E-LEARNING

(A CASE STUDY OF INVERSE MATRIX IN MATHEMATICS)

BY

UDOH, Paul Philip

PGD/MCS/2008/1248

Submitted to Department of Mathematics/Computer Science

Federal University of Technology

Minna

In Partial Fulfilment of Requirements Leading to the Award of Postgraduate Diploma (PGD) in Computer Science

Federal University of Technology, Minna.

JULY, 2010

CERTIFICATION

This project titled, E-Learning (A Case Study of Inverse Matrix in Mathematics) by Udoh, Paul Philip with matriculation No PGD/MCS/2008/1248 meets the regulations governing the award of Postgraduate Diploma in Computer Science of Federal University of Technology Minna.

------------------------------------ ------------------------------------

Mr I. Onotu Date

Supervisor

------------------------------------ ------------------------------------

Prof. N. I. Akinwande Date

Head of Department

DECLARATION

I hereby declare that this project is my original work and is a record of my research effort. It has not been presented wholly or partially in any previous application for the award of degree, diploma or certificate of any other university or institution of higher learning. All information from published and unpublished works of others has been duly acknowledged by means of reference.

....

UDOH, PAUL PHILIP DATE

DEDICATION

This work is dedicated to GOD Almighty and to my mentor Saint Jean Baptiste Marie Vianney, my family, and friends and to all people who have made this Project work possible.

ACKNOWLEDGEMENT

To GOD Almighty for the privileges He has given me to do this work, without His grace I would not have been able to do it all by my and ability alone. My profound gratitude goes to my sincere project supervisor Mr Onotu who gave me the necessary assistance despite the short time available that facilitated and made this project work a success. I also appreciate the Head of Department of Mathematics/Computer science Dr. N. I. Akiwande, my PGD co-ordinator Mallam Ndanusa, my lecturer Dr. Jiya and other lecturers.

I'll forever remain grateful to my mum Regina Udoh, brother Fidelis Udoh, my inlaw Eteka Ukoh, my sisters Victoria, Grace, Eno, Stella, Gertrude , my Nephews Philip, William, Ekeoma and nice Rachael for their spiritual, financial and moral support.

I greatly thank Mr. Lawrence Ogugua, Professor Daniya, Mrs Maimunate James, Kunle Ibitiye, Idris, Abiola, Patience, Azeez Adesina for their moral support. I'm also grateful to my friends/school mates Hakeem Lawal, Samsideen Oluokun, Chibueze Okoro, Yusuf Billal, Alfred Onah t mention but a few. Thank you all for the support and encouragement. You will continue to be part of my life.

TABLE OF CONTENTS

Contents Page

ABSTRACT

The is a need to improve the traditional methodology of teaching mathematics which instructs students on how different mathematics equations works but often fail to explain why the work or even more importantly what use they have in the real world. Contributing to this decline is the difficulty schools face in attracting and retaining high quality mathematics teachers. The outcome of these is always students' poor performance in the subject This project work is an effort to curtail these problems. This is done by developing a computer aided instruction program that will allow instructors to share their itemized knowledge across borders, allowing students to attend courses across physical fields, have the opportunity of making information available internationally to anyone who is interested. The program is designed to make the learning of mathematics (Inverse Matrix) interesting and engaging. It will also increase the amount of quality instruction each student receives.

CHAPTER ONE

INTRODUCTION

1.1 Background of Study.

Mathematics defined as the science of quality and space represented in numbers and figures have long been dreaded by many that has direct or indirect contact with it.

Anyone who has struggled with mathematics in school will appreciate how difficult learning complex mathematical formulas can be. Books, exercises and traditional teaching methods instruct students on how different mathematics equations works but often fail to explain why the work or even more importantly what use they have in the real world. This gap between what is taught in class and what applies in the real world has widened with the advent of new technology, the internet and computer games have made traditional teaching method seem antiquated and out of touch.

"Students are increasingly living in two worlds: The world of class room and the real world, and the two are growing farther apart", cautions Chronis Kynigos, a researcher at the Research Academic Computer Technology Institute (RACTI). The problem has been a point of focus by educational communities in the world. Though efforts to use computer games and digital media in mathematics teaching have often been inconsistent with varying results based on schools curricula and countries. "The situation is very messy" Kynigos notes.()

To bring order into this messy and chaos situation, a teaching aid/tool will be developed to deal with a specific area of mathematics: Finding Inverse Matrix. The tool will use traditional mathematics representations. The teaching aid is designed to make students think and help them learn mathematics work in a way that is fun and engaging.

CHAPTER ONE

INTRODUCTION

1.1 Background of Study.

Mathematics defined as the science of quality and space represented in numbers and figures have long been dreaded by many that has direct or indirect contact with it.

Anyone who has struggled with mathematics in school will appreciate how difficult learning complex mathematical formulas can be. Books, exercises and traditional teaching methods instruct students on how different mathematics equations works but often fail to explain why the work or even more importantly what use they have in the real world. This gap between what is taught in class and what applies in the real world has widened with the advent of new technology, the internet and computer games have made traditional teaching method seem antiquated and out of touch.

"Students are increasingly living in two worlds: The world of class room and the real world, and the two are growing farther apart", cautions Chronis Kynigos, a researcher at the Research Academic Computer Technology Institute (RACTI). The problem has been a point of focus by educational communities in the world. Though efforts to use computer games and digital media in mathematics teaching have often been inconsistent with varying results based on schools curricula and countries. "The situation is very messy" Kynigos notes.()

To bring order into this messy and chaos situation, a teaching aid/tool will be developed to deal with a specific area of mathematics: Finding Inverse Matrix. The tool will use traditional mathematics representations. The teaching aid is designed to make students think and help them learn mathematics work in a way that is fun and engaging.

Computer-Based Training (CBTs) are self-paced learning activities accessible via a computer or hand held device. The contents are presented in a linear form, much like reading an online book or manual. For this reason they are often use in teaching mathematics. Computer-Based trainings provides learning stimulus beyond traditional learning methodology from textbook, manual, or classroom-based instruction. Example, the computer based training offers user friendly solutions for satisfying continuing education requirements. Instead of limiting students to attending courses or reading printing manuals, students are able to acquire knowledge and skills through methods that are much more conducive to individual learning preferences.

