B.Sc. (ECS) (Part – III) (Semester – VI) (Old) Examination, 2017
COMPUTER SCIENCE
Compiler Construction (Paper – IV)

Time : 2 Hours   Total Marks : 50

Instructions : *1)* ***All*** *questions are* ***compulsory****.*
2) Figures to the ***right*** *indicate* ***full*** *marks.*

1. Choose the **correct** alternatives. **10**

1) A bottom up parser generates ____________
   a) Right most derivations
   b) Right most derivations in reverse
   c) Leftmost derivations
   d) Leftmost derivations in reverse

2) Which of the following is used for grouping of characters into tokens ?
   a) Parser   b) Code optimizer
   c) Code generator   d) Scanner

3) In a syntax directed translation scheme, if values of an attribute of a node are a function of the attributes of its children, then attribute is called __________
   a) Canonical attribute   b) Synthesized attribute
   c) Inherited attribute   d) None of these

4) Concept which can be used to identify loops is ____________
   a) Dominators   b) Reducible graphs
   c) Depth first ordering   d) All of these

P.T.O.

5) A compiler that runs on one machine and produces code for a different machine is called ____________
   a) One pass compilation   b) Two pass compilation
   c) Cross compilation   d) None of these

6) A parse tree showing the values of attributes at each node is called an ____________
   a) Derivation tree   b) Directed acyclic graph
   c) Annotated parse tree   d) All of the above

7) An important component of semantic analysis is ___________
   a) Code checking   b) Type checking
   c) Flush checking   d) All of the above

8) A memory allocates and deallocates storage as needed at runtime from data areas known as _____________
   a) Heap   b) Stack
   c) Static   d) All of these

9) Which of the following parser is most powerful ?
   a) Operator precedence   b) Canonical LR
   c) LALR   d) SLR

10) ____________ values of actual parameters are passed to caller procedu
    in call by value.
    a) R   b) L
    c) Both a and b   d) None of these

3. A) Attempt **any two.** **6**

    1) Explain in details factors affecting pass structure of compiler.

    2) What does the activation record ? Explain field of it.

    3) What is backtracking ? Explain backtracking with example.

   B) Construct the predictive parsing table for the following grammar. **4**

   G = ({E, E', T, T', F}, {+, *, (,), id, num}, P, E)

   Where P :

   $E \rightarrow TE'$

   $E' \rightarrow +TE' | \epsilon$

   $T \rightarrow FT'$

   $T' \rightarrow *FT' | \epsilon$

   $F \rightarrow (E) | id | num$

4. Attempt **any two.** **10**

    1) Explain in details storage allocation strategies.

    2) What is code optimization ? Explain in details principal source of optimization.

    3) What is intermediate code generation ? Explain types of three address code implementation of statements.

5. Attempt **any two.** **10**

    1) What is bottom-up parser ? Explain in detail shift reduce parsing using stack implementation.

    2) Construct annotated parse tree for the expression : 3 * 5 + 4n using following production.

Production	**Semantic rule**
$L \rightarrow E_n$	print (E.val)
$E \rightarrow E_1 + T$	$E.val := E_1.val + T.val$
$E \rightarrow T$	$E.val := T.val$
$T \rightarrow T_1 * F$	$T.val : T_1.val * F.val$
$T \rightarrow F$	$T.val := F.val$
$F \rightarrow (E)$	$F.val := F.val$
$F \rightarrow digit$	$F.val := digit.lexval$

    3) Explain in details issue in the design of a code generator.

____

B.Sc. (ECS) – III (Semester – VI) (New CGPA) Examination, 2017
COMPLIER CONSTRUCTION (Paper – IV)

Time : 2.30 Hours Total Marks : 70

Note : *1)* ***All*** *questions are* ***compulsory.***
2) Figures to the ***right*** *indicate* ***full*** *marks.*

1. Choose the **correct** alternative for the following: **14**

1) One complete scan of the source language is called __________

a) Pass b) Phase
c) Both a and b d) None of these

2) __________ generators generate lexical analyzer.

a) Parser generator b) Data flow engines
c) Scanner generator d) Automatic code

3) __________ set of rules describe the token.

a) Tokens b) Patterns
c) Lexemes d) None of these

4) In __________ parsing the parse tree is generated from top to bottom.

a) Topdown parser b) Bottomup parser
c) Both a and b d) None of these

5) The parser that uses collection of recursive procedures for parsing the given input string is called __________

a) Recursive descent parser b) Shift reduce parser
c) LL (1) parser d) LR parser

6) __________ is defined as the set of terminal symbol that appear immediately the right of A.

a) First() b) Goto()
c) Closure() d) Follow()

7) The syntax directed definition that uses only synthesized attributes is called __________

a) S-attributed definition b) l-attributed definition
c) Both a and b d) None of these

8) __________ allocation is done for all data objects at compile time.

a) Static b) Stack
c) Heap d) None of these

9) A __________ is a context grammar together with attributes and rules.

a) Syntax-Directed Definition b) S-attributed definition
c) L-attributed definition d) All of the above

10) __________ represents the pattern for number.

a) [a-zA-Z][a-zA-Z0-9]* b) [0-9][0-9]*
c) < =|>|==|>= d) None of these

11) For the grammar S -> (L) |a L -> L, S|S the First(L) = __________

a) {(,)} b) {(,),a}
c) {),a} d) {(,a}

12) Backtracking and no backtracking are occur on __________

a) Topdown parser b) Bottomup parser
c) Both a and b d) None of these

2. Answer the following **(any 7)** : **14**

1) Define Incremental Compiler.

2) Define Lexemes.

3) What is shift-reduce and reduce-reduce conflict ?

4) What is need of semantic analysis ?

5) Write the SDD for the following grammar.

