

KARADENIZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

COMPUTER ENGINEERING GRADUATE PROGRAM

**QR CODE-BASED ENCRYPTION AND DECRYPTION OF TRIANGULAR
GEOMETRY PROBLEMS**

MASTER THESIS

Computer Eng. CHEIKHNA LO

AUGUST 2019
TRABZON

KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE
COMPUTER ENGINEERING GRADUATE PROGRAM

**QR CODE-BASED ENCRYPTION AND DECRYPTION OF TRIANGULAR
GEOMETRY PROBLEMS**

Computer Eng. CHEIKHANA LO

**This Thesis is Accepted to Give The Degree of
“MASTER OF SCIENCE IN COMPUTER ENGINEERING”
By
The Graduate School of Natural and Applied Science at
Karadeniz Technical University**

**The date of Submission : 16.05.2019
The date of Examination: 22.07.2019**

Thesis Supervisor: Asst. Prof. Dr. Hüseyin PEHLİVAN

Trabzon 2019

KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING GRADUATE PROGRAM
CHEIKHNA LO

**QR CODE-BASED ENCRYPTION AND DECRYPTION OF TRIANGULES
GEOMETRY PROBLEMS**

**Has been accepted as a thesis of
MASTER OF SCIENCE
after the Examination by the Jury Assigned by the Administrative Board of
the Graduate School of Natural and Applied Sciences with the Decision Number
dated 21/05/2019**

Approved By

Chairman : Prof. Dr. Abdulsamet HAŞILOĞLU

Member : Asst. Prof. Dr. Hüseyin PEHLİVAN

Member : Asst. Prof. Dr. İbrahim SAVRAN

**Prof. Dr. Asim KADIOĞLU
Director of Graduate School**

ACKNOWLEDGEMENT

I want to express my gratitude and thanks to my advisor Dr.Öğr.Üyesi Hüseyin PEHLİVAN at KARADENİZ TECHNICAL UNIVERSITY, I thank him for having framed, guided, helped and advised, and thank also ARŞ. GÖR. MEHMET CEMİL AYDOĞDU his office was always open whenever I come with a trouble.

I extend my sincere thanks to all the teachers, speakers and all the people who have their words, their writings, their advice and their critics guided my thoughts and agreed to meet me and answer my questions during my research.

I also want to thank all my family, my friends, and all people who helped me from far or close because without all this people around I wouldn't finish this great work.

CHEIKHNA LO

Trabzon 2019

THESIS STATEMENT

I declare that, this Master Thesis, I have submitted with the title “Qr code-Based Encryption and Decryption of Triangular Geometry problems” has been completed under the guidance of my Master supervisor Asst. Prof. Dr. Hüseyin PEHLİVAN.

I have complied this work with all examination, moral principles and following the rules of my university KTU during my work on this, and I acknowledge all obligation whenever demonstrated something else. 30.07.2019

CHEIKHNA LO

TABLE OF CONTENTS

	<u>Page No</u>
ACKNOWLEDGEMENT.....	IV
STATEMENT.....	V
TABLE OF CONTENTS	VI
SUMMARY	X
ÖZET	XI
FIGURES LIST	XII
TABLE LIST.....	XIII
ABBREVIATION.....	XIV
1. INTRODUCTION.....	1
2. REVIEW	4
3. GENEL INFORMATION.....	7
3.1. Qr-Code	7
3.2. Cryptography	8
3.2.1 Encryption and Description Algorithm	9
3.3 Formal Languages	10
3.3.1. Type of Grammars.....	11
3.3.2 Parsing Problems	11
3.3.2.1 Ambiguity.....	11
3.3.2.2 Recursive Productions	12

3.3.2.3.	Left Factoring	12
3.3.3.	Parsing Technique	13
3.3.3.1.	First and Follow Sets	13
3.3.3.2.	Follow Set.....	13
3.3.3.3.	LR Parser	13
3.4.	Triangles and Their Types.....	14
3.5.	Language Processor.....	15
3.5.1.	Compilers	15
3.5.2.	Interpreters.....	16
3.5.3.	Variations between Compilers And Interpreters	17
3.5.3.1.	Compiler Phases	17
3.5.3.1.1.	Front-End.....	18
3.5.3.1.2.	Back-End	19
3.5.3.2.	Language Interpreters	19
3.6.	Lexical Analysis	20
3.6.1.	Interaction of Lexical Analysis With Parser	20
3.6.2	Issues in Lexical Analyzer.....	22
3.6.3.	Regular Expressions	22
3.7.	Syntax Analysis (Parsing)	23
3.8.	JavaCC.....	23
3.9.	Mobile App.....	24

4.	METHODOLOGY	26
4. 1.	The Relevant Tools and Technologies	26
4. 2.	Solving Triangular Problems.....	26
4.2.1.	Prototypes and Mock-Ups	27
4.2.2.	Error/Restart Program.....	29
4.3.	Analysis Phase	30
4.3.1.	Lexical Analysis	30
4.3.1.1.	Interaction of LA with the Parser	32
4.3.1.2.	Lexeme, Token and Pattern	33
4.3.2.	Syntax Analysis	34
4.3.3.	Semantic Analyzer.....	36
4.4.	Synthesis Phase	37
4.4.1.	Intermediate Code Generation.....	37
4.4.2.	Code Optimization.....	37
4.4.3.	Code Generation.....	38
4.5.	A Comparison of Some Encryption Algorithms.....	38
5.	STEP-BY-STEP THE ILLUSTRATION OF THE OUR APPLICATION.....	40
5.1.	The Input Data.....	40
5.1.1.	Compiler Phase	40
5.1.2.	Encryption Phase.....	41
5.1.3.	Qrcode Phase.....	42
5.1.3.1.	Generating Qrcode	42

5.1.3.	Scanning Qrcode	43
6.	CONCLUSION	46
7.	FUTURE WORK	47
8.	REFERENCES	48
	CURRICULUM VITAE	51



Master Thesis

SUMMARY

QR CODE-BASED ENCRYPTION AND DECRYPTION OF TRIANGULAR
GEOMETRY PROBLEMS

CHEIKHANA LO

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Asst. Prof. Dr. Hüseyin PEHLIVAN
2019, 50 Pages

Our thesis focuses on the planning and implementation of a QR code-based encryption and decryption system for triangular pure mathematics issues. The encoding stage begins with an event of a context-free synchronic linguistics to explain triangular issues during a formal language. The geometric descriptions of triangles are then encoded employing an ancient cryptography formula and the corresponding QR code is generated within the cryptography stage. Then the QR code is decrypted and a programmed that is mechanically created by the JavaCC program is used to analyze and show the related problem graphically. The developed system provides QR code reader like tool and presents in a chic way to be able to show mathematical issues on various devices simply. With such a tool, the full queries of associate examination is encoded in QR code and hold on in smaller sizes. This is able to considerably increase the protection of exams and reduce question data, which might be notably transmitted on smartphone, in small size.

Key words: Security, geometry, grammar, encryption, QR-code, geometry

Yüksek Lisans Tezi

ÖZET

ÜÇGENSEL GEOMETRİ PROBLEMLERİNİN QR KOD TABANLI ŞİFRELENMESİ
VE GÖRÜNTÜLENMESİ

CHEIKHNA LO

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Hüseyin PEHLIVAN
2019, 46 Sayfa

Bu çalışma, üçgen geometri problemleri için QR kod tabanlı şifreleme ve şifre çözme sisteminin tasarımına ve uygulanmasına odaklanmaktadır. Şifreleme aşaması, biçimsel bir dilde üçgen sorunları tanımlamak için bağlamsız bir gramer geliştirilmesi ile başlar. Üçgen açıklamaları daha sonra geleneksel bir şifreleme algoritması kullanılarak şifrelenir ve son olarak karşılık gelen QR kodu üretilir. Şifre çözme aşamasında, QR kodunun şifresi çözülür ve ilgili sorunu grafik olarak analiz etmek ve göstermek için JavaCC aracı tarafından otomatik olarak üretilen bir çözümleyici kullanılır. Geliştirilen sistem, QR kod okuyucu benzeri bir araç sağlar ve matematiksel sorunları çeşitli aygıtlarda kolayca görüntüleyebilmek için zarif bir yol sunar. Böyle bir araçla, sınavın bütün soruları QR kodunda kodlanabilir ve daha küçük boyutlarda saklanabilir. Bu, sınavların güvenliğini önemli ölçüde artıracak ve özellikle mobil cihazlarda iletilebilecek soru verilerini boyut olarak azaltacaktır.

Anahtar Kelimeler: Biçimsel dilbilgisi, şifreleme, QR kodu, güvenlik, geometri, Bağlamsız gramerler.

