)
    Next i
    'obtain first 4 characters from last name and first name
    'if last name is shorter than 4 characters
    If Len(txtLast) < 4 Then
      userID = UCase(txtLast & Left(txtFirst, 4 - Len(txtLast)))
    Else
      userID = UCase(Left(txtLast, 4))
    End If
    'create user ID
    userID = userID & Right(txtID, 4)
    'display the user ID and password to user
    picOutput.Cls
    picOutput.Print "User I.D. is :"; userID
    picOutput.Print "Password  is :"; Password
    'add new account to login database

    'call subroutine
    updateLoginDatabase

    With frmAccount.datLogin.Recordset
      .AddNew
      .Fields("UserID").Value = userID
      .Fields("LastName").Value = txtLast
      .Fields("FirstName").Value = txtFirst
```

```
        End With
      Else
        With frmAccount.datLogin
        .DatabaseName = App.Path & "\login.mdb"
        .RecordSource = "SELECT UserID, FirstName, LastName " & _
                        "FROM Login " & _
                        "WHERE Instructor = True"
        .Refresh
        End With
      End If

    End Sub

    Private Sub refreshLocation()
      frmAccount.Top = Me.Top
      frmAccount.Left = Me.Left
    End Sub
```

8. Modify User Account Form

```
    Private Sub Form_Load()

    Call CenterThisFormOnScreen(Me)
    Me.Top = frmAccount.Top
    Me.Left = frmAccount.Left

    With UserToModify
      txtFirst.Text = .Firstname
      txtLast.Text = .Lastname
      txtID.Text = .ID
      txtUserID.Text = .userID
      txtPassword.Text = .Password
      If .Instructor Then
        rbtLecturer.Value = True
      Else
        rbtStudent.Value = True
      End If

    End With

  End Sub
  Private Sub cmdCancel_Click()

    'refresh frmAccount grid
    updateLogin

    'call subroutine
    refreshLocation

    Unload Me

    frmAccount.Show

  End Sub

  Private Sub cmdCreate_Click()


    Dim Password As String
    Dim userID As String
    Dim passchar As Integer
    Dim i As Integer

    'check for any blank fields
    If txtFirst <> "" And txtLast <> "" And txtID <> "" Then
      'generate random password
      Randomize Timer
```

```
    If lisTestQuests.ListCount > 0 Then
      'select file for output
      Open currentTest For Output As #1
      'write array to file
      For recordNum = 1 To lisTestQuests.ListCount
        Write #1, questTest(recordNum).quest
        Write #1, questTest(recordNum).answerA
        Write #1, questTest(recordNum).answerB
        Write #1, questTest(recordNum).answerC
        Write #1, questTest(recordNum).answerD
        Write #1, questTest(recordNum).correctAns
        Write #1, questTest(recordNum).theType
      Next recordNum
      Close #1

      'open file for test layout
      Open Left(currentTest, Len(currentTest) - 3) & "lyt" For Output As #1
      'write the layout options
      If chkTimed Then
        Write #1, "T"
      Else
        Write #1, "F"
      End If
      If chkTimed And Val(txtMinutes) < 1 Then
        Write #1, 1
      Else
        Write #1, Val(txtMinutes)
      End If
      If chkAllowGoBack Then
        Write #1, "T"
      Else
        Write #1, "F"
      End If
      If optGrey.Value = True Then
        Write #1, "G"
      Else
        If optYellow.Value = True Then
          Write #1, "R"
        Else
          If optBlue.Value = True Then
            Write #1, "B"
          Else
            Write #1, "GN"
          End If
        End If
      End If
      Close #1

      'reset test has changed to false
      testHasChanged = False
    End If

End Sub
Private Sub createTest()

    Dim foundTest As Boolean
    Dim i As Integer
    Dim userResponse As Integer

    'On Error GoTo dlgError

    'if test had changed then ask to save current test before
    'creating a new one
    If testHasChanged Then
      userResponse = MsgBox("The current test has changed since you last saved it. " & _
                    "Do you want to save it before you create a new one?", _
                    vbYesNoCancel, "Warning!")
      If userResponse = vbYes Then
        Call SaveTest
      Else
```

