**Objective**

To develop a compiler for the ring language

**Grammar**

Program —> {statement}

Statement —> ‘package’ <Identifier> { ‘.’ <Identifier> } [‘{‘ {statement} ‘}’] [‘end’|’endpackage’]

Statement —> ‘class’ <Identifier> [ ‘from’|’:’|’<’ <Identifier> ] [‘{‘ {statement} ‘}’][‘end’|’endclass’]

Statement —> ‘func’|’def’ <Identifier> [ParaList] [‘{‘ {statement} ‘}’][‘end’|’endfunc’]

Statement —> ‘import’ <Identifier> { ‘.’ <Identifier> }

Statement —> ‘private’

Statement —> ‘load’ [‘package’] <Literal>

Statement —> ‘loadsyntax’ <Literal>

Statement —> ‘changeringkeyword’ <OldKeyword> <NewKeyword>

Statement —> ‘changeringoperator’ <OldOperator> <NewOperator>

Statement —> ‘see’|’put’ <Expr>

Statement —> ‘give’|’get’ <Identifier>

Statement —> ‘if’ <Expr> [‘{‘] {statement} [ {‘but’|’elseif’ <Expr> {Statement} } ] [‘else’ {Statement} ] ‘ok’|’end’|’}’

Statement —> ‘Switch’ <Expr> [‘{‘] { ‘on’|’case’ <Expr> {statement} } [‘other’ {Statement} ] ‘off’|’end’|’}’

Statement —> ‘for’ <Identifier> ‘=’ <Expr> ‘to’ <Expr> [ ‘step’ <Expr> ] [‘{‘] {Statement} ‘next’|’end’|’}’

Statement —> ‘for’ <Identifier> ‘in’ <Expr> [ ‘step’ <Expr> ] [‘{‘] {statement} ‘next’|’end’|’}’

Statement —> ‘while’ <Expr> [‘{‘] {statement} ‘end’|’}’

Statement —> ‘do’ {statement} ‘again’ <Expr>

Statement —> ‘try’ {statement} [‘{‘] ‘catch’ {statement} ‘done’|’end’|’}’

Statement —> ‘return’ <Expr>

Statement —> ‘bye’

Statement —> ‘exit’

Statement —> ‘loop’

Statement —> <Expr>

Statement —> epslion

ParaList —> epslion

ParaList —> [‘(‘] <Identifier> [{ ‘,’ <Identifier> }] [‘)’]

**Implementation Language**

Lexical analyzer and parser generating tools: flex and bison

C Language

C18 Standard

**Type of Parser**