LIST OF FIGURES

	<u>Pages No</u>
Figure 1. QR-code example.....	7
Figure 2. Caesar-cipher technique.	8
Figure 3. Symmetric Key Cryptography.....	10
Figure 4. Example of Ambiguity	12
Figure 5. Structure of a Compiler.....	18
Figure 6. Frond-End Analysis.	18
Figure 7. Back-End Synthesis.	19
Figure 8. Lexical-Analyzer to Parser.....	20
Figure 9. Parsing Phase.	23
Figure 10. JavaCC Compiler.....	23
Figure 11. QRCODE Scanned.	42
Figure 12. The Interface Student Scanning Phase.....	43
Figure 13. The Interface after the QRcode Scan.	44

LIST OF TABLES

	<u>Page No</u>
Table 1. All Triangles Types.....	14
Table 2. Tokens and Lexemes.....	21
Table 3. Tokens, Lexemes and Pattern.....	21
Table 4. Part of Check Method.....	31
Table 5. Convert the Code to Tokens.....	32
Table 6. The Typical Tokens.....	33
Table 7. The Declaration of our Tokens.....	34
Table 8. Our Grammar.....	35
Table 9. Our Syntax Source Code.....	35
Table 10. Parser fille.jj.....	36
Table 11. A Comparison of AES, DES and 3DES Algorithms.....	38
Table 12. A Sample Input.....	39
Table 13. The AES Algorithm.....	41

ABBREVIATION LIST

EBNF	Extended Backus Naur Form
BNF	Backus Naur Form
CFG	Context Free Grammar
AST	Abstract syntax tree
JavaCC	Java Compiler Compiler
Zxing	Zebra Crossing library
AES	Advanced Encryption Standard or System
QRcode	Quick Response Code
ASCII	American Standard Code for Information Interchange
SSS, SAS, ASA	Side Side Side, Side Angle Side, Angle Side Angle
AAS/SAA, SSA/ASS	Angle Angle Side, Side Angle Angle, Angle Side Side

1. INTRODUCTION

Technology has and can still impose on amendments that can affect all industries. Often, the outcomes of technology on any trade are inherently in trouble [1]. We have the transport trade as an example; the expected exploitation of self-driving technology threatens voluminous jobs across the planet. We also have publication trade as another example. The increase of the online media has forced old school publication corporate companies to rethink their business plans. The same amendment is being implemented within the education sector [2]. The implementation of technology is reshaping our understanding of the education sector. Here is a number of the foremost profound ways in which technology has modified education within the last few years [3]. These are some of the most profound ways in which technology has changed education 1- the Increasing of the accessibility, 2- flexibility, 3- interaction between professors and students, 4- on-line Tests And Assessments, 5-New Content, 6-Special requests In Education, 7-Long term Learning, 8- the increase Of Mobile Learning Content, 9- price Reduction, 10-combining fun with education and last but not least 11- the increase Of Mobile Learning Content [1].

Recent technology advancement has opened up many potential of QR codes to enhance the educational process by changing the classical method [4] of storing and presenting information, this potential will have increase the opportunity of being able to learn anywhere in the world at their convenient time which in return will improve the overall quality of the process [5,6,1]. The number of researches that address the issue of using QR codes in education is still few [1]. However some of those studies have proven significant improvement results which indicates that such technology could even expand to further applications, it is not only limited to usage in exams environment but it can also be well expanded to include pictures, text books, notes and even videos [7].

In other words, the abilities of the QR code will expand as it is used more in the education [8] process, while enriching the classical methods it will slowly replace it in a more efficient and easier way [9]. Yet however to prove these predictions right we will have to conduct further more research to top into the pros and cons of such implementation[10] and further development must be carried out to improve this sector, which was one of the motives behind this work[11,12,13].

Mathematics is one of all the foremost vital areas of scientific studies for human beings, which is applicable in most fields: science, technology, business, medicine, meteorology, astronomy, etc. From elementary addition and subtraction to rocket science, arithmetic is all around. Mathematics is a huge field of studies and has developed throughout history [14]. It is divided into many disciplines. Let us take a glance at a number of the popular disciplines.

Arithmetic, which is the oldest discipline of mathematics, deals with numbers and elementary operators (addition, subtraction, multiplication and division).

Algebra, which is the widely used discipline of mathematics, involves the study of mathematical variables, functions, equations as well as the rules related to them.

The other discipline is the geometry, which is the study of shapes, sizes, distance, location, etc. It covers length, space, volume, in one, two and three-dimensional planes. There are several applications in our daily life, like finding the world of a plot of land, finding the volume of a gas cylinder, etc. It conjointly has several advanced applications, like within the field of physical science to mark locations of stars, calculative distances between two celestial bodies, etc... [14].

Trigonometry is the discipline involving the study of the relationship between sides and angles of a triangle. Most typically it's used for right triangles. Its special functions, referred to as trigonometric functions that relate the angles and sides. In world applications, it's used unremarkably for finding heights of towers, buildings, mountains, etc. Trigonometric functions are employed in the study of waves and their characteristics.

In the past decades, continuous improvement of programming in engineering has enabled to develop useful programs to unravel human issues. Mathematics encompasses a very important role in human life. The mathematical operations employed in engineering applications can't be accomplished by human hands, therefore, mathematical programs for with efficiency determination of mathematical issues were developed. The mathematical programs or scientific programs are employed for mathematical modeling and statistical analysis.

In this work, we are trying to focus on the results and outcomes of implanting an advanced system such as QR codes based[8] on encoding and decoding of triangles geometry problems, in an old school system such as education, which can help teachers in preparing questions papers. Teachers traditionally provide students with a questions papers without encryption, which is open to an exam cheating [7]. The use of the QR codes based encryption system makes it easier for teachers to prepare the papers in a more secure way.

Teachers can prepare a question paper through an interface and edit the questions data. The encryption stage encrypts a questions paper to ensure that it cannot be easily seen and cheated on by students. In this way, teachers can use the system to prevent cheating on a test and store the questions data in a smaller size. The information was obtained by conducting interviews and surveys with the learners then it was compared with the results of self-evaluation of the learners. The encoding phase begins with developing of context-free grammar to indicate such triangles issues in a formal language. In the decoding phase the QR codes decoded to retrieve the data of the question back. Then the data are analyzed by a parser, which can be automatically produced by the JavaCC tool, and shown in a question format graphically.



2. LITERATURE REVIEW

For years, many colleges and districts have had strict policies forbidding the use of students' personal electronic devices in school rooms [14, 15]. However, some faculties are starting to embrace the academic worth of hand-held Web-enabled devices that students already bring to faculty every day. As academics begin to explore the academic opportunities that smartphones and different devices provide [14], it is good to remember that the utility of any tool is simply pretty much as good as our understanding of the way to use that tool. This text makes an attempt to introduce two easy ways in which within which QR codes will be used with success within the schoolroom [17].

One the most time consuming processes in the university is taking the daily attendance, with the rise of popularity of smartphone over any other utility such as desktops or laptops especially with users above the age 26, it shows a great indicators of possible application of such system in the university systems.[18]

Geometry is the one of the oldest and most important branch of mathematics as it develop and reshapes the students' ability to think crucially and come to valid logical solutions, not to mention that it is a crucial fundamental of other arithmetic.

In a journal published by Mustafa Zeki et. 2014 about solving geometry issue, based on the results that he concluded all educator participants to create a strategy to take care of a geometry issue. He faced a question which was should one improve the problem visually or as an expression to solve out the problem faster [29].

As well M. Tolga SAKALLI et al., 2004, describes on their paper mathematical concepts using cryptography to make students more enthusiastic about mathematics. In their study, they represented cryptosystems systems associated with mathematics to indicate that it's doable to show a student these ideas using illustrative. Additionally, there are alternative cryptosystems like hill ciphers, affine ciphers, etc. they'll even be accustomed to illustrate alternative mathematical fundamentals. Using cryptography as a teaching tool can facilitate students: To develop their skills in mathematics science and to grasp mathematical ideas effectively.

To tap into more researches, Atul Hole al. 2014 submitted a research on how the QR-code will be embedded in the education and the technical issues, which will demotivate creative teachers and educators to apply them in their teaching methods.

The security of the info could be a massive drawback. And to resolve this drawback they projected an efficient technique to demonstrate digital info presents in their documents. If an entrant tries to alter the data of the document that intruder cannot do this in QR-Code. During this Paper they use MASS Algorithm. Then the Data are entered within the QR-code later the QR-code is printed. Then the data can then be retrieved from the QR-code and decrypted using a decryption Method. And finally, it will be verified data that are already presented within the document.

One of the most practical researches done on this topic was conducted by Hitoshi Susono. 2014 where he used QR codes in class assessments to one class in July 2006, the problems he concluded after the experiment can be summed up as it follows:

- 1) The students had to pay for the cell phones which is not very expensive in Japan
- 2) Not all student had QR scanner and the student who had them where above 18.
- 3) Old phones had the issue of not being able to read the QR code because of camera focus, brightness and the size.
- 4) The displayed type on the mobile phone changes with their Contract Mobile Phone Brand

"The result was, the forty-three of the scholars answered "Yes". As a result of this study the students were asked if it was reasonable for them to use this method, shockingly 43% of them answered yes.

The students who answered yes supported their claim with follows:

- 1) It's higher to induce and browse a lot of comments from classmates in every class than within the additive form.
- 2) Easy access to the class material at any convenient time

The goal of the article they published was to explore the pros and cones of using QR codes in education in general plus to get a hands on experiment on how it would look like in real classrooms, the result they obtained were fairly positive and sparked creativity in both learners and educators [17].

49% of the students approved learning new things and 79% indicated somewhat agreement that they learned something new while 42% indicated they needed further

assistance with QR code and this due to the technical issues of the phones used in their experiment.

67% of elementary school and primary school student indicated that it was very simple to utilize QR-codes while 82% indicated that it was fairly easy to brows from the phone's screen.

95% of the first faculty one and first school two students pointed out that QR activities were a stimulating new approach to learn and ninety-eight of the scholars would really like to try to QR activities one more time. Some students had difficulties with their good phone and this is often why thirtieth of the scholars disagreed with the statement that the phone continued to work as they needed.

What's stunning is that thirty-seven of the scholars somewhat agreed that once using QR codes, the eye is drawn an excessive amount of on technology. This is often in all probability as a result of the activity itself wasn't well-planned and didn't inspire the scholars. Once the activity isn't well- planned, the eye of the scholars is also amused by something else. Additionally, technical issues might have had a negative impact on motivation.

In 2013, Yavuz TEKBAŞ presented the graduate thesis entitled "code production tools using an automatic calculation of derivatives and simplification mathematical expressions" [22]. In this work, A CFG is developed for syntactic and semantic structures of mathematical equations, JavaCC an automatic code generating tool was used to generate summary syntax tree (AST) as an object tree, and lastly evaluating object tree was handled to simplify and derive the expressions.

Baki Gokgoz In his study conducted in 2016 for his graduate thesis attempting to program numerical root finding method with simple approaches [22]. He approached with the help of automatic code generating tool while trying to describe root finding methods such as iterations expressions and functional translation. The way he did it is using processing of analysis operation of mathematical expressions which can be solved for roots using JavaCC tools.

3. GENERAL INFORMATION

3.1. QR-Code

When thinking about QR-codes in an instructional context or in education, it is necessary to see QR-code innovation as an enabler.