```
    .DatabaseName = App.Path & "\login.mdb"
    .RecordSource = "SELECT UserID, FirstName, LastName, SSN, Password, Instructor " & _
                    "FROM Login "
    .Refresh
  End With

End Sub

Private Sub updateLogin()
  If frmAccount.rbtStudent.Value = True Then

  With frmAccount.datLogin
    .DatabaseName = App.Path & "\login.mdb"
    .RecordSource = "SELECT UserID, FirstName, LastName " & _
                    "FROM Login " & _
                    "WHERE Instructor = False"
    .Refresh
  End With
  Else
    With frmAccount.datLogin
    .DatabaseName = App.Path & "\login.mdb"
    .RecordSource = "SELECT UserID, FirstName, LastName " & _
                    "FROM Login " & _
                    "WHERE Instructor = True"
    .Refresh
    End With
  End If
End Sub

Private Sub refreshLocation()
  frmAccount.Top = Me.Top
  frmAccount.Left = Me.Left
End Sub
```

9. Change User Password Form

```
  Private Sub Form_Load()
    Call CenterThisFormOnScreen(Me)
    With currentUser
      If .Instructor Then
        txtUserType.Caption = "Instructur"
      Else
        txtUserType.Caption = "Student"
      End If
      txtFirst.Text = .Firstname
      txtLast.Text = .Lastname
      txtID.Text = .ID
      txtID.Text = .userID
    End With
  End Sub

  Private Sub cmdCancel_Click()
    Unload Me
  End Sub

  Private Sub cmdFinish_Click()
    If Trim(txtPassword.Text) <> Trim(currentUser.Password) Then
      MsgBox ("Invalid Current Password")
      Exit Sub
    ElseIf Trim(txtNewPassword.Text) <> Trim(txtConfirmPassword.Text) Then
        MsgBox ("Incorrect New Password Entry")
        Exit Sub
    End If
    Call UpdateUserPassword
    Unload Me
    frmAccount.Show
  End Sub
```

```
Private Sub UpdateUserPassword()
  'Call CenterThisFormOnScreen(Me)
  With datLogin
    .DatabaseName = App.Path & "\login.mdb"
    .RecordSource = "Login"
    .Refresh
  End With

  'search userID and change password
  With datLogin
    .Recordset.MoveFirst
    userCode = .Recordset.Fields("UserID").Value
    Do Until found Or .Recordset.EOF
      userCode = .Recordset.Fields("UserID").Value
      If UCase(Trim(userCode)) = UCase(Trim(currentUser.userID)) Then
        .Recordset.Edit
        .Recordset.Fields("Password").Value = Trim(txtNewPassword.Text)
        .Refresh
        .Recordset.Update
        Exit Do
      Else
        datLogin.Recordset.MoveNext
      End If
    Loop
  End With
End Sub
```

10. About MyApplication Form

```
Private Sub cmdOK_Click()
  Unload Me
End Sub

Private Sub Form_Load()
  Call CenterThisFormOnScreen(Me)
  Me.Caption = "About " & App.Title
End Sub
```

11. Choose Test Form

```
      Private Sub cmdCancel_Click()
  End
End Sub

Private Sub cmdOpen_Click()
With dlgFile
    .CancelError = True
    .Flags = 2
    .Filter = "Test Files| *.tst"
    .DialogTitle = "Choose Test To Take"
    .ShowOpen
    If .FileName <> "" Then lblTestFileName.Caption = .FileTitle
End With
End Sub

Private Sub cmdStart_Click()
  Me.Hide
  frmStudent.Show
End Sub

Private Sub Form_Load()
```

```
    CenterThisFormOnScreen Me
End Sub
```

12. Student Test Form

```
Dim currentQuest As Integer
Dim questNum As Integer
Dim goBack As Boolean
Dim timed As Boolean
Dim minutes As Integer

Private Sub cmdFinished_Click()