Computer-Based Training s can be a good alternative to printed learning materials since media like videos, can be easily embedded to enhance the learning. The Computer-Based Trainings can easily be distributed to a variety of audience at low cost once the initial development is completed.

1.2 Reasons Why Computer Should Be Used as a Teaching Aid/Tool:

The reasons comprises the following among others:

1. Reduce overall training time.

2. Spread training over extended periods of time (even months).

3. Bookmark progress (computer remembering where the student stopped).

4. Pay less per Credit hour.

5. Participate in class activities when convenient (not tied to class meetings).

6. Access course contents from a variety of locations.

7. Remain in one location with no need to travel.

Impact of Computer on Mathematics

Computer-Based Training (CBT) is defined as all forms of electronic supported learning and teaching, which are procedural in character and aim to effect the construction of knowledge with reference to individual experience, practice and knowledge. It also refers to the use of electronic applications and processes to learn. The applications and processes comprise web-based learning, computer-based learning, virtual classrooms and digital collaboration. A content of the lessons is delivered through CD-ROM, audio or video tape, internet, intranet, etc.

The impact of Computer based training on mathematics include the following

1. Increased Access: Instructors can share their itemize knowledge across borders, allowing students to attend courses across physical political fields. Have the opportunity of making information available internationally to anyone interested at minimum cost. For example the MIT OpenCourseWare Program has made substantial portions of that University curriculum and lectures available for free online. (E-learning Wikipedia).

2. To develop the skills and competence needed in the 21st century. This helps students to be computer literate in their discipline, profession or career (Bates 2009) states that a major argument for e-learning is that it enables learners to develop essential skills for knowledge based workers by embedding the use of information and communication technologies with the curriculum. He also argues that using computer based training in this way has major implications for course design and the assessment of learners.

3. Flexibility to learners: computer based training is self paced allowing the students to work at their convenience. The lessons available at all times (24×7) students do not need to physically attend classes. The lessons can be paused and resume at a later time.

4. Improved Performance: A 12 year Meta analysis of research by U.S department of Education found that higher education student in online learning generally performed better that those in face-to-face courses (e-learning).

1.3 Matrix

In mathematics, a matrix (plural matrices) is a rectangular table of numbers. There are rules for adding, subtracting and multiplying matrices together but the rules are different for numbers. For example A.B does not always give the same result as B.A which is the case for the multiplication of ordinary numbers. A matrix can also be defined as a set of numbers arrange in rows and columns to form a rectangular array. Many natural sciences use matrices quite a lot. Many Universities courses on matrices are taught very early. e.g. In the first year of studies.

An item in matrix is called an entry element. The example has entries 1, 9,13,20,55, and 4 Entries are denoted by a variable with the subscript i.e.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Matrices of the same size can be added and subtracted entry wise and matrices of compatible sizes can be multiplied. These operations have many properties of ordinary arithmetic except that matrix multiplication is not commutative, that AB and BA are not equal.

1.3.1 Definition of matrices:

The horizontal lines in a matrix are called rows and the vertical lines are called columns. A matrix with m rows and n columns is called an m-by-n matrix or (m×n matrix) and m and n are called it dimensions. The places in the matrix where the number are called entries. The entries of a matrix A that lies in the row number I and column number j is called the entry of A. this is written as A [ij] or aij

Matrix Notation

n Columns

aij

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \ldots \\ a_{21} & a_{22} & a_{23} \ldots \\ a_{31} & a_{32} & a_{33} \ldots \end{vmatrix}$$

m rows

Entries in a matrix are often referenced by using Pairs of subscripts.

1.3.2 Types of Matrices:

A Square Matrix: A square matrix has the same number of rows as columns, so m=n=. example

$$\begin{vmatrix} 1 & 2 & 5 \\ 2 & 8 & 9 \\ 5 & 9 & 4 \end{vmatrix} \quad a = a$$

This matrix has 3 rows and 3 columns. M = n = 3.

Identity Matrix: Every square set of matrix has a special counterpart called an identity matrix. The identity matrix has nothing but zeroes on the main diagonal, where there are all ones.

Example $\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$

Is an identity matrix. There is exactly one identity matrix for each square matrix dimension set. An identity matrix is special because when multiplying any matrix by the identity matrix, the result is always the original matrix with no change.

Inverse Matrix: An inverse matrix is a matrix that when multiplied by another matrix, equals the identity matrix. For example

$$\begin{vmatrix} 7 & 8 \\ 6 & 7 \end{vmatrix} \begin{vmatrix} 7 & -8 \\ -6 & 7 \end{vmatrix} \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 7 & -8 \\ -6 & 7 \end{vmatrix} \text{ is the inverse of } \begin{vmatrix} 7 & 8 \\ 6 & 7 \end{vmatrix}$$

One Column Matrix: A matrix that has many rows but only one column is called a column vector.

e.g $\begin{vmatrix} 1 \\ 2 \\ 3 \end{vmatrix}$

1.3.3 Determinants.

The determinants takes a square matrix and return a number. To understand what the number means, take each column of the matrix and draw it as a vector. The parallelogram drawn by those vectors has an area which is the determinant. For all 2x2 matrices, the formula is very simple.

For 3x3 matrices, the formula is more complicated

$$\begin{vmatrix} Aa_1 & b_1 & c_1 \\ Aa_2 & b_2 & c_2 \\ Aa_3 & b_3 & c_3 \end{vmatrix}$$

$a_1 (b_2c_3 - c_2b_3) - a_1 (b_1c_3 - c_1b_3) + a_3 (b_1c_2 - c_1b_2)$

1.3.4 Addition of Matrices.

The sum of two matrices is the matrix (ij)th entry to the sum of the (ij)th entries of two matrices.

$$\begin{vmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{vmatrix} - \begin{vmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1-0 & 3-0 & 2-5 \\ 1-7 & 0-5 & 0-0 \\ 1-2 & 2-1 & 2-1 \end{vmatrix} = \begin{vmatrix} 1 & 3 & -3 \\ -6 & -5 & 0 \\ -1 & 1 & 1 \end{vmatrix}$$

The two matrices have the same dimensions.

Here $A + B = B + A$ is true.

1.3.5 Subtraction of Matrices.

The difference of two matrices is the matrix (ij)th entry to the difference of the (ij)th entries of two matrices.

$$\begin{vmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{vmatrix} \begin{vmatrix} 1-0 & 3-0 & 2-5 \\ 1-7 & 0-5 & 0-0 \\ 1-2 & 2-1 & 2-1 \end{vmatrix} = \begin{vmatrix} 1 & 3 & -3 \\ -6 & -5 & 0 \\ -1 & 1 & 1 \end{vmatrix}$$

The two matrices have the same dimensions.