The center of attention should be more on education and beginners than on QR-code technology.

When discussing the application of QR codes in the educational system [16], our main focus should be on the learner more than the method of implanting QR codes itself, as technology does not always guaranteed improved results of the educational process, the more the educational sectors improves the higher our chances of implanting such systems to enhance the process. The main idea is to shift the focus of education on the learners themselves. Using the tools mentioned in this paper the teacher will be able to embed an exam inside a QR code then student can simply take picture of the code using their cellphones, the scan-able QR code looks as shown in Figure 1. The result we expect is to complete immersion of the leaner and higher security level for the teachers.



Figure 1. QR-code example

3.2 Cryptography

The phrase cryptography comes from the roots ‘crypto’ and ‘graphy’, roughly translating to “secret writing”.

If we want to make facts secret, we use a cipher – an algorithm, which converts plain textual content into cipher-text, which is gibberish except we have a key that lets we undo the cipher. The system of making text secret is referred to as encryption, and the reverse process to for getting the original text is known as decryption. The Ciphers have been used long before computer systems showed up. Julius Caesar used the technic which is called now a Caesar-cipher, to encrypt non-public correspondence. He was shifting the letters in a message forward via 3 places. So, A grew to become D, and the word "ktu" became this: "nwx". To decipher the message, recipients had to be aware of both the algorithm and the range to shift by, which acted as the key. The Caesar-cipher is one instance of a large classification of methods known as substitution ciphers. These substitute each letter in a message with something else in accordance to a translation. A massive drawback of basic substitution ciphers is that letter frequencies are preserved. For example, E is the most common letter in English, so if your cipher interprets E to an X, then X will exhibit up the most often in the cipher-text. An expert cryptanalyst can work backwards from these types of data to figure out the message.

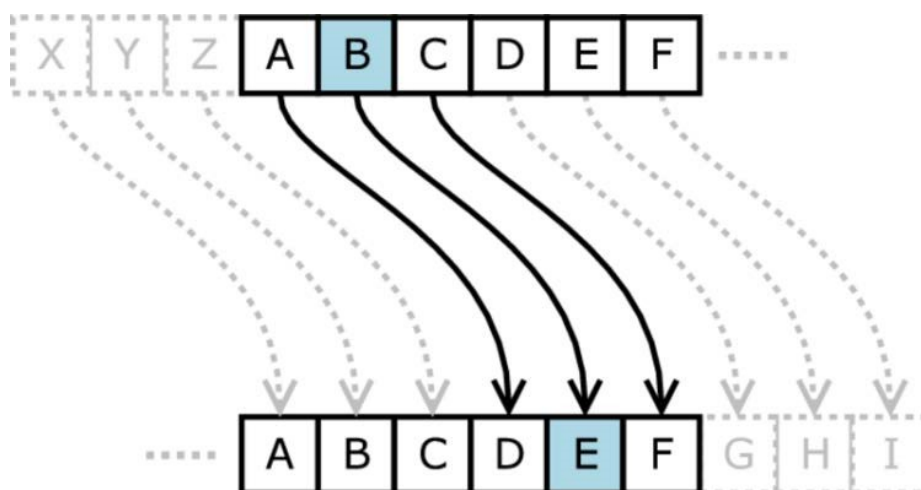


Figure 2. Caesar-cipher technique [23]

3.2.1. Encryption and Decryption Algorithm

To convert real data into what appears like garbage or to something meaningless cost this procedure is referred to as Encryption which means you are not allowing to someone to see it or locking the actual information into some other form. And the procedure of extracting actual statistics back from this meaningless textual content is called as Decryption

First, the information to be transmitted is called as plain text (or message) is fed to an Encryption system. The Encryption device makes use of a key to convert the plain textual content to encrypted form which appears like rubbish value, this is additionally called as cipher text. A corresponding key is used at the other give up to decrypt the cipher text returned to authentic message. When we say a "key" it genuinely potential a piece of string price which is fed to encryption and decryption algorithms alongside with the text for transformation. This is similar to locking your valuable things in a container and sending it across. At the different end the receiver will use the secret key to open the container and read the message you (Sender) sent.

If a hacker have been to faucet out the message being transmitted in the community he will get the encrypted message, say "1453" now he will not have the key to decrypt this message so this cipher textual content will now not suggest something to him. He might also try to use countless methods to ruin this code and get the hidden message out of this.

This artwork of trying to damage ciphers forms a distinctive department of learns about called as Cryptanalysis. However we will no longer go into that proper now. This encryption and Decryption together ensure security of the message being transmitted across the network. This total encryption and decryption approach is based totally on the premise that both sender and receiver share some unique keys which is not acknowledged with the aid of any outsiders, like the hackers. Depending on how the keys are shared.

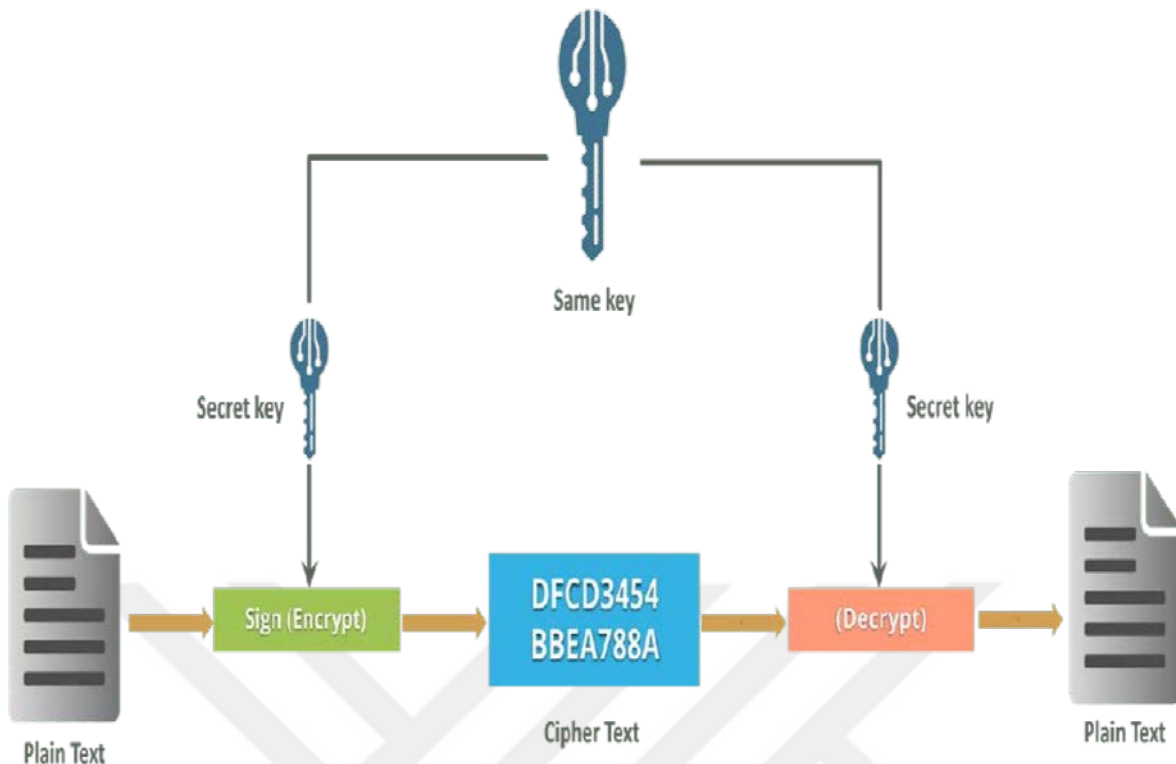


Figure 3. Symmetric Key Cryptography [24]

3.3 Formal Languages

Noam Chomsky was the first person to say that human brain is pre-wired with some basic rules of language. He says that when a child is born he's born in this world with a basic set of language skills, that means he's not a clean slate his mind is not empty. He already has a language acquisition device (LAD) there in his brain. Due to which he is able to learn language so quickly. Before Norm Chomsky a lot of psychoanalytical thinkers they believed that how a child learns a child learns by observing and then imitating parents and other people around him, Noam Chomsky was the first person who said that no the child already has basic rules of grammar there in his mind.

Formal languages mean that they will be mathematically defined, so in 1959 the linguist and philosopher gave a mathematical model of a grammar.

Formal way of representing of this Context-free grammar CFG is using this 4 variables

V – as a finite set of variables (non-terminals)

- T – as finite set of Variable
- S – Start Symbol
- P- we usually call it production of rule

3.3.1 Types of Grammars

We have generally 4 types of grammars first one is Type0 grammar, second one is Type1 grammar and the last but the least the third one is type3 .*Type0 (Unrestricted grammars) as the name suggest Unrestricted means this grammar has no restriction and known as most powerful in theory of computation.

This grammar has the production like this $\alpha \rightarrow \beta$ and in α we can have any combination of variable and terminals and β belong to any variable or terminals.

*Type1 grammar is context sensitive grammars which is derived from type0 grammar. In context sensitive grammar α can be equals of the length of β , also length of α can be less than equals to β .

But we should know that length of α is greater than β , this kind of grammar not allowed in Type1 context sensitive grammars but it is allowed in Type0.

*Type2 is known as Context free Grammar it derived from Type1. So if we put more restriction on typ1 (context sensitive grammars) we get type2

*Type3 known as Regular Grammar is delivered from type2 (Context free Grammar) It can be classified in two types Right Linear or Left Linear.

3.3.2 Parsing problems

3.3.2.1 Ambiguity

Ambiguous grammar, so what do we mean by ambiguous? A grammar is said to be ambiguous if there exists two or more derivation tree for a string Ω (that means two or more left derivation trees). So let's say that we have a grammar given and there is a string Ω that can be generated from this grammar and if this string Ω can be derived from two or more left derivation trees then that grammar is said to be ambiguous. So when we mean two or more derivation trees we should keep in mind that they should be both left derivation trees. It's not that you form one using a left derivation tree and another using a right derivation tree, and that is ambiguous because that is not the case. It is only when it can be formed using two or more left derivation trees then it is said to be an ambiguous grammar.