    'check for no answer given
    If opt1.Value = False And opt2.Value = False And opt3.Value = False _
      And opt4.Value = False Then
      MsgBox "You must select an answer!", , "Try Again!"
      Exit Sub
    End If

    'assign answer selected to users answer array for checking
    If opt1.Value = True Then
       usersAnswer(currentQuest) = "A"
    Else
       If opt2.Value = True Then
          usersAnswer(currentQuest) = "B"
       Else
          If opt3.Value = True Then
             usersAnswer(currentQuest) = "C"
          Else
             usersAnswer(currentQuest) = "D"
          End If
       End If
    End If

    testIsOver

End Sub

Private Sub cmdNextQuestion_Click()

    'check for no answer
    If opt1.Value = False And opt2.Value = False And opt3.Value = False _
      And opt4.Value = False Then
      MsgBox "You must select an answer!", , "Try Again!"
      Exit Sub
    End If

    'enable previous command button if go back is allowed
    If goBack Then
       cmdPrevious.Enabled = True
    End If

    'assigns answer to users answer array
    If opt1.Value = True Then
       usersAnswer(currentQuest) = "A"
    Else
       If opt2.Value = True Then
          usersAnswer(currentQuest) = "B"
       Else
          If opt3.Value = True Then
             usersAnswer(currentQuest) = "C"
          Else
             usersAnswer(currentQuest) = "D"
          End If
       End If
    End If
```

```
    currentQuest = currentQuest + 1

    'if current question is last question then disable next button
    If currentQuest = questNum Then
       cmdNextQuestion.Enabled = False
       cmdFinished.Enabled = True
    End If

    'load value (if any) from users answer array for next question
    If usersAnswer(currentQuest) = "A" Then
       opt1.Value = True
    Else
       If usersAnswer(currentQuest) = "B" Then
          opt2.Value = True
       Else
          If usersAnswer(currentQuest) = "C" Then
             opt3.Value = True
          Else
             If usersAnswer(currentQuest) = "D" Then
                opt4.Value = True
             Else
                opt1.Value = False
                opt2.Value = False
                opt3.Value = False
                opt4.Value = False
             End If
          End If
       End If
    End If

    'load question onto screen for user viewing
    loadQuestion

End Sub

Private Sub cmdPrevious_Click()


    'make sure question is not number one when moving back
    If currentQuest > 1 Then

       'assign current answer to users answer array
       If opt1.Value = True Then
          usersAnswer(currentQuest) = "A"
       Else
          If opt2.Value = True Then
             usersAnswer(currentQuest) = "B"
          Else
             If opt3.Value = True Then
                usersAnswer(currentQuest) = "C"
             Else
                If opt4.Value = True Then
                   usersAnswer(currentQuest) = "D"
                End If
             End If
          End If
       End If

       'enable next button
       cmdNextQuestion.Enabled = True
       currentQuest = currentQuest - 1

       'if current question is number 1 disable previous button
       If currentQuest = 1 Then
          cmdPrevious.Enabled = False
       End If
       'load current answer(if any) from users answer array for the
       'question the user is moving back to
       If usersAnswer(currentQuest) = "A" Then
```

```
        opt1.Value = True
      Else
        If usersAnswer(currentQuest) = "B" Then
          opt2.Value = True
        Else
          If usersAnswer(currentQuest) = "C" Then
            opt3.Value = True
          Else
            If usersAnswer(currentQuest) = "D" Then
              opt4.Value = True
            Else
              opt1.Value = False
              opt2.Value = False
              opt3.Value = False
              opt4.Value = False
            End If
          End If
        End If
      End If
      loadQuestion
    End If

End Sub

Private Sub Form_Load()

    Call CenterThisFormOnScreen(Me)
    Show
    Unload frmLogin
    Caption = "Online Test JAVA System - Student Screen - User - " & loggedUser
    opt1.Value = False
    cmdPrevious.Enabled = False
    cmdNextQuestion.Enabled = False
    cmdFinished.Enabled = False

    datLogin.DatabaseName = App.Path & "\login.mdb"
    datLogin.RecordSource = "Login"
    datLogin.Refresh

    datScores.DatabaseName = App.Path & "\login.mdb"
    datScores.RecordSource = "TestScores"
    datScores.Refresh