1.3.6 Multiplication of two Matrices.

$$\begin{vmatrix} a_1 & a_2 \\ a_3 & a_4 \end{vmatrix} \begin{vmatrix} b_1 & b_2 \\ b_3 & b_4 \end{vmatrix} \begin{vmatrix} (a_1.b_1 + a_2.b_3) & (a_1.b_2 + a_2.b_4) \\ (a_3.b_1 + a_4.b_3) & (a_3.b_2 + a_4.b_4) \end{vmatrix}$$

Two matrices can be multiplied with each other even if the have different dimensions, as long as the following condition holds.

The number of columns in the first matrix is equal to the number of rows in the second matrix.

The result of multiplication called the product is another matrix with the same number of rows as the first matrix and the same number of columns as thee second matrix.

The multiplication of matrix is not commutative, this means in general that

A.B != B.A.

The multiplication of matrices is associative, that is

(A.B).C = A.(B.C)

1.4 Statement of the Problem.

Success in global economy requires a degree of sophistication and facility in mathematics and sciences. Yet at a time when mathematics achievement is important for individuals and the world as a whole, student's proficiency level remains low. This means they are also unprepared to compete in world where quantitative studies are increasingly important.

Contributing to student's poor performance is the difficulty schools face in attracting and retaining high-quality mathematics teachers. Studies suggest that teacher quality has a significant impact on student achievement in mathematics (see,e.g Brasswell et al. 2001 or Boyd et al. 2007). A November 2008 Educaton trust study found that 22% of all mathematics courses in secondary schools are taught by teachers without a degree in education nor an academic background in mathematics or mathematics related subjects.

Unquestionably this is partly explained by finances as public schools cannots compete with non-education private sector salaries (Murnane and Steele, 2007).

However, this means that unqualified teachers are particularly prevalent in high poverty, and high minority schools.

It is in the light of this reason that the researcher thought of the need to develop a computer aided instruction program to improve student performance in mathematics.

1.5 Aims and Objective of the Study.

The broad objective of this study is to design a computer aided instruction program in inverse matrix. The specific objectives are as follows.

- Making mathematics learning meaningful and interactive.
- Helping in retention of what has been learnt.
- Find out the beneficial effects of computer aided instruction program in mathematics.
- Delivering the same lesson to public and private schools.
- Improve students' performance in mathematics

1.6 Research Questions.

1. Can Computer-Aided instruction program improve student performance in mathematics?
2. Can computer-Aided instruction be used in capturing the effectiveness of teachers in some topics in mathematics?

1.7 Research Hypothesis.

The research obtained the expected outcome; the researcher formulated two null hypothesis viz.

1. Computer-aided instruction programs improve students' performance in mathematics.

2. Computer-aided instruction programs can be used in capturing the effectiveness of teachers in some mathematics topics.

1.8 Basic Assumptions of Study.

This study is based on the following assumptions.

1. Both the teachers and students are computer literate.
2. The school has a computer laboratory where the program will be installed.
3. The teacher has developed his course material the way it will be represented in the program.
4. The teacher is working with the software developer who does not need to be an expert in subject in question (mathematics).

1.9 Significance of the Study:

Many schools struggling to find creative, effective and feasible approaches to enhance mathematics achievement and compensate for the number of unqualified teachers, can turn to advances in computer-aided instruction programs. Although computer accessibility has increased over the past decade, the evidence on achievement has been mixed at best (see, e.g Wang Wang, and Ye, 2002 and Wenglinsky, 1998 for Contrasting findings).

Determining the potential benefits of computer-aided instruction programs could have significant implications both in addressing the need to improve stagnantly low levels of mathematics achievement solutions for schools and in identifying practical cost-effective solutions for school.

The primary benefit of computer-aided instruction program is an increase in the amount of quality instruction each student receives.

2.0 Scope and Limitations of Study.

The scope of the study is to find out the effect of the use of computer as a mathematics teaching aid/tool to improve students' performance in mathematics. Inverse matrix is used as the case study.

This research is potentially affected by two sources of bias.

1. Teachers may assign students they think would benefit from computerized instruction to the computer labs resulting in upward biased estimates.

2. If teachers who are assigned to the lab are those who are more willing to use computerized instruction, the estimated effects could be capturing the effectiveness of the teacher rather than the effectiveness of the computer program.

2.1 Definition of Terms

Computer: A computer is a programmable machine that receives input, stores and manipulates data, and provides output in a useful format.

Computer Program: A computer program is a sequence of instructions written to perform a specified task for a computer.

Programming Language: An artificial language designed to express computations that can be performed by a machine, particularly the computer.

Software: software is the collection of computer programs and related data that provide the instructions telling a computer what to do.

System: A system is a set of components that interact together to solve a problem.

Hardware: It is the physical components of a computer system.

Mathematics: The study of quantity, structure, space and change.

Matrix: mathematical object generally represented as an array of numbers.

Square Matrix: A matrix which has the same number of rows and columns.

Transpose: The transpose of an m-by-n matrix is the n-by-m matrix A formed by turning the rows into columns and vice versa.

Inverse Matrix: A matrix that when multiplied by another matrix, equals the identity matrix.

Determinant: A single number obtained from a matrix that reveals a variety of properties.

CHAPTER TWO

LITERATURE REVIEW

2.1 Learning Mathematics with Computer Tools.

Maria Trigueros in her research paper: Learning Mathematics with computer tools is of the opinion:

The use of computer tools is part of human living experience since 'such technologies are entwined in the practices used by humans to represent and negotiate cultural experience' (Trigueros, 2000, p. 170). Tools, as material devices and/or symbolic systems, are considered to be mediators of human activity. They constitute an important part of learning, because their use shapes the processes of knowledge construction and of conceptualization (Rabardel, 1999).