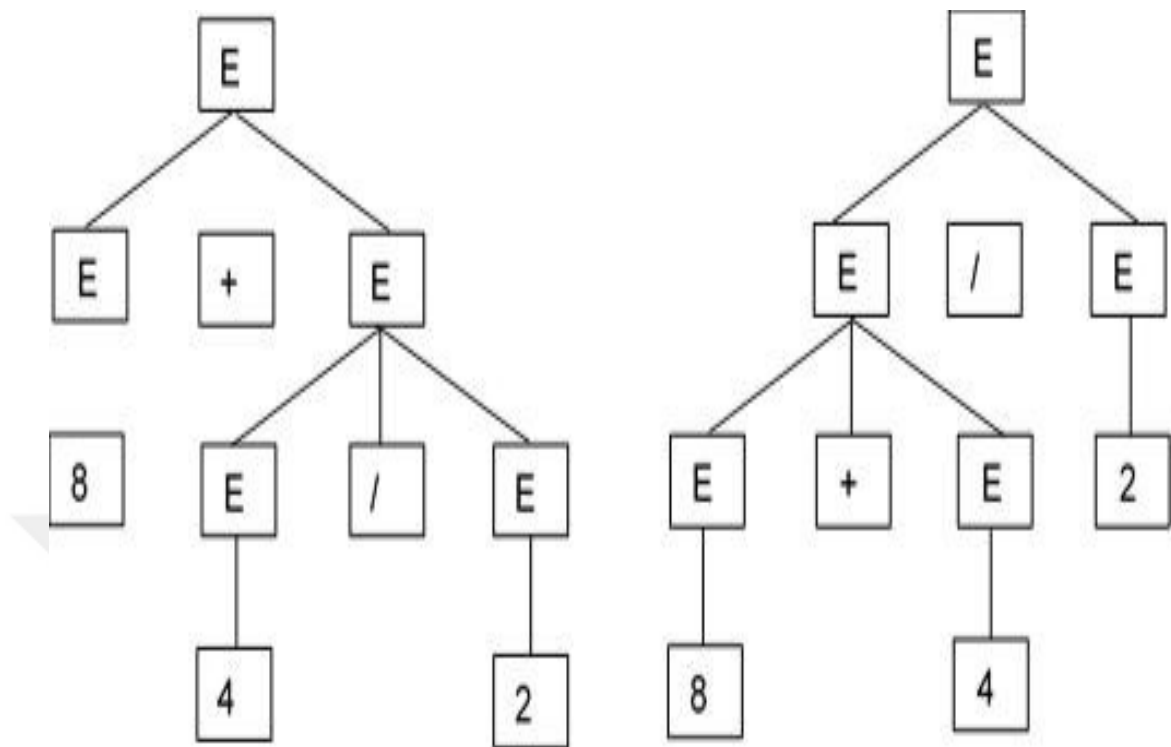


Figure 4. Example of Ambiguity

3.3.2.2. Recursive Productions

Productions are usually outlined in terms of themselves. For example an inventory of variables in an exceedingly artificial language can be indicated by this production:

$$V_L \rightarrow V \mid V_L, V$$

The higher than production is referred as algorithmic. If the algorithmic nonterminal is at the left of the right-side of the assembly, Example $B \rightarrow u \mid BA$, we tend to decision the assembly left recursive. Similarly, we will outline a right-recursive production: $B \rightarrow u \mid AB$

3.3.2.3. Left Factoring

Removing the common left issue that seems in 2 productions of the identical nonterminal is termed Left factorization, the method of factorization out the common prefix of alternates.

It's a helpful methodology for manipulating grammars into a type appropriate for algorithmic descent, it's done to avoid back-tracing by the computer program.

$A \rightarrow \alpha \beta \mid \alpha \gamma$ are 2 A-productions and $\alpha \neq \text{null}$. During this case, the computer program are going to be confused on that of the 2 productions to decide on and it'd should back-trace.

When left factorization the synchronic linguistics can become

$$B \rightarrow \alpha B' \mid B' \rightarrow \beta \mid \gamma$$

3.3.3. Parsing Techniques

On parsing we use two important function First and Follow to Construct parsing Table.

3.3.3.1. First and Follow

First (α) is the set of terminals that begin the string derived from α when α is any string of grammar Symbols.

If $\alpha \Rightarrow^* \epsilon$ then ϵ is also in First (α)

3.3.3.2. Follow Set

FOLLOW (A) for non-terminal A is the set of terminals that can appear immediately to the right of A in some sentential form.

Example: the set of terminals 'a' such that there exists a derivation of the form $S \Rightarrow^* \alpha A a \beta$ for some α and β

3.3.3.3. LR Parser

It is the foremost fashionable style of bottom-up parsing technique. Here, the "L" once more suggests that reading the input from left to right, whereas the "R" suggests that constructing the right derivation and its synchronic linguistics will describe a lot of languages than LL grammars.




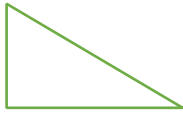
LR program can handle an outsized category of context-free grammars and may sight the syntax errors as before long as they'll occur. A lot of data regarding LL and LR parsing algorithms is documented.



3.4. Triangles and their Types

A triangle is a simple closed polygon which is made up of three line segments, a triangle has three sides three boat Isis and three angles a triangle is denoted by the Greek letter Delta the Sum of all angles in a Triangle is 180 degree. This is known as the angles sum property.

Triangle ABC can be written as triangle ABC triangle BCA or triangle C A B.

Table 1. All Triangles Types.

Equilateral		All the sides are same length(shown by the line through Each of them) and all the angles are the same length $\angle A = \angle B = \angle C = 60^\circ$
Isosceles		Two Side are equal (Shown by the lines) Angles opposite the equal side are equal. $\angle A = \angle B$
Scalene		All three angles and all three sides are difference $\angle A \neq \angle B \neq \angle C$
Right		Has one angle equal to 90°

Acute		All three angles are less than 90°
Obtuse		Has one angle greater than 90°

3.5. Language Processor

Language processor may be a special style of a laptop code program designed or won't to perform tasks and has the capability of translating the ASCII text file or program codes into machine codes. There are differing types of language processors like compilers, Assemblers, interpreters, preprocessors, and disassemblers. During this section, we are going to make a case for the foremost wide used language processors that are compilers and Interpreters.

3.5.1 Compilers

A Compiler is a program that reads a program written in one language and translates it into an equivalent program in an others language.

Input language: Source language

Output language: Target Language

A compiler also reports errors present in the source program as a part of its translation process.



So we can say that a Compiler is a computer program that transform a Source code program written in a High level language to a Target code called machine language of a computer

There is some of languages which are generally compiled are:

Java

C++

C

3.5.2 Interpreters.

An interpreter is a computer program which directly executes command written in a programming language or scripting language.

Performs line by line execution of the source code which written in high level language.

Interpreter reads source code line by line, converts it into machine understandable from, executes the line, and then proceeds with the next line.

Unlike compiler, it does not convert the high level language into machine code.

To do this, it uses one of three techniques.

- It either Analyses the code directly to perform the execution,
- Or it translates the source code into some other intermediate code and then executes it.
- Or it explicitly executes stored precompiled code made by a compiler which is part of the interpreter's system

Here are some languages which are generally interpreted are:

PHP

Perl

JavaScript

Python

3.5.3 Variations between Compilers and Interpreters

Both compilers and interpreters are translated the problem-oriented language into machine language, however there are several variations between them.

The distinction between Associate in Nursing interpreter and a compiler is as follows:

An interpreter reads one statement and translate it, when death penalty that statement it takes another statement in sequence. Whereas the compiler reads the entire program and interprets it in one go so executes it.

A compiler generates the error message solely when the scanning of the entire program. Since Associate in Nursing interpreter continues translating the program till the primary error is met, and to interpret the following statement we've to mend the error.

A compiler generates intermediate code that desires a lot of memory, and it will be generated each time when the program is being compiled. As Associate in nursing interpreter no intermediate code is generated, it directly generates code.

In analyzing and process the ASCII text file a compiler takes larger quantity of your time relatively and interpreter analyzes and processes the source code directly.

Besides the process and analyzing time, programs made by compilers run a lot of quicker than the identical programs dead by Associate in Nursing interpreter.

3.5.3.1 Compiler Phases

The basic compiler steps are displayed in Figure 5.

The two main components of the compiler are: the *front-end* and the *back-end*

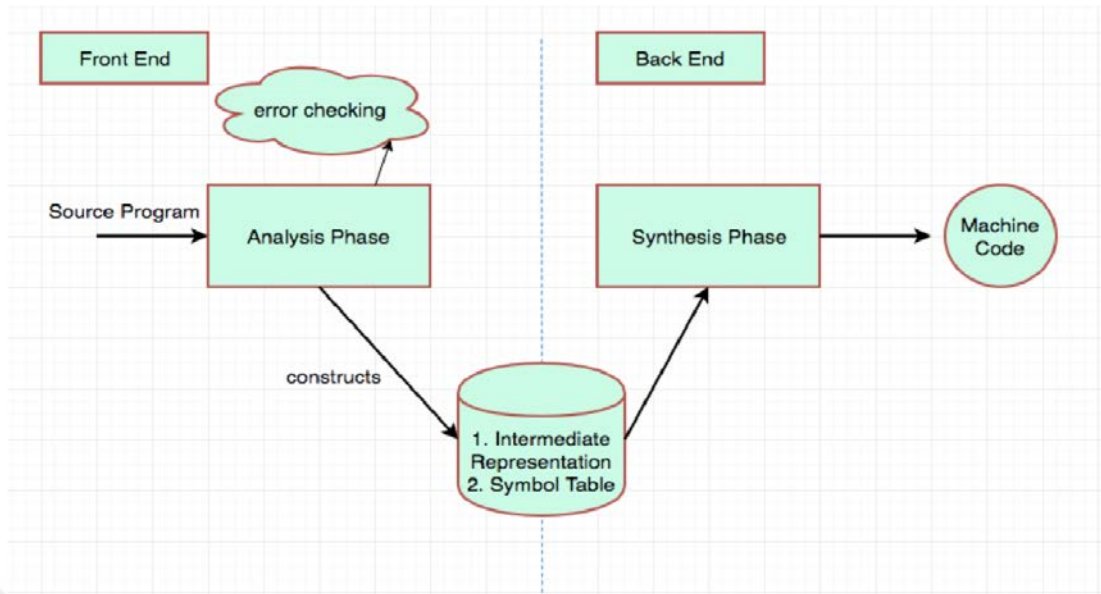


Figure 5. Structure of a Compiler [25].