    Call initialize_test


    Dim response As Integer

    'check for timed test, and give user option to start or not
    If timed Then
      response = MsgBox("You have " & minutes & " minutes to finish this test.  " & _
          "Once you have started you can't stop the test.  " & _
          "Are you ready to start?", vbYesNo, "Timed Test")
    Else
      response = MsgBox("You have an unlimited amount of time to finish this " & _
          "test.  Once you have started you can't stop the " & _
          "test.  Are you ready to start?", vbYesNo, _
          "Non-Timed Test")
    End If

    If response = vbNo Then
      Exit Sub
    Else
      If timed Then
        Timer1.Enabled = True
      End If
      cmdNextQuestion.Enabled = True
      cmdFinished.Enabled = False
      currentQuest = 1
      'mnuOpen.Enabled = False
```

```
    'mnuStart.Enabled = False
    'mnuExit.Enabled = False
    'mnuChangePassword.Enabled = False
    'mnuDisplayScores.Enabled = False
    'mnuPrintScores.Enabled = False

    loadQuestion
  End If


End Sub

Private Sub initialize_test()

  Dim i As Integer
  Dim isTimed As String
  Dim allowBack As String
  Dim theColor As String
  Dim ColorSet As Long

  On Error GoTo dlgError

  With frmChooseTest.dlgFile

    'clear test question array
    For i = 1 To 100
      usersAnswer(i) = ""
      questTest(i).answerA = ""
      questTest(i).answerB = ""
      questTest(i).answerC = ""
      questTest(i).answerD = ""
      questTest(i).correctAns = ""
      questTest(i).quest = ""
      questTest(i).theType = ""
    Next i

    'set current test for the program to reference
    currentTest = .FileName
    'set the label to indicate current test
    'lblTestName.Caption = currentTest

    questNum = 0
    'load test question array from test file
    Open currentTest For Input As #1
    Do Until EOF(1)
      questNum = questNum + 1
      Input #1, questTest(questNum).quest
      Input #1, questTest(questNum).answerA
      Input #1, questTest(questNum).answerB
      Input #1, questTest(questNum).answerC
      Input #1, questTest(questNum).answerD
      Input #1, questTest(questNum).correctAns
      Input #1, questTest(questNum).theType
    Loop
    Close #1

    'load layout for test
    Open Left(currentTest, Len(currentTest) - 3) & "lyt" For Input As #1
      Input #1, isTimed
      Input #1, minutes
      Input #1, allowBack
      Input #1, theColor
    Close #1

      If isTimed = "T" Then
        timed = True
      Else
        timed = False
      End If
```

```
        If timed Then
          lblMinute.Caption = minutes
        Else
          lblMinute.Caption = 0
        End If
        lblSeconds.Caption = "00"
        Timer1.Enabled = False

        If allowBack = "T" Then
          goBack = True
        Else
          goBack = False
        End If

        cmdPrevious.Enabled = False

        If theColor = "R" Then
          ColorSet = vbRed
        Else
          If theColor = "GN" Then
            ColorSet = vbGreen
          Else
            If theColor = "B" Then
              ColorSet = vbBlue
            Else
              ColorSet = -2147483633
            End If
          End If
        End If

        frmStudent.BackColor = ColorSet
        Label2.BackColor = ColorSet
        lblQuestionNumber.BackColor = ColorSet
        lblTestName.BackColor = ColorSet
        lblQuestionType.BackColor = ColorSet
        frmAnswers.BackColor = ColorSet
        opt1.BackColor = ColorSet
        opt2.BackColor = ColorSet
        opt3.BackColor = ColorSet
        opt4.BackColor = ColorSet
        cmdPrevious.BackColor = ColorSet
        cmdNextQuestion.BackColor = ColorSet
        cmdFinished.BackColor = ColorSet

      mnuStart.Enabled = True
    End With

dlgError:
    On Error GoTo 0
    Exit Sub

End Sub

Private Sub mnuPrintScores_Click()

