Heaven's Light Is Our Guide

RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
4th Year ODD Semester Examination 2019

COURSE NO: **CSE 4101** COURSE TITLE: **Compiler Design**
FULL MARKS: **72** TIME: **3 HRS**

N.B. (i) Answer any **SIX** questions taking any **THREE** from each section.
(ii) Figures in the right margin indicate full marks.
(iii) Use separate answer script for each section.

SECTION : A

Q.1. (a) Explain how a High level language is converted to Target Machine code with figure. 5

(b) Symbol table is necessary for compiler construction – Do you agree with this statement? Justify your answer with example. 4

(c) Consider following figures: 3

Machine 1 (M1)	Machine 2 (M2)
Architecture: X	Architecture: Y
Available Compiler: P language Compiler	Available Compiler: None

Assume P Language compiler can also act as cross compiler for Y architecture. Suggest best suitable strategies to implement follow:
i) Port Language P to M2 machine.
ii) Make Language P self compiling in M2 machine.
iii) Implement new language Q in M2 machine.

Q.2. (a) What are the differences between tokens, patterns and lexemes? Determine tokens and lexemes of the following C statement: 5

```
printf("University = %s\n", RUET);
```

(b) What is a regular expression? State the rules, which define regular expression. 3

(c) How can a compiler be generated using LEX and YACC? Draw the block diagram. 4

Q.3. (a) Eliminate left recursion of the following grammar 3

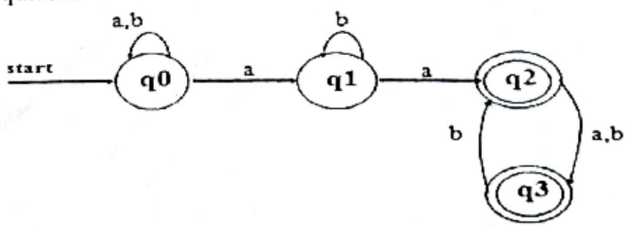
$X \rightarrow Yp \mid Xq \mid pq$
$Y \rightarrow d$

Draw a parse tree for string dpq with and without left recursion.

(b) Prove whether the following grammar is SLR(1) or not: 3

$S \rightarrow AS \mid b$
$A \rightarrow a \mid S$

(c) Convert the following NFA to DFA that accepts the same language as NFA. [Calculate step by step. Transition table and transition diagram are mandatory]. Why is such type of conversion required? 6



Q.4. (a) Distinguish between syntax tree and parse tree. 2

(b) Which parsing is practically implemented – top down or bottom up? Justify your answer. 2

(c) Define Syntax Directed Translation(SDT) and Syntax Directed Definition(SDD) with example. 4

(d) What is Abstract Syntax Tree (AST)? Write SDT to construct AST for following code: 4

```
while(i<10)
        if(a<b)    i = i +1;
```

Draw constructed syntax tree for this code.

SECTION : B

Q.5. (a) Explain importance of FIRST and FOLLOW in predictive parsing. Design a predictive parser by creating parsing table for the following CFG: 6

$A \rightarrow BX$
$X \rightarrow +C \mid -C$
$B \rightarrow a \mid b$
$C \rightarrow x \mid y$

Parse a+x string using this parser.

(b) What are the different strategies that a parser can employ to recover from a syntactic error? Discuss any three of them. 3

(c) What is type checking? Write down SDD with necessary type checking code to generate three address code from the following code: 3

While(x<y | | i!=10) i = i*x

Constraints: No type conversion is allowed and only Long & Double data types are available.

Q.6. (a) Suppose that we have a production A→BCD. Each of the four nonterminals A, B, C and D have two attributes: s is a synthesized attribute and i is an inherited attribute. For each of the sets of rules below, determine whether i) the rules are consistent with an S-attributed definition ii) the rules are consistent with an L-attributed definition and iii) whether the rules are consistent with any evaluation order at all? 6

α) A.s = B.i + C.s
β) A.s = B.i + C.s
D.i = A.i + B.s
γ) A.s = B.s + D.s
δ) A.s = D.i
B.i = A.s + C.s
C.i = B.s
D.i = B.i + C.i

(c) Consider following C program: 6

```
int factorial(int i){
        if (i<=1) return 1;
        int temp = factorial(i-1);
        return i*temp;}
```

i) Show complete activation tree for factorial(3).
ii) Show how Stack of activation records grow and shrink during entire run time.

Q.7. (a) Define activation record and activation tree. Explain with example. 4

(b) Write the comparison among static allocation, stack allocation and Heap allocation with their merits and limitations. 4

(c) Translate the assignment statement x = f(y+1) + 2 into: 4
i) A syntax tree
ii) Quadruples
iii) Triples
iv) Indirect triples

Q.8. (a) What are primary tasks of code generation? Write assembly code for the following three address code: 4

```
L: b = a[i]
   a[j] = b
   If b < 10 goto L
```

Assume all variables are stored in memory locations

(b) Use code generation algorithm to generate assembly code for the following three address code: 4

```
a = t – u
t = w
w = a + u
```

Assume t & u are live on exit variables and a & w are temporaries. Number of registers is 2.