3.5.3.1.1 Front-End

This is the structure of the Front End

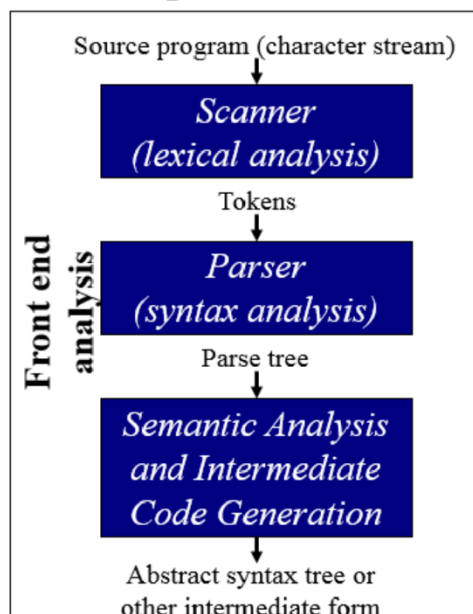


Figure 6. Frond-End Analysis [26].

3.5.3.1.2 Back-End

The second a part of the compiler is that the back-end part that enthusiastic about the target machine. The back-end part embody code improvement phase, the necessary error handling, image table operations, and therefore the final code generation. This part of compiler is freelance of program.

The main task of the front-end part is to investigate the supply knowledge and generate the thing tree representation whereas the rear finish synthesizes the computer program from the object tree (intermediate code).

The generation of associate degree intermediate code is also referred as middle finish, because it depends upon program and target machine.

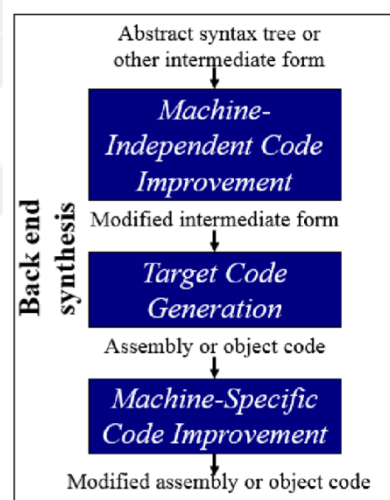


Figure 7. Back-End Synthesis [26].

3.5.3.2 Language Interpreters

In any language interpreter translating any program from one language to a different, initial the compiler breaks the supply information to know the means and also the grammar structure of the program, then it recombines during a completely different and meaningful manner. The compiler performs two main tasks; analysis at the face, and will synthesis the rear finish. The analysis is sometimes variable into: Lexical analysis, Syntax analysis and linguistics analysis.

3.6. Lexical Analysis (Scanning)

Lexical analysis is the first phase of compiler. It works closely with the syntax instrument that reads input characters from the program and teams them into lexemes to supply output as a sequence of tokens which may be handled more easily by a parser, by eliminates comments and white spaces in the form of blanks, tabs and newline characters.

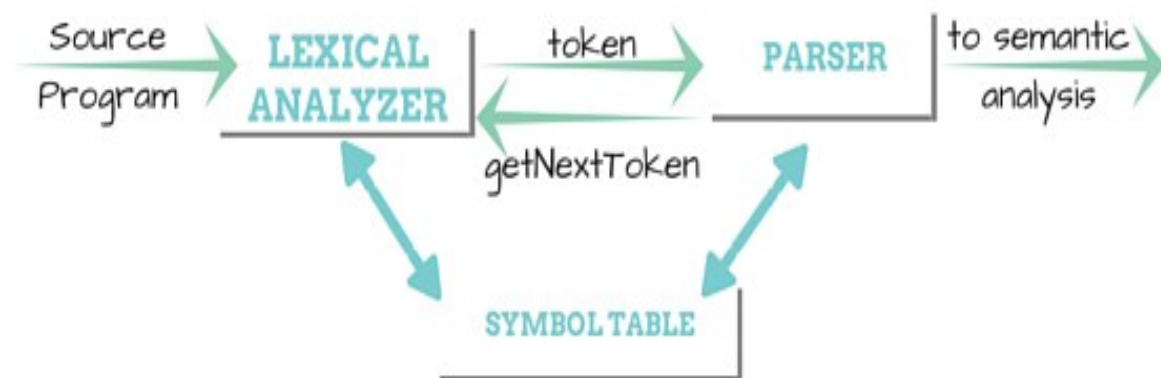


Figure 8. Lexical-Analyzer to Parser [27]

3.6.1 Interaction of lexical analysis with parser

Lexical analyzer produces the token and passes it to parser, upon receiving a GetNextToToken command from the parser.

The lexical analyzer reads the input character until it can identify the next token.

SymbolTable module interacts with all the phases of compiler.

What is tokens and lexemes?

Tokens is a sequence of character that can be treated as a single logical unit. Tokens can be identifier, keywords, operators, special symbols and constant

What is lexemes?

Lexeme is a sequence of character in a source program that can be matched by a pattern for token.

What is tokenization?

The process of forming tokens from the input stream of characters is called tokenization.

Ex: Time= 8*4;

Table 2. Tokens and Lexemes

Lexeme	Token
Time	Identifier
=	Assignment Symbol
8	Num
*	Multiplication operator
4	Num
;	End of statement

What is pattern?

A rule that describe set of strings associated to tokens.

EX: L(L/D)*

L is considered as a letter and D as a digit, so letter by digit star is a pattern to symbolize the set of strings which consists of letter followed by letter or digit.

Patterns are specified using regular expressions.

Ex: Table of Tokens, Lexemes and Patterns

Table 3. Tokens, Lexemes and Patterns

Token	Sample Lexemes	informal Description of Pattern
if	If	If
while	while	while
Relation	>, <, =, >=, <=, <	> OR < OR => OR = OR >= OR <
id	Sun, j, count, K4	Letter followed by letter and digits
Num	0, 123, 5, 2019, 8.61E45	Any numeric constant

What is attribute for token?

When more than one pattern matches a lexeme, lexical analyzer must provide additional information about particular lexeme.

That is matched to the subsequent face of compiler

Two Lexeme: 0,1;

Example pattern: num matches both 0 and 1

But it is essential for the code generator to know what strings are actually matched.

Ex:19

<num 19>

in integer 19 constants are constructed by converting number to token num, and passing the attribute number as its attribute.

3.6.2. Issues in Lexical Analyzer

There are several reasons for separating the lexical analysis from syntax analysis the reasons are compiler efficiency is improved and compiler portability is enhanced. Compiler efficiency is improved.

Compiler portably is enhanced.

3.6.3. Regular Expressions

For lexical analysis, definitions are written associate degree expressed victimization regular expressions that is an algebraically notation designed to explain sets of strings. Regular expressions are a useful gizmo designed for describing, matching and extracting patterns in text. Regular descriptive linguistics is understood because the grammar outlined by regular expressions and therefore the language defined by regular descriptive linguistics is understood as regular language. Additional reading for normal expressions is documented by Mogensen, and Torben Ægidius book.

The lexical analyzer must scan and acknowledge solely a finite set of valid strings/tokens/lexemes that belong to the predefined language. It searches for the pattern outlined by the language rules.

Every computer user ought to have the data of implementing a tool for matching regular expressions from scratch.

3.7. Syntax Analysis (Parsing)

Syntax analysis or parsing is the second phase of a compiler.

After lexical analysis splits the input into tokens, the goal of parsing is to recombine these tokens not into a list of characters, but into something that has meaning and reflects the structure of the text.

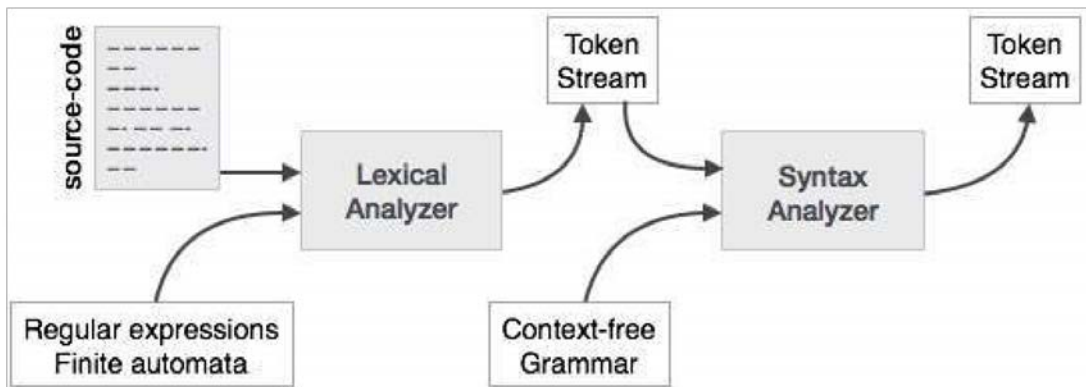


Figure 9. Parsing Phase [28]

3.8. Java CC

JavaCC is a compiler generator similar to Yacc that accepts language in BNF as formatted as input. The generated computer program contains the main elements of corresponding compiler of the desired language, which has a lexical instrument and a syntax-analyzer. Fig a pair of shows the general the structure of a computer program generated by JavaCC

