

# Transition of Source Code From C to Java Using YACC

by Mak Un Pong

Call\_No: MST(S)013.MA

TABLE OF CONTENTS	
List of Figures	v
List of Tables	vi
Glossary	viii
List of Abbreviations	x
<b>Chapter 1: INTRODUCTION</b>	<b>1</b>
1.1    RUNNING ENVIRONMENT	1
<b>Chapter 2: STAGE FOR LANGUAGE TRANSLATION</b>	<b>2</b>
2.1    TYPICAL COMPILER	2
2.1.1    The Scanner	3
2.1.2    The Parser	3
2.1.3    Symbol Tables and Error Handles	3
2.1.4    Semantic Checkers	3
2.2    USING LEX AND YACC FOR LANGUAGE TRANSITION	3
<b>Chapter 3: INTRODUCTION TO C PROGRAMMING</b>	<b>4</b>
3.1    STRUCTURE OF C PROGRAMS	4
3.1.1    C's Character Set	4
3.1.2    Keywords	4
3.1.3    The layout of C Programs	4
3.1.5    Comments	5
3.2    VARIABLES, CONSTANTS, OPERATORS, AND EXPRESSION	5
3.2.1    Identifier Names	5
3.2.2    Basic Data Types	6
3.2.3    Type Modifiers	6

3.2.4	Access Modifiers	7
3.2.5	Variables	7
3.2.6	Storage Class Specifiers	8
3.2.7	Assignment Statement	8
3.2.8	Arithmetic Operators	9
3.2.9	Relational and Logical Operators	9
3.2.10	Bitwise Operators	10
3.3	PROGRAM CONTROL STATEMENTS	10
3.3.1	Input Statement And Output Statement	10
3.3.2	Control Statements	11
3.3.3	Loop	12
3.3.4	Functions	13
3.4	ARRAYS	15
3.5	POINTERS	16
3.6	STRUCTURES, UNIONS AND USER-DEFINED VARIABLES	16
3.6.1	Structures	16
3.6.2	Bit Fields	17
3.6.3	Unions	17
3.6.4	Enumerations	17
3.6.5	typedef	18
3.7	REFERENCE	18
<b>Chapter 4: JAVA</b>		<b>19</b>
4.1	JAVE SOURCE FILES	19
4.1.1	Begining Comments	19
4.1.2	Package and Import Statements	20
4.1.3	Class and Interface Declarations	20
4.2	STATEMENTS	20
4.2.1	Simple Statements	20
4.2.2	Compound Statements	20
4.2.3	return Statements	21
4.2.4	if, if-else, if else-if else Statements	21
4.2.5	for Statements	21

4.2.6	while Statements	21
4.2.7	do-while Statements	22
4.2.8	switch Statements	22
4.2.9	try-catch Statements	22
4.3	<b>MAIN FEATURES OF THE JAVA PROGRAMMING LANGUAGE</b>	23
4.3.1	Primitive Data Types	23
4.3.2	Arithmetic and Relational Operators	24
4.4	<b>FEATURES REMOVED FROM C AND C++</b>	27
4.4.1	No More Typedefs, Defines, or Preprocessor	27
4.4.2	No More Structures or Unions	27
4.4.3	No Enums	27
4.4.4	No More Functions	27
4.4.5	No More Goto Statements	28
4.4.6	No More Pointers	28
4.5	<b>REFERENCE</b>	29
<b>Chapter 5: LEX - A LEXICAL ANALYZER GENERATOR</b>		<b>30</b>
5.1	INTRODUCTION	30
5.2	LEX SOURCE	31
5.3	LEX REGULAR EXPRESSIONS	32
5.4	LEX ACTIONS	34
5.5	AMBIGUOUS SOURCE RULES	36
5.6	LEX SOURCE DEFINITIONS	37
5.7	USAGE	39
5.8	LEX AND YACC	39
5.9	CHARACTER SET	39
5.10	SUMMARY OF SOURCE FORMAT	40
5.11	REFERENCE	41
<b>Chapter 6: YACC - YET ANOTHER COMPILER-COMPILER</b>		<b>42</b>
6.1	INTRODUCTION	42
6.2	BASIC SPECIFICATIONS	42
6.3	ACTIONS	44
6.4	LEXICAL ANALYSIS	46

6.5	AMBIGUITY AND CONFLICTS	46
6.6	PRECEDENCE	48
6.7	ERROR HANDLING	50
6.8	THE YACC ENVIRONMENT	52
6.9	REFERENCES	53
<b>Chapter 7: A SIMLE LANGUAGE TRANSLATOR</b>		<b>54</b>
7.1	A TEXT SCANNER	54
7.2	SEMANTIC CHECKERS	55
7.3	A SIMPLE TRANSLATOR	57
7.4	AN EXAMPLE CAUSES THE SHIFT/REDUCE CONFLICT	60
7.5	REFERENCE	60
<b>Chapter 8: TRANSITION TABLE</b>		<b>61</b>
8.1	THE EASY AND DIFFICULT PART HANDLE	61
8.2	TRANSITION TABLE	62
<b>Chapter 9: SYMBOLS TABLE</b>		<b>82</b>
9.1	USING AN HASH TABLE	82
9.1.1	Bucket	83
9.1.2	Symbol	83
9.1.3	Diagram Of Symbols Table	86
9.1.4	Diagram Of A Program Using Hash Table	86
9.1.4	Diagram Of A Program Using Hash Table	87
9.2	THE LIST DATA STRUCTURE FOR SYMBOLS TABLE	88
9.3	REFERENCE	89
<b>Chapter 10: CODE GENERATION AND ERROR HANDLING</b>		<b>90</b>
10.1	A LEXICAL ANALYZERS	90
10.2	SYNTACTICAL ANALYSIS	91
10.2.1	Grammars	91
10.2.2	Symbol Values	91
10.2.3	Code Generation	91
10.2.4	Symbol Tables	92
10.2.5	Compile and Run the Program	94
10.3	ERROR HANDLING	94

10.4	REFERENCE	94
<b>Chapter 11:</b>	<b>CONCLUSION</b>	<b>95</b>
11.1	PROBLEMS IN DEVELOPMENT	95
11.2	CONCLUSION	95
<b>APPENDIX A:</b>	<b>GRAMMER OF C</b>	<b>96</b>
<b>APPENDIX B:</b>	<b>SOURCE CODE OF LEXICAL ANALYZER - C_LEX.L</b>	<b>101</b>
<b>APPENDIX C:</b>	<b>SOURCE CODE OF YACC PARSING PROGRAM - C_YACC.Y</b>	<b>104</b>
<b>APPENDIX D:</b>	<b>RESULT OF RUNNING PROGRAM</b>	<b>140</b>