(c) Consider the following code statement: 4

```
w = 0;
x = x + y;
y = 0;
if(x>z){
        y = x;
        x++ ;  }
else{
        y = z;
        z++;  }
w = x + z;
```

Now, i) Find out the basic blocks and their leaders.
ii) Draw the corresponding control flow graph and find out the loops.

***

Heaven's Light Is Our Guide

RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
4th Year Odd Semester Examination 2018
COURSE NO: CSE 4101 COURSE TITLE: Compiler Design
FULL MARKS: 72 TIME: 3 HRS

N.B. (i) Answer any SIX questions taking any THREE from each section.
(ii) Figures in the right margin indicate full marks.
(iii) Use separate answer script for each section.

SECTION : A

Q.1. (a) Define compiler and interpreter? Is a Java language processors compiler or interpreter? Explain your answer. 4

(b) Consider the following instruction where b, c, d, w, y are float type and x is integer type variables. 4

z=x/(w+y)*(b-c)*d

Now, generate the output of the first four phases of the compiler.

(c) Compiler front-end consists of lexical analyzer, parser, semantic analyzer, intermediate code generator and each of them processes parts of the source program in particular ways. For each of the following language rule, specify which phase of the compiler should verify that rule. 4

(i) A function is called with the correct number of arguments.
(ii) Undescore characters (_) may appear in the middle of identifiers, but not at the beginning or end.
(iii) Every variable must be declared before it is used in the pogram (Classic C rule).
(iv) Assignment statements must end with a semicolon(;).

Q.2. (a) What are the two forms of intermediate code representation? Describe it's two important properties. 4

(b) Consider the following grammar: 4

$S \rightarrow A$
$A \rightarrow A + A \mid B + +$
$B \rightarrow y$

(i) Draw a parse tree for the input "y+++y++".
(ii) Show a leftmost derivation of "y+++y++".

(c) What language is generated by the following grammars? In each case justify your answer. 4

i) $S \rightarrow aSbS \mid bSaS \mid \varepsilon$
ii) $S \rightarrow +SS \mid -SS \mid a$

Q.3. (a) What are the goals of error handler? Describe the advantages and disadvantages of panic mode recovery. 4

(b) Consider the following grammar: 4

$X \rightarrow YaYb \mid ZaZb$
$Y \rightarrow \varepsilon$
$Z \rightarrow \varepsilon$

i) Using the definition of LL(1), explain why the grammar is or is not LL(1).
ii) Show that whether the grammar is or is not SLR(1).

(c) What is left recursion? How to eliminate it? Consider the following grammars and eliminate the left recursion: 4

$E \rightarrow E+T \mid T$
$T \rightarrow T*F \mid F$
$F \rightarrow (E) \mid id$

Q.4. (a) Write the impacts of the following functions in NFA to DFA conversion with necessary examples. Here, S and T are single state and set of states respectively. 3

(i) ε- closure(S)
(ii) ε- closure(T)
(iii) move(T, a)

(b) Why do we use FIRST and FOLLOW? Find out the FIRST and FOLLOW of the following grammar: 5

$C \rightarrow P\ F$ class id $X\ Y$
$P \rightarrow$ public $\mid \varepsilon$
$F \rightarrow$ final $\mid \varepsilon$
$X \rightarrow$ extends id $\mid \varepsilon$
$Y \rightarrow$ implements $I \mid \varepsilon$
$I \rightarrow$ id J
$J \rightarrow$, $I \mid \varepsilon$

(c) What is input buffering? How can you speed up the lexical analyzer using buffer 4

pairs and sentinels?

SECTION : B

Q.5. (a) What do you mean by dependency graph? Draw a dependency graph for 3*5. 4

(b) Design a syntax Directed Definition (SDD) for arithmetic expression with operators '+' and '*'. Then from the SDD, generate the annotated parse tree for 1*2*3*(4+5)n. 5

(c) Consider the following source program:

```
if (flag)  x=-1;    else  x=2;
z=x+b;
```

Now, find out the static single Assignment form of the above source program. 3

Q.6. (a) What is intermediate code? What are the advantages of it? 4

(b) Write -down the three-address code representation of the following code segment.

```
while(true){
        do  i=i+1; while(a[i]<v);
        do  j=j-1; while(a[j]>v);
        if(i>=j)  break;
        x=a[i]; a[i]=a[j]; a[j]=x;}
```

3

(c) Explain narrowing and widening of type conversions with necessary diagram. 2

(d) When are two type conversions equivalent?