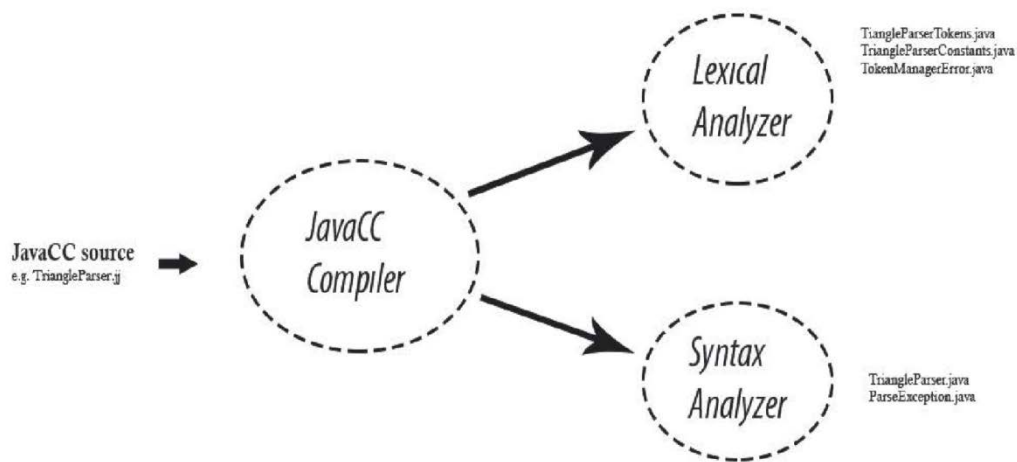


Figure 10. JavaCC Compiler

3.9. Mobile Apps

What is an app? Anyway this is a simple question that millions of people ask daily. It turns out that app is short for application, it used to be the application alluded to the sort of software that was provided to a user's need during a business issue, then application got abbreviated to app when software applications hit cell phones, and app is simply faster.

Where do people download your app? Once it has been built up, the two most famous spots to go for app are androids Google Play and Apple's App Store. These stores hold the file representing the application holding back to be downloaded and installed onto their gadgets. Let's take an example real estate investors, they are spending an increasing amount of their free time looking for good deals using various apps on a daily basis. These apps become more and more sophisticated, and faster bringing properties that fit their needs. Consequently they are inclined to connect more often with the app by habit to see what the next treasure is they will find.

But back to you in your dreams it's simple in concept you keep providing the value of a working app on a daily basis as long as your user subscribes to using it. This leads to a

model of doing business that is subscription-based instead of requiring a larger one-time payment. This model is typically known as SAS or software as a service.

Whether you're a small business and are looking to add rocket fuel to your revenue picture, you are a part of a large company looking to streamline services to customers and employees or maybe you just have a great idea for changing the world with a startup. No matter who you are and what your purpose is? One thing is unavoidable the app must be coded first, so it can be delivered to the people you want to use it. Let's talk about what that entails. Code is just a set of instructions that tell the app and phone or other device what to do, and coders are the software developers who write these instructions. So if you want your logo to appear in an app, a coder needs to put it there, if you want a menu that gives your users choices to navigate through your app or just have people flip through your screens. Coders have to put those choices screens, and menu options there. Fortunately methods of building apps have been refined over the years and there's a community of developers that can pretty easily just code up whatever it is you need make it look individual to your brand and infinitely appealing to your users. Coders they write their own code from scratch to craft to shape exactly what an app does on each type of smartphone, but most likely this often met their custom code with code from other coders. These access points are called APIs short for application program interfaces and are bundled together in software development kits or SDKs. to get more coders to bring value and users to their product or service existing resources want to make it easy on coders that want to connect to their product or service. Therefore they offer these bundles of instructions APIs and SDKs that are pre-written to a degree so other coders can use them again and again and get routine things done inside their app. in another example of using API what if your users wanted to know instantly that they have a message waiting in your app? You can enable something called push notifications that notify your users that a message is and then take them directly to the message in your app with simply a tap. You need to re-engage customers after a slow week of business? You can send a push notification out offering a discount directing the customers to purchase the item through your app, and not even bother coming into your physical location. Just have an amazing app idea, you can access just about anything from your mobile device and combine it with what other people have produced and any number of unique ways.

4. METHODOLOGY

This chapter present the methodological aspects of the work described in this thesis.

4.1. The Relevant Tools and Technologies

In this thesis we used many tools and technology, which are generally used by java technology and in the development area in general.

These Tools and technology are:

Java Compiler Compiler (JavaCC)

1. Android Studio for the application mobile that we had built
2. The Zxing is an abbreviation of Zebra Crossing library. This library developed by Google to allow us to generate and scan the QR-Code Technology.
3. AES algorithm, which stands for Advanced Encryption Standard or System. It is a Symmetric encryption we have used it to secure our sensitive data.

4.2. Solving Triangular Problems

After a teacher enters a geometry problem related to triangles, the program draws the relevant triangle and tries to determine some side lengths and angles of the triangle. If the program receives the required input for sides or angles of a triangle, then the rest is calculated by the program. Solving angles and sides:

In order to solve for the third angle of a triangle whose two angles are given:

$$\angle C = 180^\circ - \angle A - \angle B$$

In order to solve for an angle of a triangle whose three sides are given:

$$\angle A = \cos^{-1}\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$

In order to solve for a side, given 2 sides and the angle in between:

$$a^2 = b^2 + c^2 - 2bc \cos \angle A$$

In order to solve for an angle, given its opposite side, another angle and the opposite side to that angle:

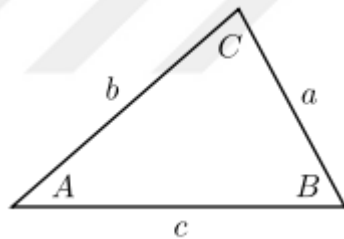
$$\angle B = \sin^{-1}\left(\frac{b \sin \angle A}{a}\right)$$

In order to solve for a side, given its opposite angle, another side and the opposite angle to that side:

$$b = \left(\frac{a \sin \angle B}{\sin \angle A}\right)$$

4.2.1. Prototypes and Mock-Ups

In order to create a triangle to match the Teacher specifications, several combinations of information can be entered.



For all cases we use the following triangle as a visual:

In order to solve the triangle to the Teacher specifications, the Teacher needs to enter some information about the triangle. There are several combinations of fields the Teacher can enter. The following are the different combinations:

1. SSS

- Inputs:

- a, b, c
- b, c, a
- c, a, b
- c, b, a

- b, a, c
- a, c, b

2. SAS

- Inputs:
 - a, B, c
 - b, A, c
 - a, C, b
 - etc.

3. ASA

- Inputs:
 - A, b, C
 - B, a, C
 - A, c, B
 - etc.

4. ASS/SSA

- Inputs:
 - a, b, A
 - b, a, B
 - b, c, B
 - c, b, C
 - a, c, A
 - c, a, C
 - etc.

5. AAS/SAA

- Inputs:
 - A, B, a
 - B, A, b
 - B, C, b

- C, B, c
- A, C, a
- C, A, c
- etc.

4.2.2. Error/Restart Program:

As soon as the Teacher enters incorrect data, an error message will be outputted according to the type of error, there are three cases.

Case 1 (Negative Side Length)

Case 2 (Angle Out of Range)

Case 3 (The Sum of Two Side Lengths is Less Than or equal to the length of the Third Side)

Case 4 (Ambiguous Case of Sine Law where there are no solutions)

- $a < b \sin A$

Case 5 (The Sum of Three Angles are greater than 180°)

For all above cases, the program will output:

A triangle cannot be constructed based on these specifications.

Reinitializing...Complete

4.3. Analysis Phase

4.3.1. Lexical Analysis

We know that the first phase in the process of a compiler of our program will be LA, which is lexical analysis. Let us consider an analogy to better understand the tasks involved in the lexical analysis phase of our project.

For instance: If a student x wants to learn Turkish language or any other language, he will start learning from the alphabets then he will learn to write words combining the alphabets, once he is capable of writing whole words he will be eager to know the meaning of those words, So to know the meaning he will revert to the dictionary, where the predefined words are already explained with its meaning.

The process of our compilation also works in the similar way performing tokens from individual characters and referring to the regular expressions that can be compared to a dictionary.

When the source-code enters the lexical phase, the lexical analyzer or the scanner reads the text character by character.

The main task of lexical analyzer (Scanner) is to convert Lexemes in the tokens.

Table 4. Part of Check Method

```

public static boolean Check(Double a, Double b, Double c, Double A, Double B, Double C) {

    boolean temp = true;

    int angleCount=3, int sideCount=3;

    double h;

    double s1, s2, angle;

    if (a==null) {sideCount--;} if
(b==null) {sideCount--;}

    if (c==null) {sideCount--;} if
(A==null) {angleCount--;} if
(B==null) {angleCount--;} if
(C==null) {angleCount--;} if
(angleCount >= 2) {

        if (A==null) {angleCount--;A = 0.0;}
if (B==null) {angleCount--;B = 0.0;} if
(C==null) {angleCount--;C = 0.0;}

        //no 2 obtuse, no 2 right angles, and sum of 2 angles less than 180

        if (A + B + C >= 180) {

            temp = false;

            }

        }

        else if (sideCount == 2) {

            if (a!=null && b!=null && A!=null) { //abA
s1 = a;      s2 = b;

            angle = A;

            }
}
}

```

In this part of our source code in the third line (`int angleCount=3, sideCount=3;`) the word `int`, `angleCount` and `sideCount` are denoted as lexemes similarly `comma`, `3`, and `equal` to are also like lexemes.

The lexical analyzer replaces the lexemes with tokens. For example `int` is a token similarly `angleCount`, `sideCount`, `=` (equals), `,` (comma) and `3` are also tokens.

In the process of converting lexemes into tokens, first LA has to identify the possible tokens in the source code. For this purpose it introduces the regular expressions or RE.

Regular expressions are the notations for describing a set of character strings.

If the lexical analyzer finds any invalid Tokens, it generates an error message by representing the line number associated with the error.

The program gets read line by line only in the lexical phase. It also performs secondary tasks such as removing the comment lines and extra white spaces in the source code. At the end of this program we can see only the tokens, which are the output of this phase.