    'set printer properties
    Printer.ScaleMode = 4
    Printer.FontSize = 12
    Printer.CurrentY = 5

    'print all test scores for currently logged user
    With datScores.Recordset
      .MoveFirst
      Do Until .EOF
        If .Fields("ID").Value = loggedUser Then
          Printer.CurrentX = 5
          Printer.Print .Fields("Test").Value;
          Printer.CurrentX = 25
          Printer.Print .Fields("Date").Value;
          Printer.CurrentX = 45
```

```
            Printer.Print .Fields("Grade").Value
          End If
          .MoveNext
        Loop
    End With

    Printer.EndDoc
End Sub
Private Sub mnuChangePassword_Click()

    'edit currently logged on user's password
    Load frmTeachPass
    frmTeachPass.Caption = "Change Student Password"
    'search for user in database and display current(old) password
    With datLogin.Recordset
      .MoveFirst
      Do Until .EOF
        If loggedUser = .Fields("UserID").Value Then
          frmTeachPass.picOutput.Cls
          frmTeachPass.picOutput.Print "Old Password:  "; .Fields("Password").Value
          Exit Do
        End If
        .MoveNext
      Loop
    End With

End Sub

Private Sub mnuDisplayScores_Click()

    Load frmDisplayTestStudent

End Sub

Private Sub mnuExit_Click()

    End

End Sub

Private Sub mnuOpen_Click()

    Dim i As Integer
    Dim isTimed As String
    Dim allowBack As String
    Dim theColor As String
    Dim ColorSet As Long

    On Error GoTo dlgError

    With dlgFile
      .CancelError = True
      .Flags = 2
      .Filter = "Test Files| *.tst"
      .DialogTitle = "Choose Test To Take"
      .ShowOpen

      'clear test question array
      For i = 1 To 100
        usersAnswer(i) = ""
        questTest(i).answerA = ""
        questTest(i).answerB = ""
        questTest(i).answerC = ""
        questTest(i).answerD = ""
        questTest(i).correctAns = ""
        questTest(i).quest = ""
        questTest(i).theType = ""
      Next i

      'set current test for the program to reference
```

```
currentTest = .FileName
'set the label to indicate current test
lblTestName.Caption = currentTest

questNum = 0
'load test question array from test file
Open currentTest For Input As #1
Do Until EOF(1)
   questNum = questNum + 1
   Input #1, questTest(questNum).quest
   Input #1, questTest(questNum).answerA
   Input #1, questTest(questNum).answerB
   Input #1, questTest(questNum).answerC
   Input #1, questTest(questNum).answerD
   Input #1, questTest(questNum).correctAns
   Input #1, questTest(questNum).theType
Loop
Close #1

'load layout for test
Open Left(currentTest, Len(currentTest) - 3) & "lyt" For Input As #1
   Input #1, isTimed
   Input #1, minutes
   Input #1, allowBack
   Input #1, theColor
Close #1

   If isTimed = "T" Then
      timed = True
   Else
      timed = False
   End If

   If timed Then
      lblMinute.Caption = minutes
   Else
      lblMinute.Caption = 0
   End If
   lblSeconds.Caption = "00"
   Timer1.Enabled = False

   If allowBack = "T" Then
      goBack = True
   Else
      goBack = False
   End If

   cmdPrevious.Enabled = False

   If theColor = "R" Then
      ColorSet = vbRed
   Else
      If theColor = "GN" Then
         ColorSet = vbGreen
      Else
         If theColor = "B" Then
            ColorSet = vbBlue
         Else
            ColorSet = -2147483633
         End If
      End If
   End If

   frmStudent.BackColor = ColorSet
   Label2.BackColor = ColorSet
   lblQuestionNumber.BackColor = ColorSet
   lblTestName.BackColor = ColorSet
   lblQuestionType.BackColor = ColorSet
   frmAnswers.BackColor = ColorSet
   opt1.BackColor = ColorSet
```