Q.7. (a) What is activation tree? Which informations are considered as an activation record? Explain briefly. 4

(b) How can you access nonlocal data using access link? What is it's limitation? How can you remove this using display? 4

(c) What are the problems of a block format? Which approaches are used for implicit deallocation? Explain briefly. 4

Q.8. (a) Consider the following instructions: 6

(1) prod: =0
(2) i:=1
(3) t_1:= 4*i
(4) t_2: =a[t_1]
(5) t_3: =4*i
(6) t_4: =b[t_3]
(7) t_5: =t_2*t_4
(8) t_6: =prod + t_5
(9) prod: = t_6
(10) t_7:=i+1
(11) i:=t_7
(12) if i<=20 goto (3)
(13) t_8:= t_5+t_7

(i) Find out the basic blocks and their leaders.
(ii) Draw a control flow graph. 3

(b) Which issues are considered in code generation? Explain. 3

(c) What are the problems of using registers for code generation? 3

***

Heaven's light is our guide

Rajshahi University of Engineering & Technology

B.Sc. Engineering 4th Year Odd Semester Examination, 2017

Department of Computer Science & Engineering

Course no: **CSE 4101** Course Title: **Compiler Design**

Full marks: 72 Time: Three (03) hours

N.B. Answer **six** questions, taking **three** from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION-A

Q1. (a) Depict diagrammatically how a language is processed? **5**

(b) What are the advantages in a language processing system where the compiler produces assembly language rather than machine language? **4**

(c) Is it possible to use compiler to translate a high-level language into another high-level language? If it is possible then what are the advantages of using C as a target language for a compiler? If the answer is false than write the reasons behind this. **3**

Q2. (a) Write short notes: Dependency graph, annotated tree, synthesized attribute. **3**

(b) "A grammar can have more than one parse tree generating given string of terminals." What does this statement indicates? Explain with example. **3**

(c) What language is generated by the following grammars? In each case justify your answer. **2**
i) A→+AA | -AA |a
ii) B→a | B+B | BB | B* | (B)

(d) Suppose a robot can be instructed to move one step east, north, west or south from its current position. Design and explain a grammar for such sequence of instructions. **4**

Q3 (a) What are the problems with top down parsing? How does bottom-up parsing solve these? **4**

(b) What is left recursion? Eliminate left recursion from the grammar: **4**
S→Aa|b
A→ AC |Sd|e

(c) Can we run out of buffer space in case of Lexical analysis phase of compiler design? Explain. **4**

Q4. (a) Develop a regular definition for unsigned numbers that includes integer or floating point numbers. [Use shorthand notations] **3**

(b) Convert the NFA of Figure 1 to DFA that accepts the same language as NFA. Show the necessary steps. Verify your answer that it works. [Transition table is mandatory] **6**

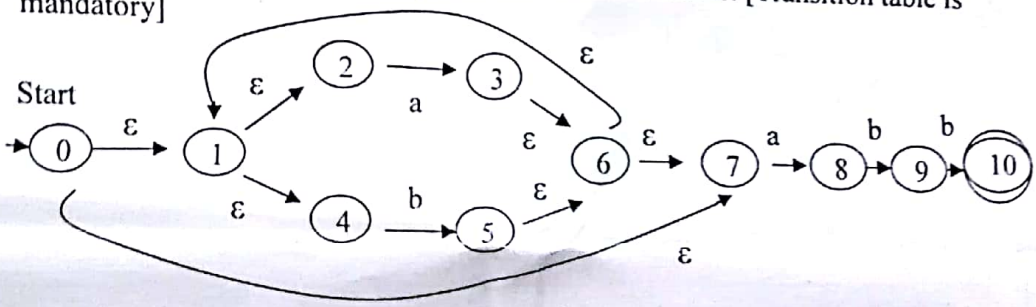


Figure 1: NFA for (a|b)*abb

(c) Suppose, we have two tokens: i) the keyword if and ii) identifiers, which are strings of letters other than if. Show: **3**
a) The NFA for these tokens, and
b) The DFA for these tokens.

SECTION-B

Q5. (a) Compare the different implementations of three address codes with example. **4**

(b) Are the attributes in the following CFG synthesized or inherited? Give reasons. 4

var→ IntConstant
{ $ 0.val=$ 1.lexval;}
Expr→var
{$0.val=$1.val;}
Expr→Expr B_op Expr
{$0.val=$2.val($1.val,$3.val);}
B_op→ +
{$0.val=PLUS;}
B_op→*
{$0.val=TIMES;}

(c) Describe the syntax directed translation procedure for assignment statement with integers and mixed types and explain. 4

Q6. (a) Construct the predictive parser for the following grammar 4

E→ TE'
E'→ +TE' | ε
T→FT'
T'→*FT' | ε
F→(E) | id

(b) Discuss the issues in code generation with example. 4

(c) What does it mean for two type expressions to be equal? Explain with example. 4

Q7. (a) Consider a program that reads nine integers into an array 'a' and sorts them using the recursive quicksort algorithm. Now write down the corresponding program and generate the activation tree representing calls during an execution of quicksort. Also, sketch the activation records when control returns to q(1,3) where q() is quicksort function. 6

(b) Explain the deep access and shallow access approaches for implementing dynamic scoping of non-local variables. 4

(c) What is peephole optimization? Mention some characteristics of it. 2

Q8. (a) Explain the significance of instructions selection to have efficient target program in code generation. Associate your explanation with suitable assembly code instruction. 3

(b) Consider the following Pseudocode that turns a 10×10 matrix A into an identity matrix. 7

```
for i from i to 10 do
    for j from 1 to 10 do
        A[i,j]=0.0;
for i from i to 10 do
        A[i,i]=1.0;
```

Now, i) Translate the above program into three-address statement. Assume the matrix entries are real-valued numbers that requires 8 bytes, and that matrices are stored in row-major order.
ii) Construct the flow graph for your code from (i).
iii) Identify the loops in your flow graph from (ii).