Table 5. Convert the code to Tokens

Part of the program	At the end of Lexical phase
<code>int angleCount=3, int sideCount=3;</code>	<code><int><id,1><op,=><const,3><,><int><id,2><op,=><const,3></code>

4.3.1.1. Interaction of LA with the Parser

Next the tokens that are produced as the output are used by the parser to generate the syntax tree, which is the next phase of the compiler.

Lexical analyzer sends the tokens to the syntax analyzer whenever it demands, upon receiving a request from the parser, the lexical analyzer reads the character string until it recognizes the next token, then if the lexical analyzer finds any token it responds to the parser representation. If the token is a parentheses, comma or colon then it is represented as an integer code.

4.3.1.2. Lexeme, Token and Pattern

We know that the lexeme is a stream of characters in the source code. Data are matched by the pattern for a token. For every lexeme, there is a predefined rule called patterns, which identifies if the token is valid or not. These rules are described by the grammar rules in pattern. A pattern has a set of predefined rules, which contain a list of valid tokens. These patterns are defined by means of regular expressions the lexemes which are a series of atomic units that can be split further are categorized into blocks called tokens.

The typical tokens are identifiers, keywords, operators, special symbols and constants.

Table 6. The typical tokens.

identifiers	keywords	operators	special symbols	constants
A,b,c,d,f,k,d	double, main, print Boolean, int, String	+, =, *, -, /	%, (()), {}, ;, <>	8, 78.5

Table 7. The declaration of our Tokens

TOKEN:{	
<COMM: ", ">	<TRIANG: "Tr">
<ASSIG: "=">	<TEXT: "Tex">
<DIV: "/">	<LPARENT: "(">
<TIM: "*">	<RPARENT: ")">
<PLU: "+">	<NUMB: (["0" - "9"] +)>
<MINS: "-">	<SEGM: ["a" - "z"] ["a" - "z"]>
<UNDERL: "_">	<ANGL: ["a" - "z"] ["a" - "z"] ["a" - "z"]>

4.3.2. Syntax Analysis

The next phase of the compiler after lexical analysis is the syntax analyzer, also known as parsing. It takes the output from the lexical analysis, that mean it takes the tokens as an input and generates a parse tree or syntax tree. Parse tree is a hierarchical structure, which represents the semantic structure of a string. Also it check for the source code grammar in token arrangements, scope of a variable and array bound exception.

In the Table 5 and Table 6 respectively shown our grammar roles and the Syntax classes.

Table 8. Our Grammar

```

SS -> M ; SS | T

M -> numb : ( E | D | F | tr "(" A ")" )

E -> ( K | A ) "=" numb

D -> P "<-" K

...

T -> text "(" numb ( , numb ) ? ")"

```

Table 9. Our Syntax source code

```

class EqualsList extends Exp {

Exp a,b;

abstract class Exp {

Object t = null;

public abstract Object accept(Visitor v);

}

class NumList extends Exp {

public Exp a,b;

public NumList(Exp a,Exp b) {
this.a=a;

this.b=b;

}

public Object accept(Visitor v) {

```

```
return v.visit(this);  
  
}}  
  
class Num extends Exp {  
public double n;  
  
public Num(double x) {  
  
n = x; }  
  
public Object accept(Visitor v) {  
return v.visit(this); }.....
```

4.3.3 Semantic analyzer

The input for semantic analyzer is the parse tree. This phase checks whether the parse tree is constructed by following the rules of the language, for example it checks the value assigned between the compatible data types. This analyzer keeps track of identifiers their types and expressions, it also checks whether the identifiers are declared before use. The output of this phase will be an annotated parse tree, annotation refers to the addition of attributes and roles to the syntax tree.

Table 10. Parser fille.jj

```

Exp T() :{
Exp a;Token t1,t2;}
{
<TEXT><EQUALS><LCOTS>t1=<NUM>
{ a = new Num(Integer.parseInt(t1.image));
}
(<COMMA>t2=<NUM>
{ a= new NumList(a, new
Num(Integer.parseInt(t2.image))); })*<RCOTS>
{ return a; }}

```

4.4. Synthesis Phase

4.4.1. Intermediate Code Generation

After semantic analysis this phase generates an intermediate code of the source code, which makes it easier to be translated into the target machine code. This phase acts as a bridge between the analysis phase and the synthesis phase. The final machine language code is produced in this stage.

4.4.2. Code Optimization

Code optimizer takes intermediate code as input from the previous phase this phase performs the code optimization for the intermediate code it removes unnecessary temporary

variables generated in the previous phase. Compiler takes less space and avoids wastage of resources such as CPU and memory.

4.4.3. Code Generation

The final phase of a compiler is the code generation phase. The optimized output from the previous phase is given as the input for this space. It translates the intermediate code into a relocatable machine code. The length of the machine language program is reduced here. The output of the code generation phase is the machine language program.

4.5. A Comparison of Some Encryption Algorithms

We take three of encryption algorithms in order to make a comparative study. The results are given in Table 11.

These three encryption algorithms are AES, DES and 3DES. We present them in 9 factors, which are key length, cipher type, block size, developed, cryptanalysis resistance, security, possibility key, possible ACSII printable character keys, time required to check all possible key at 50 billion second, these eligible proved the AES is better than DES and 3DES.

Table 11. A Comparison of AES, DES and 3DES Algorithms

Factors	AES	3 DES	DES
Key Lenght	128,192, Or 256 Bits	(k1,k2 and k3) 168 bits, (k1 and k2 is same)112 bits	56 bits
Cipher Type	Symmetric Blok Cipher	Symmetric Blok Cipher	Symmetric Blok Cipher
Block Size	128, 192, or 256 bits	64 bits	64 bits
Developed	2000	1978	1977
Cryptanalysis Resistance	Strong against differential truncated differential, Linear, Interpolation and sguare attacks	Vulnerable to differential, Brute Force attacker could be analyze plaint text using differential cryptanalysis	Vulnerable to differential and linear cryptanalysis; weak subsitution tables

Security	Considered secure	One only weak which is Exit in DES.	Proven Inadequate
Possible Keys	2^{128} , 2^{192} , or 2^{256}	2^{112} or 2^{168}	2^{56}
Possible ASCII printable character keys	95^{16} , 95^{24} , or 95^{32}	95^{14} or 95^{21}	95^7
Time required to check all possible keys at 50 billion keys per second**	For a 128-bit key: 5×10^{21} years	For a 112-bit key: 800 Days	For a 56-bit key: 400 Days



5. STEP-BY-STEP THE ILLUSTRATION OF THE OUR APPLICATION

5.1. The Input Data

5.1.1. Interpreter Phase

The following sample input in the Table 12 will be parsed and should respect our grammar rules otherwise it generates an error message. This error message is handled by our program and ask the user to try again to enter a valid data.

Table 12. A Sample Input

```
1:t(abc);
2:ab=20;
3:ke j bc;
4:e<-bc;
5:abc=45;
6:acb=90;
7:cab=?;
8:kc=?;
9:Msg("What kind of triangle is shown?");
Text={ 1,3,4,6,7,8,9}
```

The parser file called TriangleParser, to generate our parser for the first time we have to run javaCC followed by our jj file, which is ParserTriangleQr.jj that content our

syntax description and grammar rules. Then java files are created by the parser. One of our class is TriangleParserTokenManager.

This class TriangleParserTokenManager contains the static method getNextToken(). Every call to getNextToken() returns the next token in the input stream. When getNextToken is called, a regular expression is found that matches the next characters in the input stream in our grammar the input start with T(Name), End withText {Num,Num..} in which telling us which Num of the data we have to display for the students and in-between consist of the other data like the angles of the triangle $abc = 90^\circ$, the sides like $ab = 40$, $k \in ab$, $ak = ?...$

In addition in our parser basis of our grammar there is a public method declaration for each non-terminal and this method return an object of type Exp.

5.1.2 Encryption Phase

After Compiler Phase all input data are add in ArrayList object then returned by the Parser.

This data are later encrypted by our AES algorithm

Table 13. The AES algorithm

The Encryption method	<pre> public String Encryption_QR(Stribg inputData) throw Exception{ Key our_key=genertekey(); Cipher ci = Cipher.getInstance(ALGO_QR); ci.init(Cipher.ENCRYPT_MODE, our_key); byte[] encrypted_val=ci.doFinal(inputData.getBetes()); String Encry_value= BASE64Encoder().encode(encrypted_val); Return Encry_value; </pre>
-----------------------	--

The Decryption method	<pre> public String Decryption_QR(String inputData1) throw Exception{ Key our_key1=generatekey(); Cipher ci1 = Cipher.getInstance(ALGO_QR); ci.init(Cipher.DECRYPT_MODE, our_key1); byte[] decrypt_val= new ASE64Decoder().decodeBuffer(encryptedData); byte[] decrypt_value= new ci1.doFinal(decrypt_val); String decrypted_value = new String(decrypt_value); Return decrypt_value; </pre>
This method generate the secret key	<pre> Private key generatekey() throws Exception{ Key key1 =new SecretKeySpec(keyValue, ALGO_QR) Return key1; </pre>

5.1.3 QRcode Phase

5.1.3.1 Generating QRcode

After the Encryption phase the QRcode time comes, and the data encrypted stocked in a qrcode. The QRcode generated by our system cannot be read by another QRcode system because the information will be encrypted by the AES algorithm as it shows in the table below. So later Teacher can print this QRcode on paper and give it to the students.



Figure 11. Qrcode Scanned

5.1.3.2 Scanning Qrcode

In the scanning phase the decryption method of AES algorithm is called and using the same key of decryption, which can be found in the end of QRcode data.

Another algorithm is called to draw the Triangle on the phone and display to the student on his phone.