When tools are incorporated into students' activities they become instruments. Instruments are mixed entities that include both tools and the ways these are used. For this reason, instruments are not merely auxiliary components or neutral elements in the teaching of mathematics; they shape students' actions and therefore they are important components of the learning processes:

Instruments constitute the means that shape and mediate knowledge and our registers of situations and because of that they exert an influence that can be considerable… they influence the construction of knowledge (*ibid*, 1999, p. 204)

Every tool generates a space for action, and at the same time it poses on users certain restrictions. This makes possible the emergence of new kinds of actions. In that sense, the use of a tool can contribute in the opening of the space of possible actions for the learner (Rabardel, 1999). The influence that tools exercise on learning is not immediate in all cases. Actions are shaped gradually most especially, in a complex process of interactions.

Instruments are not given, they do not exist in themselves, and they do not imply a predetermined way of working. Rather, people incorporate tools into their activities and they shape them as they use them (ibid, 1999). Solving mathematical problems with the use of computer programmes is closely related to the tools available, and Lozano, Sandoval & TriguerosPME30 — 2006 4 - 91

Students need, on the one hand, adequate actions related to the mathematics involved and, on the other, actions that are effective in relation to the use of the tools themselves. In the classrooms, students construct meanings through their actions which are contextualized in phenomenological experience, that is, in a process of social interaction and with the guide of the teacher (Mariotti, 2001).

2.2 Analyzing the Way in Which the Use of Computer Interactive Programs Contributed in Shaping Students Actions.

Maria Dolores Lozano, Ivonne Twiggy Sandoval in their research paper: Investigating Mathematics Learning with the Use of Computer Programs in Primary Schools: opined that

Students are active when they work with programs. They constantly interact with the programs and with their peers.

They also ask questions and often want to explain or show things to the teacher and researchers.

Many students are eager to participate in whole-group discussions. A few of them are quiet, but all of them are attentive.

Students get distracted when, working with the interactive program, they cannot solve a problem after many attempts.

Individual work seems to be more frequent when students are working with activities from the textbook; when they start exploring the problem with the interactive program; and when their solutions are giving them unexpected feedback (due to incorrect answers).

Students appear to work in groups more frequently once they have an understanding of the problem.

The program seems to give s students freedom to explore with different situations and to experiment with different strategies.

The textbook and at times the teacher restrict students' actions.

Most students are looking for correct answers. This seems to be reinforced by the teacher who stresses the importance of getting them.

2.3 Using Computers in Mathematics Classroom

Michael Thomas, Jackie Tyrrell and John Bullock (2007) in their research paper: Using Computers in the Mathematics Classroom: The Role of the Teacher. Wrote the following:

The computer, due to its processing power, enables students to be exposed to a far greater range of problems and solutions than is possible with pen, paper and even calculator. This has implications for the kinds of learning that take place. Students could now explore possible solutions without vast expenditure of effort in calculating or graphing data. Thus in a single lesson they tried a variety of approaches to solving a problem. Unsatisfactory' results provided learning experiences, rather than being discarded as "wrong answers." Also students could quickly generate results for a large number of examples and then use this information to make decisions or generalizations.

They also observed the following

Many teachers discovered, to their surprise, that there was no sudden loss of respect or teacher control when they embraced the idea of learning alongside their students.

The development of critical analysis based on mathematical results became a real possibility.

2.4 Using Computers to Help Teachers.

Lisa Barrow, Elizabeth Debbragio(December 2008) in their paper: Failing in Mathematics was of the opinion:

For any software which can be used to encourage mathematical understanding, there is a minimum level of teacher preparation which is essential. For example, the teacher needs some understanding of the concepts on which the software is based, and some notion of the kinds of situations for which it can be used. With microworlds or other open-ended, software, it takes considerable time and experience to develop an understanding of the power of the underlying concepts, and of the kinds of situations that can provide a springboard for student learning.

The teachers tended to fall into two categories: those who explore, and those who plan. A preference for one approach emerged early in the project, and tended to persist even when the teacher had adopted a changed viewpoint.

The "explorers" are generally teachers who are thoroughly conversant with

both the computer and the mathematics they are teaching. They expect their

students to share their confidence, and are likely to model an enthusiastic, open ended exploratory behaviour. Their belief is that, given the right conditions, students have the inclination and ability to find their way around the software, and learn for themselves what they need to do for a particular task. Because these teachers have built up a thorough working knowledge of the computer (and perhaps the software) by trial and error rooted in their wide experience, they believe the students will do the same. They believe that computer skills are incidental; picked up in context as needed.

Teachers who plan appear to do so for a variety of reasons. They may feel confident about the mathematics but unsure about their computer knowledge. Some feel that too much uncertainty is inappropriate for their students. Their class may include a wide range of abilities, attitudes and social adjustment, and the teacher may want to be available to all students rather than spending time sorting out computer hassles. The teacher may wish to have time available during lessons to observe students, or to work with small groups on social skills: such as working collaboratively. One teacher of 11 to 13-year-old, children gave a vivid picture of her preparation for using new software. Her approach is to prepare carefully specified objectives, prepare resources very thoroughly, prepare instructions on how to get started-then leave the children to it.

The research also revealed the following:

During the lesson, the teacher was readily accessible, because the tasks themselves encouraged the students to be self-teaching rather than teacher-dependent.

The teacher, in fact, often just moved around the room observing pupil behaviour and interactions rather than being directly involved in "teaching".

The students were given the tools and experiences and tasks to learn for themselves the skill that the teacher had set.

2. 5 Can Computer Aided Instruction Help?

Holland, and Jamison Banerjee (December 2007) in their paper: The Role of Computer in Teaching was of the opinion,:

Concluding whether computer-aided instruction programs improve student performance in math is the obvious first step in formulating policy. While there are conflicting findings on this question, the majority of current research lacks randomised control study designs to account for factors like individual teacher effects and student ability that might be correlated with the use of computers in the classroom and student outcomes.

Meaningful policy discussion requires understanding the underlying reasons why a reform works. On this point, even among the studies that include a randomised evaluation of computer-aided instruction programs, few examine the mechanisms explaining why such technology either helps or hinders achievement. For example, Ragosta, Holland, and Jamison (1982) and Banerjee et al. (2007) both found beneficial effects of computer-aided instruction programs in math, but neither offers evidence on why these effects occur. While several hypotheses explaining the benefits of computer-aided instruction programs exist – including increased student engagement and motivation, and providing students with a greater level of individualised instruction – there is no supporting empirical evidence.