(c) Construct the DAG for the following basic block 2

```
d=b*c
e=a+b
b=b*c
a=e-d
```

******

Heaven's light is our guide

Rajshahi University of Engineering & Technology

B.Sc. Engineering 4th Year 7th Semester Examination, 2016

Department of Computer Science & Engineering

Course no: **CSE 701** Course Title: **Compiler Design**

Full marks: 70 Time: Three (03) hours

N.B. Answer **six** questions, taking **three** from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION-A

Q1. (a) Give a comparison among parse tree, syntax tree and dag. Draw a dag for the following expression: **4**
(((a+b)-e)*(d+e))+(a+b))*(((a+b)*(d+e))-e)

(b) What is lex? How lex is used to create a lexical analyzer? **$3^{2/3}$**

(c) Define the ambiguous and unambiguous grammar. Consider the following grammars and show that these grammars are ambiguous. **4**
E→EAE|(E)|-E|id
A→+ | - | * | ↑ | /

Q2. (a) Suppose a robot can be instructed to move one step east, north, west, or south from its current position using the following grammar: **5**
seq→seq inst | begin
inst→ east | north | west | south

i) Change in the position of the robot on input: begin east south west west north

ii) Draw the annotated parse tree for begin east south.

(b) What are the roles of lexical analyzer in compiler design? Why the analysis portion of a compiler is normally separated into lexical analysis and parsing phases? **4**

(c) Why the attributes of tokens are so important? What is sentinels? **$2^{3/3}$**

Q3 (a) In which phase of compiler every character of source text is checked? **2**

(b) Describe different error recovery strategies. **4**

(c) Construct a syntax-directed translation scheme that translates arithmetic expressions from the postfix notation into infix notation. Give annotated parse trees for inputs **$5^{2/3}$**
95-2* and 952*-

Q4. (a) Determine whether the following statement can be achieved by the given grammar using stack structure (id+id*id) **3**
E→E+T |T
T→T*F|F
F→(E)|id

(b) What is lexical scope and dynamic scope? Write a program and indicate the output by these scopes **$4^{2/3}$**

(c) What do you mean by lexical error and syntax error? Is it possible to detect all the errors present in the source program in lexical analysis and syntax analysis? Explain your answer with example. **4**

SECTION-I

Q5. (a) **4**

```
a:=0;
i:=1;
repeat{
   a:=a+1;
   i:=i+1;
   until i>20
      }
```

For the above program
i) Find out the basic blocks
ii) Draw the control flow graph

(b) Mention the various notational shorthands for representing regular expression. **3**

(c) S→aABc
A→Abc|b
B→d
For the above grammar derive the string abbcdc using top-down parsing and bottom-up parsing (step by step derivation).

Q6. (a) Are there any relation between finite automata and compiler design? Design a nondeterministic finite automaton for the language: "All strings of lowercase letters that contain the five vowels inorder" 5

(b) Define the following terms: 3
i) Σ^*, ii) Σ^+, iii) $\Sigma?$

(c) Compare quadruples, triples, and indirect triples to represent three address codes. $3^{2/3}$

Q7. (a) Eliminate the left recursion from the following grammar 2
A→Aabc |a|b|c

(b) Find out the FIRST and FOLLOW of the non-terminals using the following production rules and also construct a parse table. $5^{2/3}$
E→(F) |a
F→EF'
F'→*E F'| ϵ

(c) Briefly explain deep access and shallow access with example. 4

Q8. (a) Consider the following grammar 5
S→iE{SS'
S→a
E→b
Construct parsing table for these grammar. Do you face any problem to construct the table? Then mention the problem and solve it.

(b) A.a=f(X.x,Y.y) $2^{2/}$
X.x=q(A.a,Y.y)
Which type of attribute is showed by the above two statements. Is there any difference between them?