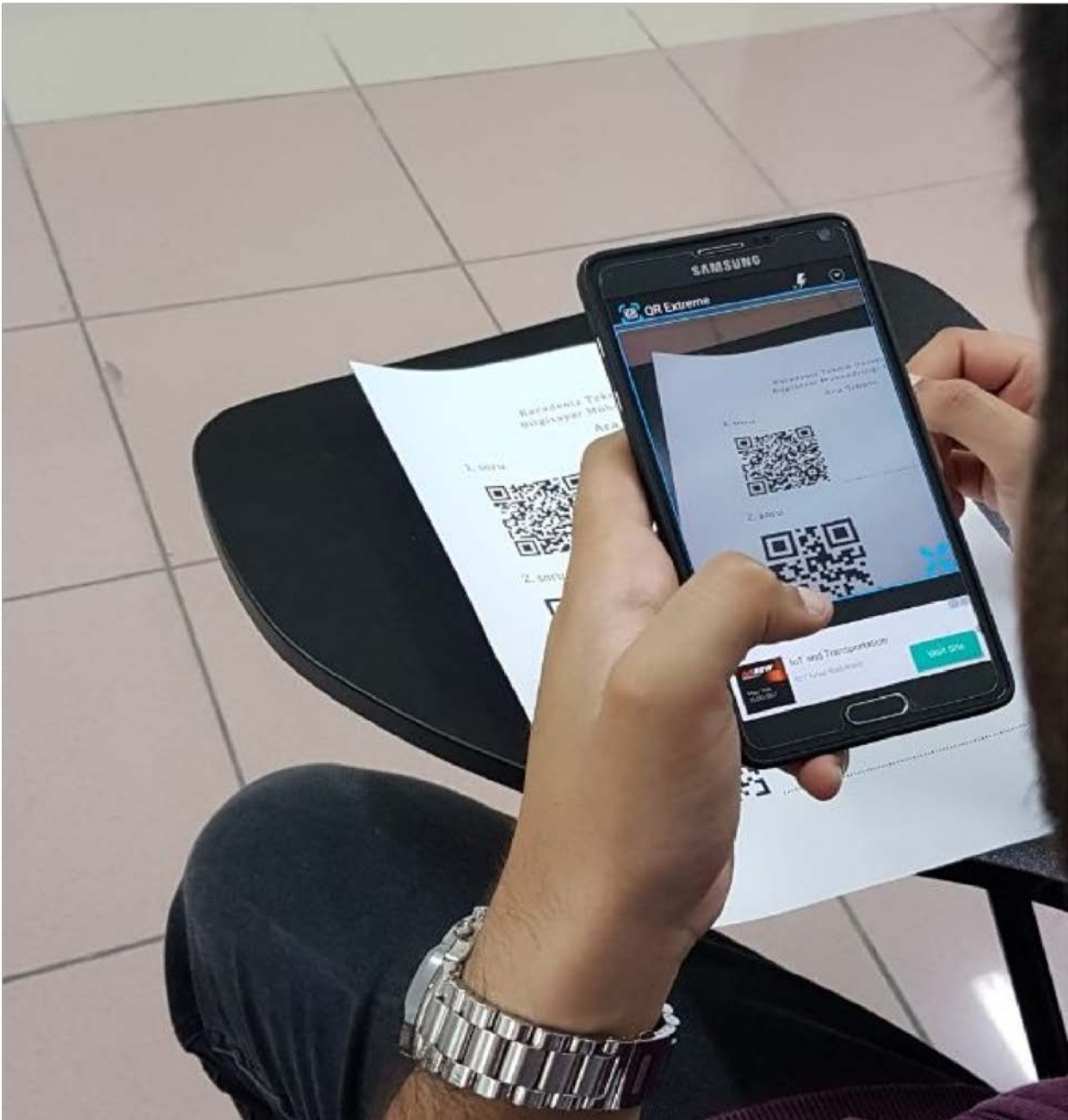


Figure 12. The Interface Student Scanning Phase.

KTU

$|BC|=20$ $|AC|=20$ $|BE|=7$ $\angle BAC=45^\circ$

Consider the giving diagram. find the values of the following . $\angle ABC=?^\circ$, $\angle BCA=?^\circ$
 $|AB|=?$, $|AE|=?$
Find the area of the triangle ECA.
Find the perimeter of the triangle ABC.

The diagram shows a triangle ABC with vertices A, B, and C. Point E is located on the side BC. The length of side AC is 40.0, and the length of side BC is 20.0. The segment BE is labeled 7. The angle at vertex A is 45 degrees.

Figure 13. The Interface After the QRcode Scan.

6. CONCLUSION

In this study, we implement a QR code based encryption and decryption system to help both the educators and learners, it helps the educators by saving time as they provide all the necessary information of problem description and then the program automatically encrypts it, which is a more innovative way than the classical paper-based exam which could consume more time and energy. Even though the studies in the literature focuses on the security or the triangle problems individually, we conduct a study that mixes both of them, in order to present a better solution which facilitates the education process. When it comes to security, which is a major important issue nowadays, the application maximizes the security. Through other applications used by the educator, no other QR code scanner will be able to decrypt the AES algorithm, which maximizes the security and terminates any opportunity of hacking the exam questions. Finally it reduces the paper work pressure on the educator by reducing the printed paper.

7. FUTURE WORK

In our work the input data is typed by teacher in the program, so to make the work easier for teacher we will implement the OCR technology, which stand for Optical Character Recognition. The application will be able to read a handwriting, so the teacher will be able to write the exam on a paper by pen or pencil. Our program now is solving only the missed part which is need to draw the triangle, our next work will focus on solving the given problem to the students that need to solve for, but the solution will be hidden from the student only the teacher can see this solution, and it will help him to give a great for the students in a short time, additionally we will make the system be to be compatible others mobile operating system like ios (IPhone), and others, and make it also available on the network.



8. REFERENCES

1. www.useoftechnology.com/how-has-technology-changed-education/ How Technology has Changed Education. 22 March 2017.
2. Traxler, J., 2009. Current State of Mobile Learning. In Ally, Mohamed (Ed.).
3. Ally, M., 2009. Transforming the Delivery of Education and Training (pp. 9–24). Athabasca University Press, Edmonton, Canada.
4. Chen, N., & Lee, C., 2010. Augmenting Paper-Based Reading Activities with Mobile Technology to Enhance Reading Comprehension. Taiwan, pp. 201-203.
5. Crompton, H., 2013. A Historical Overview of M-Learning: Toward Learner Centered Education.
6. Rikala, J., & Kankaanranta, M., 2012. The Use of Quick Response Codes in the Classroom.
7. Traxler, J., 2009. Current State of Mobile Learning. In Ally, Mohamed (Ed.), Mobile Learning: Transforming the Delivery of Education and Training (pp. 9–24).
8. Law, C., and So, S., 2010. QR Codes in Education, 3(1), pp. 85-100.
9. Rikala, J., & Kankaanranta, M., 2012. The Use of Quick Response Codes in the Classroom. 11th Conference on Mobile and Contextual Learning. Helsinki, Finland, pp.148-155.
10. Rikala, J., & Kankaanranta, M., 2014. Blending Classroom Teaching and Learning with Qr Codes.
11. Osawa, N., & Noda, K., 2007. System with Location Awareness Using RFID and Symbology Tags.

12. De Pietro, O., & Fronter, G., 2012. Mobile Tutoring for Situated Learning and Collaborative Learning in AIML Application Using QR-Code. pp. 799-805.
13. Ozcelik, E. & Acarturk, C., 2011. Reducing the Spatial Distance between Printed and Online Information Sources by means of Mobile Technology Enhances Learning: Using 2D Barcodes.
14. www.study.com/academy/lesson/what-are-the-disciplines-of-mathematics.html
Disciplines of Mathematics 4-8 (114).
15. Shiobaru, D. & Naomi, F., 2006. Collecting Students' Degree of Comprehension with Mobile Phones.
16. Law, C., and So, S., 2010. QR codes in education. *Journal of Educational Technology Development and Exchange*, 3(1), pp. 85-100.
17. Rikala, J., and Kankaanranta, M., 2012. The Use of Quick Response Codes in the Classroom. 11th Conference on Mobile and Contextual Learning. Helsinki, Finland, pp.148-155.
18. Osawa, N. et al., 2007. Outdoor Education Support System with Location Awareness Using RFID and Symbology Tags. *Journal of Educational Multimedia and Hypermedia*, 16(4), pp. 411-428.
19. Mustafa Zeki, 2014. *Geometry Problem Solving*.
20. Crompton, H., 2013. A Historical Overview of M-Learning: Toward LearnerCentered Education. In Z. Berge & L. Muilenburg (Eds.), *Handbook of mobile learning* (pp. 3-14). Routledge, NewYork, USA.
21. Tekbaş, Y., *Code Production Tools Using Automatic Calculation of Derivatives and Simplification Mathematical Expressions*. Master Thesis, Karadeniz Technical University, Institute of Science and Technology, Trabzon, 2013.
22. Gökgöz, B., *Design and Implementation of a general Interpreter for Numerical Root Finding Methods Using Symbolic Approaches*, Master Thesis, Karadeniz Technical University, Institute of Science and Technology, Trabzon, 2016.

23. <https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/>
24. <https://www.edureka.co/blog/what-is-cryptography/>
25. <https://www.log2base2.com/compiler/basics/analysis-and-synthesis-phase-of-compiler.html>
26. <http://yoseph.tech/completely-useless-fun-project-parts-of-the-compiler/>
27. <https://www.thedailyprogrammer.com/2016/03/role-of-lexical-analyzer.html>
28. https://www.tutorialspoint.com/compiler_design/compiler_design_quick_guide.htm



CURRICULUM VITAE

Cheikhna LO from Mauritania, eight years of higher education Master of Computer engineering at the International Black Sea University in Trabzon Turkey, exchange student program Erasmus+ at the University of Czestochowa in Poland, Master's Computer Science without thesis at the University of Sciences Technology and Medicine in Nouakchott Mauritania, and Bachelor degree in IT Management at University of Nouakchott ISCAE in Nouakchott Mauritania.

2017 he obtained a Higher Certificate from ICAT 6th International Conference on Advanced Technology & Sciences after sharing the article Qr code-based encryption and decryption of triangular geometry problems

Language Skills: Arabic as a native language, French: Fluent, English: Advanced and Turkish: Advanced