Barrow, Markman, and Rouse (2008), examines the effect of a popular pre-algebra/algebra computer-aided instruction program, I Can Learn, in three US urban schools districts suffering similar problems of underachievement and teacher recruitment. Each school district agreed to the implementation of a within-school random assignment design at the classroom level, thereby avoiding the sources of student and teacher bias previously described. Additionally, since I Can Learn subject lessons are designed so each student progresses through the material at her own pace and the teacher's primary role is to provide targeted help when needed, it is a well-suited program for testing the individualized instruction.

These findings are consistent with Ragosta, Holland, and Jamison (1982), Banerjee et al. (2007), and Wang, Wang, and Ye (2002) suggesting that computer-aided instruction programs can have a significant impact on student achievement levels in mathematics performance. However, effective policy implementation requires more than suggestive evidentiary support; policymakers must also have some understanding of how and under what circumstances the proposed reform works. When Barrow, Markman, and Rouse (2008) estimate models that include a classroom characteristic of interest, such as the average attendance in the prior year, the class size, or the heterogeneity in mathematics achievement, the data show greater effects for students in large, heterogeneous classes with poor attendance rates. Since these classroom characteristics would normally be disruptive and suggest a potential advantage to more individualised instruction, these findings support the theory that one of the primary benefits of a computer-aided instruction program is an increase in the amount of quality, individual instruction time each student receives.

CHAPTER THREE

SYSTEM DESIGN

3.1 Introduction:

The design methodology used for this program is OBJECT-ORIENTED DESGN (OOD).

Object-oriented design provides a natural and intuitive way to view the software design process namely, modelling objects by their attributes and behaviours just as we describe real-world objects. OOD also models communication between objects. Just as people send messages to one another (e.g., a sergeant commands a soldier to stand at attention), objects also communicate via messages. A bank account object may receive a message to decrease its balance by a certain amount because the customer has withdrawn that amount of money. An inverse matrix object can receive a message to calculate the determinant, cofactors, transpose of cofactors etc as we'll be seen in our final system.

3.2 Requirement Documents:

The design process begins by processing a requirement document that specifies the overall purpose of the system and what it must do. The requirement documents determines precisely what functions the system must include. This program is aim to allow users(students) to the following

A. Read lessons on inverse matrix.

B. Solve problems on inverse matrix(3 x 3).

The solutions to the problems will include the following

1. The determinant of a 3 x 3 matrix.

2. The matrix of cofactors.

3. The transpose of cofactors.

4. The inverse matrix (the transpose of cofactors divided by the determinant).

The user interface of the program will be the computer that will display the lecture contents to the user. The screen will also prompt the user to key in numeric inputs from the keyboard for the exercises that comes at the end of the lesson.

Upon first approaching the system, the screen displays a welcome message that tells the user to click ok to begin the lesson.

after the user has gone through the lesson the system prompts the user to enter three integer values for the first row. This prompt will continue until the user enters all the integers elements for the 3 x 3 matrix.

If the user inputs all the valid integers, the screen displays the appropriate results i.e.

The determinant

The matrix of cofactors.

The transpose of cofactors.

The inverse matrix.

3.3 Identifying the Classes in a System:

In this section, the researcher will identify the classes that are needed to to build the system by analyzing the nouns and nouns phrases that appear in the requirements documents. This classes will be described using the Unified Markup Language (UML) class diagrams and will be implemented in java.

First i'll review the requirements documents and identify the nouns and noun phrases to help us identify the classes that comprises the system. I may decide that some of these nouns donot correspond to the system and should not be modeeled at all.

NOTE: only the nouns that are relevant to the system are indentified.

Fig 3.1

The Nouns and Noun Phrases in the requirement document
User Screen Solve Inverse Matrix

Classes will only be created for nouns that have significance in the system.

Users will not be modelled as class because its an entity that is outside the system. Its only important to the system because it interacts with the system.

The screen will not be modelled because it is the computer screen will used for viewing the lessons.

The only class that will be modelled in this case study will be the Solve Inverse Matrix.

Modelling Classes:

The UML enables us to model classes in a system via class diagrams. In UML, each class is modelled as a rectangle with 3 compartments. The top compartment contains the name of the class centered horizontally in bold face, the middle compartment contains the class attributes, the bottom compartment contains the class operations. The middle and bottom compartments are empty because we've not yet determined the class attribute and operations.

Fig 3.2 UML class diagram for Solve Inverse Matrix

SOLVE INVERSE MATRIX

3.4 Identifying Attributes:

Classes have attributes (data) and operations (behaviour). Class attributes are implemented in Java programs as fields and class operations are implemented as methods. In this section I'll determine the many attributes needed in the system.

Consider the attributes of some real world objects; A person attributes include height, weight, complexion etc. A car attributes includes its speedometer and odometer readings, what gear it is in (manual or automatic). A personal computer attributes includes its manufacturer (e.g Dell, IBM, Sun, HP etc), the type of screen eg LCD or CRT, main memory size, hard disk size etc.

Attributes in a class can be identified by looking at the descriptive words and phrase in the requirement documents. Lets look at the descriptive words or pharse that describes a matrix.

In the case of a matrix, what is usually used to describe it are number of elements it contains. These elements correspond to its rows and columns. For the an Inverse Matrix (3x3) the attributes include the following :

Elements of row 1, elements of row 2, elements of row 3, the determinant, the matrix of cofactors for each of the elements etc.

Fig 3.3 UML Class Diagram for Solve Inverse Matrix Attributes

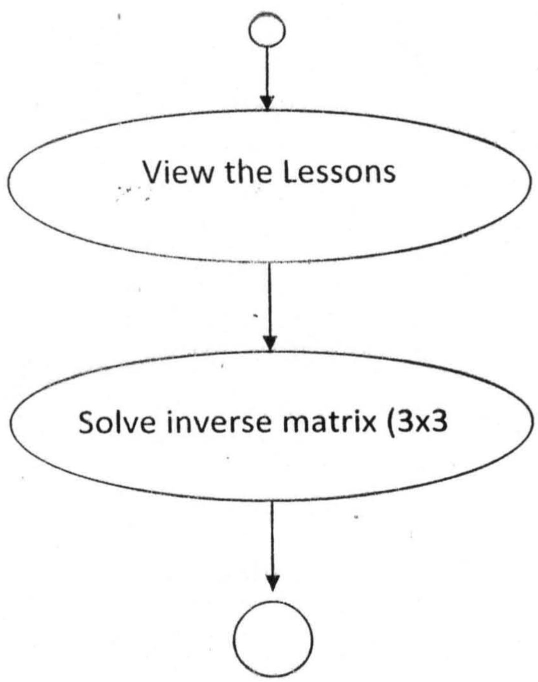
SOLVE INVERSE MATRIX
- a11: int - b12: int - c13: int - a21: int - b22: int - c23: int - a31: int - b32: int - c33: int - determinant: int - cofactorA11: int - cofactorB12: int - cofactorC13: int - cofactorA21: int - cofactorB22: int - cofactorC23: int - cofactorA31: int - cofactorB32: int - cofactorC33: int

3.5 Identifying Objects Activities

The object activities describes the workflow or activities the object performs on the system. In this case The activities of learning inverse matrix is presented.