```
        opt2.BackColor = ColorSet
        opt3.BackColor = ColorSet
        opt4.BackColor = ColorSet
        cmdPrevious.BackColor = ColorSet
        cmdNextQuestion.BackColor = ColorSet
        cmdFinished.BackColor = ColorSet

      mnuStart.Enabled = True
    End With

dlgError:
    On Error GoTo 0
    Exit Sub

End Sub



Private Sub mnuStart_Click()

    Dim response As Integer

    'check for timed test, and give user option to start or not
    If timed Then
      response = MsgBox("You have " & minutes & " minutes to finish this test.  " & _
          "Once you have started you can't stop the test.  " & _
          "Are you ready to start?", vbYesNo, "Timed Test")
    Else
      response = MsgBox("You have an unlimited amount of time to finish this " & _
          "test.  Once you have started you can't stop the " & _
          "test.  Are you ready to start?", vbYesNo, _
          "Non-Timed Test")
    End If

    If response = vbNo Then
      Exit Sub
    Else
      If timed Then
        Timer1.Enabled = True
      End If
      cmdNextQuestion.Enabled = True
      cmdFinished.Enabled = False
      currentQuest = 1
      mnuOpen.Enabled = False
      mnuStart.Enabled = False
      mnuExit.Enabled = False
      mnuChangePassword.Enabled = False
      mnuDisplayScores.Enabled = False
      mnuPrintScores.Enabled = False

      loadQuestion
    End If

End Sub

Private Sub loadQuestion()

    'set the label to display current question number
    lblQuestionNumber.Caption = currentQuest

    'load answers and question on to user screen
    opt1.Caption = questTest(currentQuest).answerA
    opt2.Caption = questTest(currentQuest).answerB
    opt3.Caption = questTest(currentQuest).answerC
    opt4.Caption = questTest(currentQuest).answerD
    txtQuestion.Text = questTest(currentQuest).quest

    'if there is only 1 question in test disable all buttons except
    'for finished.
    If questNum = 1 Then
```

```
    cmdNextQuestion.Enabled = False
    cmdFinished.Enabled = True
    cmdPrevious.Enabled = False
  End If

  'set the label to indicate the type of question
  If questTest(currentQuest).theType = "T" Then
    opt3.Enabled = False
    opt4.Enabled = False
    lblQuestionType.Caption = "True/False"
  Else
    opt3.Enabled = True
    opt4.Enabled = True
    lblQuestionType.Caption = "Multiple Choice"
  End If

End Sub



Private Sub Timer1_Timer()

  'stop test if no time is left
  If Val(lblMinute.Caption) = 0 And Val(lblSeconds.Caption) = 0 Then
    testIsOver
    Exit Sub
  End If

  'decrement timer by 1 second every 1 second
  If Val(lblSeconds.Caption) = 0 Then
    lblSeconds.Caption = 59
    lblMinute.Caption = Val(lblMinute.Caption) - 1
  Else
    lblSeconds.Caption = Val(lblSeconds.Caption) - 1
    If Val(lblSeconds.Caption) < 10 Then
      lblSeconds.Caption = "0" & lblSeconds.Caption
    End If
  End If

End Sub

Private Sub testIsOver()

  Dim i As Integer
  Dim score As Integer
  Dim numOfQuestions As Integer
  Dim numberCorrect As Integer

  'shut off timer, and display test over
  Timer1.Enabled = False
  MsgBox "Time is up!!", , "Test Over"
  'enable menu options
  'mnuOpen.Enabled = True
  'mnuExit.Enabled = True
  'mnuChangePassword.Enabled = True
  'mnuDisplayScores.Enabled = True
  'mnuPrintScores.Enabled = True
  'disable command buttons
  cmdPrevious.Enabled = False
  cmdNextQuestion.Enabled = False
  cmdFinished.Enabled = False

  'check answers
  numberCorrect = 0
  numOfQuestions = questNum
  For i = 1 To numOfQuestions
    If usersAnswer(i) = RTrim(questTest(i).correctAns) Then
      numberCorrect = numberCorrect + 1
    End If
  Next i
```