(c) Describe the type checking of the following expression: 4
i) $E=E_1[E_2]$, ii) $E=E_1\uparrow$, iii) S→if E then S_1, iv) $E\rightarrow E_1$ op E_2

******

B.Sc. Engineering 4th Year 7th Semester Examination, 2015
Department of Computer Science & Engineering
Course No. CSE 701 Course Title: **Compiler Design**
Full Marks: 70 Time: **THREE** (03) hours

N.B:

Answer **SIX** questions taking **THREE** from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION A

		Marks
Q.1(a)	What do you mean by programming language? Did you notice any benefits of compiler over interpreter and interpreter over compiler? Explain with example.	03⅓
(b)	Explain the phases of a compiler with the help of the following statement: d = a + b – c, where a, b, c and d all are real values.	05
(c)	What advantages are there to language processing system in which the compiler produces assembly language rather than machine language?	03
Q.2(a)	How can you differentiate between token and lexeme? Where the attributes for token are stored?	04
(b)	Is symbol tabling an essential part of a compiler? Justify your answer.	02⅔
(c)	Consider the following grammars: S→ a | A | (T), T→ T, S | S. In this grammar, find out the leftmost and right most derivation for the following: (i) (a, (a, a)) and (ii) (((a, a), A, (a)), a) A, (a)), a).	05
Q.3(a)	What is CFG? Are there any relationship between CFG and compiler? Explain briefly.	03⅔
(b)	Consider the following infix expression Q: = ((A + B) * D) ↑ (E - F). Convert to postfix notation and shows the stack contents.	03
(c)	Is it possible to detect all the errors present in the source program in lexical analysis and syntax analysis phase? Explain with example.	05
Q.4(a)	What is type checking? When is it usually done?	03
(b)	Translate the expression (a + b) * (c + d) + (a + b + c) into (i) quadruples (ii) Triples and (iii) Indirect Triples.	04
(c)	For predictive parsing, the grammar A→ A A |(A) | ε is suitable or not? If not then why?	02⅔
(d)	Distinguish between static type checking and dynamic type checking.	02

SECTION B

Q.5(a)	When the attributes of syntax directed translation is called synthesized and when is called inherited? Explain with example.	03⅓
(b)	Consider you are designing a compiler for C language for your personal use. Which error recovery techniques will you implement for handling lexical error? Why?	04
(c)	Draw a transition diagram for unsigned numbers in C.	04
Q.6(a)	What do you mean by coercions in type conversion of a programming language?	02⅔
(b)	Determine the FIRST () and FOLLOW () of the following grammar: S→ Bc|DB, B→ab| cS, D→d|ε.	06
(c)	Consider the following three address code: a := b + c, b := a – d, c := b + c, x := a – d, y := b + c. Eliminate common sub-expression.	03
Q.7(a)	Describe the error recovery strategies.	03⅓
(b)	Generate the grammar that produces real number like ± 10.02 x $E^{\pm 10.0}$.	03
(c)	Consider the context free grammar S→ ss^+ | ss^* | a. (i) Show how the string $aata^*$ can be generated by this grammar and (ii) Construct a parse tree for this string.	03
(d)	What are the advantages of notational shorthand?	02
Q.8(a)	What is the purpose of optional control link?	02
(b)	For the expression, a + b * c – d/b*c. Construct (i) Syntax tree and (ii) DAG.	04
(c)	What is static allocation? What is its limitation?	02⅔
(d)	In a string of length n, how many of the following are there? (i) Prefixes (ii) Suffixes (iii) Substrings.	03

Rajshahi University of Engineering & Technology

Department: CSE Course No: 701 Mark: 20 Roll:

1. Translate the arithmetic expression a * - (b +c) into a) a syntax tree b) three-address code. 3

2. Translate the expression x[i] = y + 1 and x = y[i] + 1 into Triples. 4

3. Write the type checking semantic rule for the expressions: 3

a) S → **while** E **do** S_1 b) E → E_1 [E_2] c) E → E_1 op E_2

P-360

E. type =

3 (P-373)

P-374,

4. Consider the following fragment of code:

0: (A) t0 = read_num
1: (A) if t0 mod 2 == 0
2: (B) print t0 + " is even."
3: (B) goto 5
4: (C) print t0 + " is odd."
5: (D) end program

How many blocks are there? Identify them. Drew the control flow diagram of the code. 1 + 2 + 2

5. What are the benefits of quadruples over triples?

Heaven's Light is our Guide

Rajshahi University of Engineering & Technology

B.Sc. Engineering 4th Year 7th Semester Examination, 2014

Department of Computer Science & Engineering

Course No. CSE 701 Course Title: **Compiler Design**

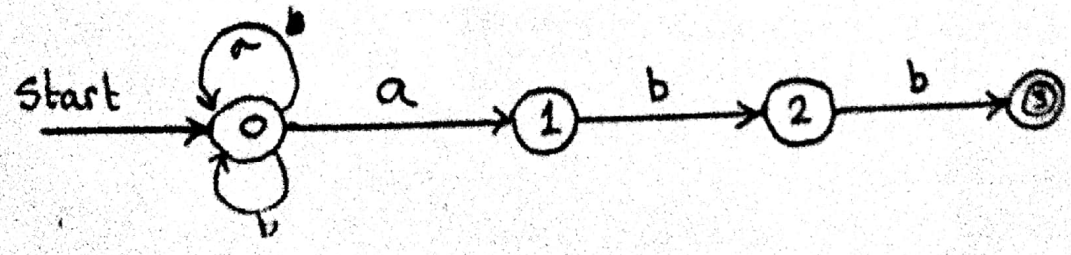
Full Marks: 70 Time: **THREE** (03) hours

N.B.