Activity diagrams models aspects of system behaviour . it models the work flow (sequence of events) during program exexcution. An activity diagram models the actions the object will perform and in what order.

Fig 3.4 Activity Diagram for Solve Inverse Matrix Object



3.6 Identifying Operations:

An operation (or behaviour) is a service that objects of class provide to clients of the class. Lets look some of the operations of some real world objects. A radio's operation include setting its station and volume etc. A car operation include accelerating, decelerating(invoked by the driver pressing the break pedal), turning and shifting gears.

Software objects offer operations as well for example: a software graphic object might offer operations for drawing a circle, drawing a line, drawing a square and the like. A spread sheet software object might offer operations like printing the spreadsheet, totaling the elements in a row or column and graphing information in the spreadsheet as a bar chart or pie chart.

We can derive many of the operations of each class by examining the key verbs and verb phrases in the requirements document. We then relate each of these to particular classes in our system. The verb phrases help us determine the operations of each class

Fig 3.5 Verbs and verb phrases for each class in the System.

Class	Verb and Verb Phrases
Screen	Displays the lecture contents (notes)
Solve Inverse Matrix	Find the determinant, Find the matrix of cofactors, transpose the cofactors, and divide the transpose of the cofactors by the determinant.

3.7 Modelling Operations:

To identify operations, we examine the verb phrases listed for each class in the system The "Find the determinant" phrase associated with class solve inverse matrix implies that class Solve Inverse Matrix instructs the method determinant to execute i.e. perform an operation(to calculate the determinant of the 3x3 matrix). During finding inverse matrix session, solve inverse matrix object will invoke the operations (determinant, cofactors, transpose of cofactors, inverse) to tell what to do. This means calculating each of these.

Fig 3.6 Uml Class diagram for Solve Inverse Matrix Class With Attributes and Operations

SOLVE INVERSE MATRIX
- a11: int - b12: int - c13: int - a21: int - b22: int - c23: int - a31: int - b32: int - c33: int - determinant: int - cofactorA11: int - cofactorB12: int - cofactorC13: int - cofactorA21: int - cofactorB22: int - cofactorC23: int - cofactorA31: int - cofactorB32: int - cofactorC33: int
Determinant (): integer Cofactors (): integer Transpose (): integer Inverse (): integer

3.8 Choice of Programming Language

Java programming is used not only because of the task it intends to perform but of its object oriented capabilities (ability to support modular programming). It contains swings which are used for designing the user interface. It provides a complete list of tools to simplify Rapid Application Development (RAD).

3.9 System Requirements:

The program will run a system with the following configuration:

Operating System: Microsoft windows 200/xp/vista/7/server 2000/2003, Linux.

Java Virtual Machine (JVM).

Hard Disk Size: 50 megabytes or more.

Processor: Intel Pentium, AMD.

Random Access Memory (RAM): 128.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 Introduction:

Once the design is complete, most of the major decisions about the system have been made. The goal of the coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly. A well written code reduces the testing and maintenance effort. Since the testing and maintenance cost of software are much higher than the coding cost, the goal of coding should be to reduce the testing and maintenance effort. Hence, during coding the focus should be on developing programs that are easy to write. Simplicity and clarity should be strived for, during the coding phase.

4.2 Software Testing:

Testing is the major quality control measure employed during software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and non-executable. After the coding, computer programs are available that can be executed for testing. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of testing is to uncover requirement, design or coding errors in the programs.

The program can be tested by using an illustration, suppose a student was ask find the inverse of a 3x3 matrix, this program has the ability to present the lessons and at the end of it gives the result based on user inputs (integers) The results presented by this program are the following: the determinant, The matrix of Cofactors, The transpose of cofactors and the Inverse.

4.3 Presentation of Result:

WELCOME TO THE CLASS ON FINDING INVERSE OF A SQAUARE MATRIX.

THIS PERSONAL TUTOR WILL TAKE THROUG THE STEPS ON OF FINDING THE INVERSE MATRIX.

The adjoint of a square matrix is important, since it enables us to form the value of the determinant of A

(Provided A is not equal to 0).

To form the inverse of a square matrix A:

(A). Evaluate the determinant of A i.e. |A|.

(B).Form a matrix of the Cofactors of the elements of |A|

(C).Write the transpose of C, i.e. C to obtain the elements of |A|

(D).Divide each element of the transpose of C by ?A

(E).The resulting matrix is the inverse A?╣ of the original matrix.

To find the inverse of A = | 1 2 3 |

| 4 1 5 |

| 6 0 2 |

(A). Evaluate the determinant of , i.e. |A|

|A| | 1 2 3 | =1(2-0)-2(8-30)+3(0-6) = 28

| 4 1 5 |

| 6 0 2 |

(B). Now find the Matrix of the cofactors.

A11 = +(2-0) = 2; A12 = -(8-30) = 22; A13 = +(0-6) = -6

A21 = -(4-0) = -4; A22 = +(2-18) = -16; A23 = -(0-12) = 12

A31 = +(10-3) = 7; A32= -(5 - 12) = 7; A33 = +(1-8) = -7

The transpose of cofators = | 2 22 -6 |

| -4 -16, 12 |

| 7 7 -7 |

(C). Next we have to write the transpose of the C to obtain adjoint of A.

Adj A = CT = | 2 -4 7 |

| 22 -16 7 |

|-16 12 -7 |

(D). Finally, we divide the elements of adj A by the value of|A|,i.e.28, to arrive at the inverse of A

| 22/28 -4/28 7/28 |

| 22/28 -16/28 7/28 |

| -6/28 12/28 -7/28 |

Everyone is done the same way. Work the following right through on your own.