```
    score = numberCorrect / questNum * 100
    userScore = score
    numOfQ = questNum
    numCorrect = numberCorrect

    'add test, id, date, and grade to test scores database
    With datScores.Recordset
       .AddNew
       .Fields("Test").Value = Mid(frmChooseTest.dlgFile.FileTitle, 1, 50)
       .Fields("ID").Value = currentUser.userID
       .Fields("Date").Value = Date
       .Fields("Grade").Value = score
       .Update
    End With

    Load frmDisplayGrade

End Sub
```

13. Display Grade Form

```
Private Sub cmdDisplay_Click()
    Dim i As Integer
    Dim x As Integer
    Dim numOfQuest As Integer

    'clear pic box and display score
    picScore.Cls
    picScore.Print "Out of " & numOfQ & " questions"
    picScore.Print "you answered " & numCorrect & " correctly."
    picScore.Print "Your score is " & userScore & "."


    numOfQuest = numOfQ
    x = 0
    For i = 1 To numOfQuest
       x = x + 1
       'display question numbers and X's if answer is incorrect
       If usersAnswer(i) = Trim(questTest(i).correctAns) Then
          picAnswers.Print "Question" & i & " - " & usersAnswer(i),
       Else
          picAnswers.Print "Question" & i & " - " & usersAnswer(i) & "  X",
       End If
       'allow 4 answers to be displayed per line
       If x = 4 Then
          picAnswers.Print
          x = 0
       End If
    Next i
End Sub

Private Sub cmdFinished_Click()
    Unload frmStudent
    Unload frmChooseTest
    Unload Me
    Load frmChooseTest
End Sub

Private Sub cmdPrintScore_Click()
    Dim i As Integer
    Dim x As Integer
    Dim numOfQuest As Integer

    'print score on printer
    Printer.FontSize = 14
    Printer.Print "Out of " & numOfQ & " questions"
    Printer.Print "you answered " & numCorrect & " correctly."
    Printer.Print "Your score is " & userScore & "."
```

```
    Printer.Print: Printer.Print

    numOfQuest = numOfQ
    x = 0
    For i = 1 To numOfQuest
      x = x + 1
      'print question number and X's on wrong answer to printer
      If usersAnswer(i) = RTrim(questTest(i).correctAns) Then
        Printer.Print i & " - " & usersAnswer(i),
      Else
        Printer.Print "X" & i & " - " & usersAnswer(i) & "X",
      End If
      'allow 4 answers on each line
      If x = 4 Then
        Printer.Print
        x = 0
      End If
    Next i

    Printer.EndDoc
End Sub

Private Sub Form_Load()
    Show
    Call CenterThisFormOnScreen(Me)
End Sub
```

14. Test Scores Form

```
        Private Sub cmbStudent_Click(Area As Integer)

    'when user selects an user ID, display the user's name as
    'a tool tip
    With datLogin.Recordset
      .MoveFirst
      .FindFirst ("UserID = '" & cmbStudent & "'")
      cmbStudent.ToolTipText = RTrim(.Fields("FirstName").Value) & " " & _
              .Fields("LastName").Value
    End With

End Sub

Private Sub cmdCancel_Click()
    Unload Me
    FlushGlobalVariables
    ToggleMenu
End Sub

Private Sub cmdDisplayStudent_Click()

    Dim numOfTests As Integer
    Dim totalScore As Integer
    Dim average As Single

    'move to correct records and display info
    picDisplay.Cls
    With datScores.Recordset
      .MoveFirst
      Do Until .EOF
        If cmbStudent.Text = .Fields("ID").Value Then
          numOfTests = numOfTests + 1
          totalScore = totalScore + .Fields("Grade").Value
          picDisplay.Print "Test: " & .Fields("Test").Value; _
                  Tab(25); "I.D. "; .Fields("ID").Value; _
                  Tab(55); .Fields("Date").Value; _
                  Tab(70); .Fields("Grade").Value
```