Answer **SIX** questions taking **THREE** from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION A

		Marks
Q1(a)	With suitable block diagram, describe a language processing system.	03½
(b)	Define the following terms: (i) Compiler (ii) Interpreter (iii) Assembler.	03
(c)	Explain the phases of a compiler with the help of the following statement. Position:= initial + rate/60 (where position, initial and rate all are real values)	05
Q2(a)	Is there any difference between tokens and terminals? If the answer is "yes" then differentiate or if the answer is "no" then explain why.	02½
(b)	What is parse tree? How a parse tree is produced according to a context free grammar? Explain with example.	04
(c)	Consider the context free grammar S→ ss+|ss*|a (i) Show how the string aa+a* can be generated by this grammar. (ii) Construct a parse tree for this string.	03
(d)	Eliminate left recursion from the following grammar: (i) E→E+E (ii) E→ id	02
Q3(a)	Why symbol table is essential in compiler design? Who creates symbol table entries?	03½
(b)	What is the role of the lexical analyzer? Why the analysis portion of a compiler is normally separated into lexical analysis and parsing phases?	04
(c)	Describe the languages denoted by the following regular expressions: (i) (a|b)(a|b) (ii) a|a*b (iii) (a|b)* (iv) ((a|∈)b*)*	04
Q4(a)	Consider the following grammar: E→ E + E|E*E|(E)|-E|id Prove that, -(id+id) is a sentence of that grammar.	03½
(b)	What is parse tree? How can you remove ambiguity from the grammar?	03
(c)	Convert the NFA given below into corresponding DFA using the necessary algorithm.	05



SECTION B

Q5(a) What is meant by aligned and padding? Give some examples.

(b) Write the rules of FIRST () and FOLLOW () functions. Determine the FIRST () and FOLLOW () of the following grammar. 04 06

E→E+T|T
T→T*F|F
F→(E)|id

(c) What is flow graph? A flow graph is a graphical representation of a program in which the nodes of the graph are basic blocks and the edges of the graph show how control can follow among the blocks. 03

Q6(a) Write the benefits of syntax-directed definition than semantic action. 02

(b) What does it mean for two type expressions to be equal? 03

(c) Consider the followings: 03

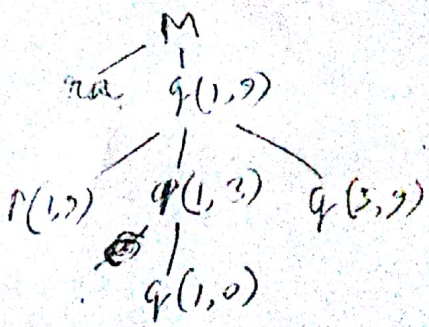
Type vector = array [1....10] of real
Type weight = array [1....10] of real
var x,y : vector; z : weight
Mention which are name equivalent and which are structural equivalent.

(d) What do you mean by coercions? Why is it important? 03

Q7(a) Which issues are dealing by the run time environment? 02

(b) What is activation trees? Draw the activation tree of the following statements: 03

Execution begins ..
enter readarray
leave readarray
enter quicksort(1, 9)
enter partition(1, 9)
leave partition(1, 9)
enter quicksort(1, 3)
enter quicksort(1, 0)
leave quicksort(1, 0)
leave quicksort(1, 3)
enter quicksort(5, 9)
leave quicksort(5, 9)
leave quicksort(1, 9)
execution terminated



(c) int x;
x = 5;
Here, what is environment and what is state? Why? 02

(d) Describe the storage allocation strategies. 04

Q8(a) Define a Directed Acyclic Graph. Construct a DAG and write the sequence of instructions for the expressions, 04

a+a* (b-c) + (b-c)*d

(b) Translate the arithmetic expression Z = a*(c+d)(a+b+c) into – 05

(i) a syntax free
(ii) three address code.

(c) Find out the type of error of the following statements: 02

(i) if (a=2) a=a+1; Logical
(ii) return statement in a Java method with result type void. semantic
(iii) missing braces. syntactic

****

Rajshahi University of Engineering and Technology
B.Sc. Engineering 4th Year 7th Semester Examination, 2013
Department of Computer Science and Engineering
Course No. CSE 701 Course Title: Compiler Design
Full Marks: 70 Time: THREE (03) hours

N.B.