Determine the inverse of the matrix A = | 2 7 4 |

| 3 1 6 |

| 5 0 8 |

Determine the inverse of the matrix A = | 2 1 4 |

| 3 5 1 |

| 2 0 6 |

Enter 3 integer values for the first row 2 7 4

Enter 3 integer values for the second row 3 1 6

Enter 3 integer values for the third row 5 0 8

The Determinant is:38

The matrix of cofactors are

8 6 -5

-56 -4 35

38 0 -19

The Transpose of the Cofators are

8 -56 38

6 -4 0

-5 35 -19

THE INVERSE OF THE MATRIX IS

8/38 -56/38 38/38

6/38 -4/38 0/38

-5/38 35/38 -19/38

Determine the inverse of the matrix A = | 2 1 4 |

| 3 5 1 |

| 2 0 6 |

Enter 3 integer values for the first row 2 1 4

Enter 3 integer values for the second row 3 5 1

Enter 3 integer values for the third row 2 0 6

The Determinant is:4

The matrix of cofactors are

30 -16 -10

-6 4 2

-19 10 7

The Transpose of the Cofactors are

30 -6 -19

-16 4 10

-10 2 7

THE INVERSE OF THE MATRIX IS

30/4 -6/4 -19/4

-16/4 4/4 10/4

-10/4 2/4 7/4

4.3 Analysis of Result:

The result is based purely on user inputs (when the system prompts the user to enter the elements of the matrix). This system of solving inverse matrix is, faster, interesting and engaging. It allows the user to get immediate answers to the problems compare to the traditional method where the students need to check the answer page of the text book for answers.

4.4 Maintenance:

Maintenance includes all the activity after the installation of software that is performed to keep the system operational. As we have mentioned earlier, software often has design faults. The two major forms of maintenance activities are adaptive maintenance and corrective maintenance.

It is generally agreed that for large systems, removing all the faults before delivery is extremely difficult and faults will be discovered long after the system is installed. As these faults are detected, they have to be removed. Maintenance activities related to fixing of errors fall under corrective maintenance.

Removing errors is one of the activities of maintenance. Maintenance also needed due to a change in the environment or the requirements of the system. The introduction of a software system affects the work environment. This change in environment often changes what is desired from the system. Furthermore, often after the system is installed and the users have had a chance to work with it for some time, requirements that are not identified during requirement analysis phase will be uncovered. This occurs, since the experience with the software helps the user to

REFERENCES

A Jones & P. Scrimshaw (Eds.). Computers in Education (pp. 203-:-229). Milton Keynes: Open University.

A V. Kelly (Ed.), Microcomputers and the curriculum (pp. 20-35). London: Harper & Row.

Ainley, J., & Pratt, D. (1995). Planning for portability: Integrating mathematics and technology in the primary curriculum.

Barrow, Lisa, Lisa Markman, and Cecilia Elena Rouse. 2008. "Technology's Edge: The Educational Benefits of Computer-Aided Instruction." National Bureau of Economic Research Working Paper 14240.

Britt, M., Irwin, K... Ellis, J., & Ritchie, G. (1993). Teachers raising achievement in mathematics:

Burton & B. Jaworski (Eds.), Technology in mathematics teaching: A bridge between teaching and learning (pp. 435-448). Bromley, UK:

Davis, B., Sumara, D. and Luce-Kapler, R. (2000). Engaging Minds: Learning and Teaching in a Complex World. London:

Deitel, Harvey and Paul Deitel (2005). Java How to Program:

Freudenthal, H. (1981). Major problems of mathematics education. Educational Studies in Mathematics, 12(2), 133-150.

Fraser, R, Burkhardt, H., Coupland, J., Phillips, R., Pimm, D., & Ridgway, J. (1988). Learning activities and classroom roles with and without the microcomputer.

G. E. Kearsley (Ed.), Artificial intelligence and instruction: Applications and methods. Addison-Wesley.

Hoyles, C. (1992). Mathematics teaching and mathematics teachers: A meta-case study. For the Learning of Mathematics, 12(3), 32-44.

Irwin, K., Britt, M., Ellis, J., & Ritchie, G. (1993). Teacher change in the mathematics classroom. Paper presented at the New Zealand Association for Research in Education Conference, Hamilton, New Zealand.

Kaput, J. (1992). Technology and mathematics education. Grouws (Ed.), Ha'ndbook of research on mathematics teaching and learning (pp. 515-556).

Lisa Barrow, Elizabeth Debragio Cecilia, Elena Rouse "Failing in mathematics". 23 december 2008.

Logan, L., & Sachs, J. (1987). School based in service education. St. Lucia, QLD: Department of Education, University of Queensland.

Maria Dolores Lozano, Ivonne Twiggy, Maria Triguerios. Invesigating Mathematics Learning with the use of Computer Programmes in Primary Schools.

Micheal Thomas, Jackie Tyrell and John Bullock. Using Computers in the Mathematics

Classroom: The Role of the Teacher. Mathematics Education Research Journal, 1996, Vol 8, No: 1, 38-57.

Murnane, Richard J. and Jennifer L. Steele. 2007. "What is the Problem? The Challenge of Providing Effective Teachers for All Children." The Future of Children, 17(1): 15-43.

Reston, VA: National Council of Teachers of Mathematics.

Smith, L. A (1984). Overcoming computer-induced anxiety.

Thompson, P. W. (1985). Mathematical microworlds and intelligent computer assisted instruction.

Treagust, D. F., & Rennie, L. J. (1993). Implementing technology in the school curriculum:

Wang, Xiaoping, Tingyu Wang, and Renmin Ye. 2002. "Usage of Instructional Materials in High Schools: Analyses of NELS Data." Paper presented at the Annual Meeting of American Educational Research Association, New Orleans, LA.

Wenglinsky, Harold. 1998. "Does it Compute? The Relationship Between Educational Technology and Student Achievement in Mathematics.