```
        End If
        .MoveNext
      Loop

      'calculate average if any tests existed
      If numOfTests > 0 Then
        average = totalScore / numOfTests
        picDisplay.Print
        picDisplay.Print "Average of tests for student " & cmbStudent.Text & _
                  " is " & FormatNumber(average, 1)
      Else
        MsgBox "Test Scores Not Available", , "Attention"
      End If
    End With

End Sub

Private Sub cmdDisplayTest_Click()


    Dim numOfTests As Integer
    Dim totalScore As Integer
    Dim average As Single

    'move to current records and display
    picDisplay.Cls
    With datScores.Recordset
      .MoveFirst
      Do Until .EOF
        If cmbTest.Text = .Fields("Test").Value Then
          numOfTests = numOfTests + 1
          totalScore = totalScore + .Fields("Grade").Value
          picDisplay.Print "Test: " & cmbTest.Text; _
                    Tab(25); "I.D. "; .Fields("ID").Value; _
                    Tab(55); .Fields("Date").Value; _
                    Tab(70); .Fields("Grade").Value
        End If
        .MoveNext
      Loop

      'if tests existed then display average
      If numOfTests > 0 Then
        average = totalScore / numOfTests
        picDisplay.Print
        picDisplay.Print "Average of test " & cmbTest.Text & _
                  " is " & FormatNumber(average, 1)
      Else
        MsgBox "Test Scores Not Available", , "Attention"
      End If
    End With

End Sub

Private Sub cmdPrintStudent_Click()

    Dim numOfTests As Integer
    Dim totalScore As Integer
    Dim average As Single

    Printer.FontSize = 12
    'find records and print
    With datScores.Recordset
      .MoveFirst
      Do Until .EOF
        If cmbStudent.Text = .Fields("ID").Value Then
          numOfTests = numOfTests + 1
          totalScore = totalScore + .Fields("Grade").Value
          Printer.Print "Test: " & .Fields("Test").Value; _
                    Tab(35); "I.D. "; .Fields("ID").Value; _
                    Tab(55); .Fields("Date").Value; _
```

```
                    Tab(70); .Fields("Grade").Value
        End If
        .MoveNext
      Loop

      'if tests existed then print the average
      If numOfTests > 0 Then
        average = totalScore / numOfTests
        Printer.Print
        Printer.Print "Average of tests for student " & cmbStudent.Text & _
                " is " & FormatNumber(average, 1)
        Printer.EndDoc
      Else
        MsgBox "Test Scores Not Available", , "Attention"
      End If
    End With

End Sub

Private Sub cmdPrintTest_Click()

    Dim numOfTests As Integer
    Dim totalScore As Integer
    Dim average As Single

    Printer.FontSize = 12
    'find correct records and print
    With datScores.Recordset
      .MoveFirst
      Do Until .EOF
        If cmbTest.Text = .Fields("Test").Value Then
          numOfTests = numOfTests + 1
          totalScore = totalScore + .Fields("Grade").Value
          Printer.Print "Test: " & cmbTest.Text, _
                    "I.D. " & .Fields("ID").Value, _
                      .Fields("Date").Value, _
                      .Fields("Grade").Value
        End If
        .MoveNext
      Loop

      'if tests existed then print the average
      If numOfTests > 0 Then
        average = totalScore / numOfTests
        Printer.Print
        Printer.Print "Average of test " & cmbTest.Text & _
              " is " & FormatNumber(average, 1)
        Printer.EndDoc
      Else
        MsgBox "Test Scores Not Available", , "Attention"
      End If
    End With
End Sub

Private Sub Form_Load()

    Call CenterThisFormOnScreen(Me)
    Show
    ToggleMenu 'disable main menu
    datLogin.DatabaseName = App.Path & "\login.mdb"
    datLogin.RecordSource = "SELECT * " & _
                    "FROM Login " & _
                    "WHERE Instructor = False"
    datLogin.Refresh

    datScores.DatabaseName = App.Path & "\login.mdb"
    datScores.RecordSource = "TestScores"
    datScores.Refresh

    datTest.DatabaseName = App.Path & "\login.mdb"
```

```
    datTest.RecordSource = "Test"
    datTest.Refresh

End Sub
```