Answer SIX questions taking THREE from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION A

		Marks
Q1(a)	What are the two parts of compilation? Explain briefly.	03⅓
(b)	Depict diagrammatically how a language is processed.	05
(c)	Mention some of the cousins of a compiler.	03
Q2(a)	Write down the output of each phase for expression: $a:=b+c*50;$	05⅓
(b)	Describe the language denoted by the following regular expression:	03
	(i) $(0\|1)^*0(0\|1)(0\|1)$	
	(ii) $0(0\|1)^*0$	
	(iii) $(00\|11)^*((01\|10)(00\|11)^*(01\|10)(00\|11)^*)^*$	
(c)	Define, (i) $L.L^*=?$ (ii) $\bigcup_{i=0}^{\infty} L^i=?$ (iii) $(\pi\|\varepsilon)^*=?$	03
Q3(a)	What optimization can you propose for the code $a=b*c;$ $x=b*c+5;$	03⅓
(b)	What is LL(1) grammar? Write shorthand notation of rr^*	04
(c)	State and discuss different storage allocation strategies.	04
Q4(a)	Why the analysis portion of a compiler is normally separated into lexical analysis and parsing phases?	02⅔
(b)	Write the regular expression for (i) an identifier (ii) an unsigned number.	06
(c)	Draw the transition diagram for the statement $(a(b^*c))\|(a(b\|c^*))$	03

SECTION B

Q5(a)	Consider the following grammar $S\rightarrow(L)\|a$ $L\rightarrow L,S\|S$ (i) What are the terminals, non terminals and start symbol? (ii) Find parse three for the following sentence $(a,(a,a))$	04 02⅔
(b)	Why do we study lexical and syntax analyzer?	05
(c)	Eliminate left recursion from the grammar $S\rightarrow Aa\|b$ $A\rightarrow Ac\|Sd\|e$	04
Q6(a)	Can you minimize the number of states of *DFA*? Write an algorithm to minimize and justify your answer with examples.	04
(b)	Give an overview of shift reduces parsing and write the issues in it.	03⅓
(c)	Write an algorithm to construct *SLR* parsing table with all sub routines.	04⅓
Q7(a)	Define a Directed Acyclic Graph. Construct a *DAG* and write the sequence of instructions for the expression $a+a^*(b-c)+(b-c)^*d$	04
(b)	What are the possibilities of non recursive predictive parsing? Write a short note on token patterns and Lexemes.	03
(c)	Explain the specification of simple type checker for statement, expressions and functions.	03
Q8(a)	What are the benefits of intermediate code generation?	04
(b)	Write the three address code for the statement $a=b^*-c+b^*-c$	02⅔
(c)	"A basic block has two leaders." Is it true? Justify your answer?	02
(d)	What is flow graph?	

****

Rajshahi University of Engineering & Technology
B Sc. Engineering 4th Year 7th Semester Examination, 2012
Department of Computer Science & Engineering
Course no: CSE 701 Course Title: Compiler Design
Full marks: 70 Time: Three (03) hours

N.B. Answer six questions, taking three from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION-A

Q1. (a) Explain the phases of a compiler with the help of the following statement: Slide1-Page17 4

Position := initial + rate/60

(where position, initial and rate all are real values)

(b) Define loader, linker and assembler. Distinguish between one pass and two pass assembler. S-1, P-20,21 3⅔

(c) Determine whether the following [illegible] can be achieved by the given grammar using stack structure. $(id + id * id)$ 4

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

S1-9

Q2. (a) Distinguish between linear analysis and hierarchical analysis. Discuss the role of symbol table in compiler design. S1-15 4

(b) What is pre-processors? Discuss three important functions of pre-processors. S1-19 4

(c) Suppose a robot can be instructed to move one step east, north, west or south from its current position. Generate the grammar and show the graphical path for the input-begin west south east east east north north. S2-15 3⅔

Q3. (a) What are the phases of a compiler? Why the intermediate code generation phase is needed? S1-16 2⅔

(b) Why symbol table is used in compiler? What are the contents of it? If we do not use symbol table what will have happened? S1-15 4

(c) Suppose you have a computer in your office, where you have created a program code. Now you want to convert the code (which is now in high level language) to it's corresponding machine code. [illegible] to convert the program. But after conversion, [illegible] machine code to your personal computer at home. So which will you choose, either compiler or interpreter for conversion? Why? 5

Q4. (a) What is control stack? How a control stack is maintained when procedure are executed? S7-6 4

(b) [illegible] are studying [illegible] you may need to design a new compiler? 3

(c) [illegible] stack allocation strategy can not be used? By which technique this problem can be eliminated? S7-15 4⅔

SECTION-B

Q5. (a) Construct recursive descent parser for the grammar, $R \rightarrow +RR \mid -RR \mid a$ S4-20 4

(b) Discuss the model of a non recursive predictive parser with suitable diagram. S4-21 4

(c) Why we have to build parsing table in compiler design? Is it possible to built a parsing table without the calculation of FOLLOW? 3⅔

Q6. (a) Why garbage collection is needed? 2⅔

(b) Explain the function of loader and linker. B-19 4

(c) Show how each phase of a compiler will change the following statement: 5

*Total_Score := 6 * Total_Six + 4 * Total_Four + Single + Extra.*

[illegible] integer type.