SOURCE CODE

```
import java.util.Scanner;



public class Matric

{

        private int a1;

        private int b1;

        private int c1;

        private int a2;

        private int b2;

        private int c2;

        private int a3;

        private int b3;

        private int c3;


        private int deter;

        private int cofacA1;

        private int cofacB1;

        private int cofacC1;

        private int cofacA2;

        private int cofacB2;

        private int cofacC2;
```

```
        private int cofacA3;

        private int cofacB3;

        private int cofacC3;


public void determinant()

{

                System.out.println();

        System.out.println("WELCOME TO THE CLASS ON FINDING INVERSE OF A
SQAUARE MATRIX.");

        System.out.println();

                System.out.println("THIS PERSONAL TUTOR WILL TAKE THROUG THE
STEPS ON OF FINDING THE INVERSE MATRIX.");

                System.out.println();

                System.out.println("The adjoint of a square matrix is important,since it enables us
to form the value of the determinant of A");

                System.out.println("(Provided A is not equal to 0).");

                System.out.println();

                System.out.println("To form the inverse of a square matrix A:");

                System.out.println();

                System.out.println("(A). Evaluate the determinant of A i.e.  |A|.");

                System.out.println();

                System.out.println("(B).Form a matrix of the Cofactors of the elements of |A|");

                System.out.println();

                System.out.println("(C).Write the transpose of C, i.e. C to obtain the elements of
|A|");

                System.out.println();
```

```
System.out.println("(D).Divide each element of the transpose of C by ?A");

System.out.println();

System.out.println("(E).The resulting matrix is the inverse A?¹ of the original
matrix.");

System.out.println();

System.out.println("To find the inverse of A = | 1  2  3 |");

System.out.println("                          | 4  1  5 |");

System.out.println("                          | 6  0  2 |");

System.out.println();

System.out.println("(A).      Evaluate the determinant of , i.e. |A|");

System.out.println();

System.out.println("|A|  | 1  2  3 |    =1(2-0)-2(8-30)+3(0-6)  = 28");

System.out.println("     | 4  1  5 |");

System.out.println("     | 6  0  2 |");

System.out.println();

System.out.println("(B).      Now find the Matrix of the cofactors.");

System.out.println();

System.out.println("A11 =  +(2-0) =  2;  A12 = -(8-30)  =  22;  A13 = +(0-6)  = -
6");

System.out.println();

System.out.println("A21 =  -(4-0) = -4;  A22 = +(2-18)  = -16;  A23 = -(0-12) =
12");

System.out.println();

System.out.println("A31  = +(10-3) = 7;  A32= -(5 - 12) = 7;   A33 = +(1-8)  = -
7");

System.out.println();
```

```
System.out.println("The transpose of cofators = |  2  22  -6 |");
System.out.println("                            | -4 -16  12 |");
System.out.println("                            |  7   7  -7 |");
System.out.println();
System.out.println("(C). Next we have to write the transpose of the C to obtain adjoint of A.");
System.out.println();
System.out.println("Adj A = CT = | 2  -4  7 |");
System.out.println("             | 22 -16  7 |");
System.out.println("             |-16  12 -7 |");
System.out.println();
System.out.println("(D). Finally, we divide the elements of adj A by the value of|A|,i.e.28, to arrive at the inverse of A ");
System.out.println();
System.out.println("| 22/28     -4/28     7/28 |");
System.out.println("| 22/28    -16/28     7/28 |");
System.out.println("| -6/28     12/28    -7/28 |");
System.out.println();
System.out.println("Everyone is done the same way. Work the following right through on your own.");
System.out.println();
System.out.println("Determine the inverse of the matrix A = | 2  7  4 |");
System.out.println("                                        | 3  1  6 |");
System.out.println("                                        | 5  0  8 |");
System.out.println();
System.out.println("Determine the inverse of the matrix A = | 2  1  4 |");
```

```
        System.out.println("                              | 3  5  1 |");
        System.out.println("                              | 2  0  6 |");
        System.out.println();
        System.out.println();
        System.out.println();
        System.out.println();


Scanner input = new Scanner(System.in);


System.out.print("Enter 3 integer values for the first row");
int a1 = input.nextInt();
int b1 = input.nextInt();
int c1 = input.nextInt();
        System.out.println();
System.out.print("Enter 3 integer values for the second row");
int a2 = input.nextInt();
int b2 = input.nextInt();
int c2 = input.nextInt();
        System.out.println();
System.out.print("Enter 3 integer values for the third row");
int a3 = input.nextInt();
int b3 = input.nextInt();
int c3 = input.nextInt();
        System.out.println();
```

```
int result = deter(a1,b1,c1,a2,b2,c2,a3,b3,c3);

int cofacA1 = b2*c3-b3*c2;

int cofacB1 = -(a2*c3-a3*c2);

int cofacC1 = a2*b3-a3*b2;

int cofacA2 = -(b1*c3-b3*c1);

int cofacB2 = a1*c3-a3*c1;

int cofacC2 = -(a1*b3-a3*b1);

int cofacA3 = b1*c2-b2*c1;

int cofacB3 = -(a1*c2-a2*c1);

int cofacC3 = a1*b2-a2*b1;


System.out.println("The Determinant is:" + result);
System.out.println();
```

```
System.out.println("The matrix of cofactors are" );
System.out.printf("\n%d %d %d\n", cofacA1, cofacB1, cofacC1);
System.out.printf("\n%d %d %d\n", cofacA2, cofacB2, cofacC2);
System.out.printf("\n%d %d %d\n", cofacA3, cofacB3, cofacC3);
System.out.println();
System.out.println("The Transpose of the Cofators are");
System.out.printf("\n%d %d %d\n", cofacA1, cofacA2, cofacA3);
System.out.printf("\n%d %d %d\n", cofacB1, cofacB2, cofacB3);
System.out.printf("\n%d %d %d\n", cofacC1, cofacC2, cofacC3);
System.out.println();
System.out.println("THE INVERSE OF THE MATRIX IS");
            System.out.printf("\n%d/%d %d/%d %d/%d\n", cofacA1,deter, cofacA2, deter,
cofacA3, deter);
                System.out.printf("\n%d/%d %d/%d %d/%d\n", cofacB1,deter, cofacB2,
deter, cofacB3, deter);
                    System.out.printf("\n%d/%d %d/%d %d/%d\n", cofacC1,deter,
cofacC2, deter, cofacC3, deter);
                        System.out.println();
}

public int deter(int a1,int b1,int c1, int a2,int b2, int c2, int a3,int b3,int c3)
{

deter = a1*(b2*c3-b3*c2)-b1*(a2*c3-a3*c2)+c1*(a2*b3-a3*b2);
```

```
return deter;

  }

}
```

Test Program

```
public class MatricTest
{
        public static void main(String args[])
        {
                Matric matric = new Matric();
                matric.determinant();
        }
}
```