Q7. (a) Write an algorithm to closure and complete. 3⅔

(b) Write the type checking expressions for the following expression and functions: 4

(i) $E \rightarrow E_1 [E_2]$

(ii) $E \rightarrow E_1 \uparrow$

(iii) $E \rightarrow (E)$ Book-350

(c) What is the mechanism if parameter is passed by Copy-Restore? 7-26 4

Q8. (a) What is equivalences of type expressions? Discuss about the structural equivalence of type expressions. S6-10 3

(b) What is meant by aligned and padding? Give some examples of alignment and padding. S4-14 Book 399 4

(c) Design a parsing table for grammar, 4⅔

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$

Where i, t and e stand for if then and else. S4,27

*****

Scanned by CamScanner

Heaven's light is our guide

Rajshahi University of Engineering & Technology

B.Sc. Engineering 4th Year 7th Semester Examination, 2011

Department of Computer Science & Engineering

Course no: CSE 701 Course Title: Compiler Design

Full marks: 70 Time: Three (03) hours

N.B. Answer six questions, taking three from each section.
The questions are of equal value.
Use separate answer script for each section.

SECTION-A

Q1. (a) Discuss the analysis-synthesis model of compilation. S1-07 — $2\frac{2}{3}$

(b) "An important component of semantic analysis is type checking" if the statement is correct then, explain with the help of example. S6-02 — 03

(c) Define predictive parsing method. Is there any backtracking occur here? If not, then why? S2-26 — 03

(d) Mention why symbol table is used in compiler. S1-15 — 03

Q2. (a) What is context free grammar? Write components of CFG. S2-3 — 04

(b) Construct a syntax directed translation scheme that translates roman numerals into integers. — 04

(c) Construct recursive descent parser for the following grammar — $3\frac{2}{3}$
$R \rightarrow +RR \mid -RR \mid a$ S4-20 S4-5 S2-16

Q3 (a) What is parsing? Describe different types of error recovery strategies. J64 — $3\frac{2}{3}$

(b) Briefly explain how a non-recursive predictive parser can be built by maintaining stack. S4-21 — 04

(c) Consider the following grammar — 04
$E \rightarrow E+T \mid T$
$T \rightarrow TF \mid F$
$F \rightarrow F^* \mid a \mid b$
Construct the SLR and LALD parsing table for this grammar.

Q4. (a) What is control stack? How a control stack is maintained when procedures are executed? S7-16 — 04

(b) When the stack allocation strategy cannot be used? By which technique this problem be eliminated? Explain. S7-15 — 04

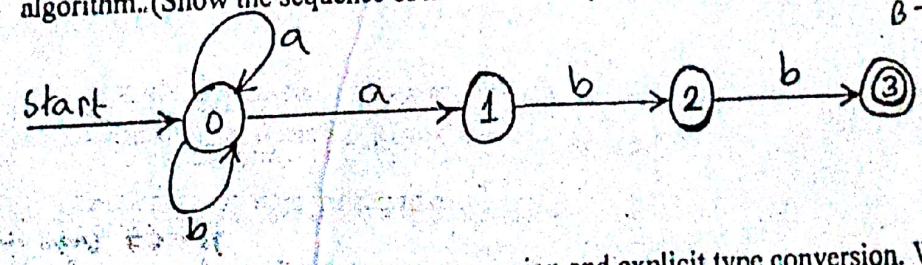
(c) Show the triple representation of — $3\frac{2}{3}$
$X[i] := y+1$ & $X := y[i]+1$

SECTION-B

Q5. (a) Discuss the interaction between the lexical analyzer and the parser. S3-2 — $2\frac{2}{3}$

(b) What is the difference between NFA and DFA? Construct a NFA for the regular expression $R=(a|b)^*abb(a|b)^*$ S3-27 — 03

(c) Convert the NFA given below into corresponding DFA using the necessary algorithm. (Show the sequence of moves made by each processing step) — 06
B-119



Q6. (a) Distinguish between implicit type conversion and explicit type conversion. Write the type checking rules for coercion from integer to real. S6-12 — $4\frac{2}{3}$

(b) If a program consists of following statement – — 02
$S \rightarrow$ *if E do* $(S_1 \text{ mod } S_2)$
Write the type checking statements for the expression.

(c) Using syntax directed approach draw the tree and dag for the following type expression: — 03
char×*char* → *pointer (integer)*

(d) Draw the DAG for the following expression: — 02
$Z = X-Y+X^*Y^*U-V/W+X+V$

Q7. (a) What is meant by aligned and padding? Give some example of alignment and padding? 04

(b) What is a display structure? How display is implemented for activation record? Show with example. 05

(c) Design a parsing table for grammar
$S \rightarrow iEts \mid iEtSeS \mid a$
$E \rightarrow b$
Where i, t and e stand for if, then and else.

Q8. (a) What is a block? Write the scope of declaration in a block given by the most closely nested rule. 04

(b) What are the problems in implicit deallocation if block structure is used? Briefly explain the approaches for implicit deallocation. 04

(c) What is flow graph? Determine the basic blocks of the following program:

```
int n=5, count=1;
float x, average, sum=0;
do{
printf("x= ");
scanf("%f", &x);
sum+=x;
++count;
} while (count<=n);
average =sum/n;
printf("\n Average= %f", average);
```

******