   S->EN
   E->E+T | T
   T->T*F| F
   F-> digit | (E)
   N-> ;

6) What is Symbol Table ?

7) Define Basic Blocks.

8) Explain Pre-Header.

9) Generate the Three address code for a following expression.

   X :=a + b*c + d ;

3. A) Answer the following **(any 2)** : **10**

1) Consider the CFG as,

   S->EN
   E->E+T
   E->E–T
   E->T
   T->T*F
   T->T/F
   T-> F
   F->(E)
   F->digit
   N->;

tree for the input string 5*6+7:

2) Consider the following grammar,

S->aABe A->Abc|b B->d

Find the Handles for the String "abbcde".

3) Explain Activation Record in detail.

B) Explain the concept of Input Buffering. 4

4. Answer the following (**any 2**) : 14

1) Find out SLR parse table for the following grammar :

E->E+T|T

T->T*F| F

F->(E)| id

2) Construct the DAG for the following block.

a:=b*c

d:= b

e:=d*c

b:=e

f:=b + c

g:=f + d

3) Explain principle sources of optimization.

5. Answer the following (**any 2**) : 14

1) Write the following expression in syntax tree, postfix notation and Three address code of intermediate representations.

(a – b)*(c+d) – (a+b)

2) Explain Peephole optimization.

3) Test whether following grammar is LL (1) or not.

E-> E+T| T

T->TF|F

F-> F*T| a | b.

B.Sc. (ECS) – III (Semester – VI) (Old) Examination, 2017
COMPILER CONSTRUCTION (Paper– IV)

Day and Date :Friday, 3-11-2017 Max. Marks : 50

Time : 2.30 p.m. to 4.30 p.m.

Instructions : *1) **All** questions are **compulsory**.*
*2) **All** questions carry **equal** marks.*

1. Choose correct alternatives. **10**

1) The __________ determines the structure of the source string by grouping the tokens together.

a) Lexical Analyzer b) Syntax Analyzer
c) Semantic Analyzer d) None of these

2) The ——— is a sequence of consecutive statements in which flow of control enters at the beginning and leaves at the end without halt or possibility of branching.

a) basic blocks b) flow graphs
c) directed acyclic graph d) all of the above

3) The attributes that can be computed from the values of the attributes at the children's of that node in the parse tree is called as ____________

a) inherited b) synthesized c) both a and b d) none of these

4) ________ is the activity of filling up unspecified information of labels using appropriate semantic actions in during code generation process.

a) Backtracking b) Triplets
c) Intermediate code d) Backpatching

5) The ________ is a new block created such that successor of this block is the header block.

a) inner loops b) reducible c) pre-header d) dominators

6) In case of __________ the size of data objects is known at compile time.

a) stack allocation b) heap allocation
c) static allocation d) both b and c

7) The _____________ parser uses reduction process.

a) Topdown Parser b) Bottom up Parser
c) Either a or b d) Both a and b

8) ________________ parsers begin at the start symbol and try to apply productions to arrive at the target string.

a) L L parser b) LR parser
c) none of these d) all of above

9) ____________ is a compiler which performs the recompilation of only modified source rather than compiling the whole source program.

a) Cross compiler b) Boot strapping compiler
c) One pass compiler d) Incremental compiler

10) __________ is a translation scheme in which the type of each expression is obtained from the types of subexpressions.

a) Type checking b) Type analysis
c) Both a and b d) None of these

2. Solve **any five** : **10**

1) Give the name of phases of compiler.

2) Explain left factoring with example.

3) Role of semantic analysis.

4) Give the types of errors handled by syntax analyzer.

5) Construct the DAG for the following blocks.

a : = b * c, d : = b, e : = b*c, b : = e, f : = a + c, g : = f + d

6) Define :

a) dominators

b) natural loops.

3. A) Solve **any two** from following questions : **6**

1) For the following grammar construct Syntax Directed Definition (SDD).

S-> EN
E-> E + T | E – T | T
T->T*F | T/F | F
F-> (E) | digit
N-> ;

2) What is runtime storage ? Explain the runtime storage allocation in detail.

3) Explain compiler construction tools.

B) Write the Intermediate Representation like syntax tree and three address code for the following expression. **4**

(a-b)*(c+d)–(a+b)

4. Solve following questions : **10**

1) Check whether following grammar is LR(1) grammar or not.

S-> AaAb|BbBa, A->∈ B->∈

2) Why symbol table is used ? Explain symbol table with its operation.

3) What is three address code ? Explain implementation of three address code statements.

5. Solve following questions : **10**

1) Construct LL(1) parse table for following grammar :

S->aBDh B->cC C->bC|∈ D->EF E->g|∈ F->f|∈

2) Define Code optimization. Explain principle sources of code optimization.

3) Explain Activation Record. Draw the Activation Record for the factorial program.

____

SLR-SZ – 51

Seat No.	

Set **P**

B.Sc. (ECS) – III (Semester – VI) (New CGPA) Examination, 2017
COMPILER CONSTRUCTION (Paper – IV)

Day and Date : Friday, 3-11-2017 Max. Marks : 70
Time : 2.30 p.m. to 5.00 p.m.

Instructions : *1)* ***All*** *questions are* ***compulsory****.*
2) ***All*** *questions carry* ***equal*** *marks.*

1. Choose **correct** alternatives. **14**

1) The ________________ determines the semantic meaning of the source string.

a) Lexical analyzer b) Syntax analyzer
c) Semantic analyzer d) None of these

2) Once an intermediate-code program is partitioned into ________________ we represent the flow of control between them by a

a) Basic blocks and flow graphs
b) Flow graphs and basic blocks
c) Directed acyclic graph
d) None of these

3) The attributes that can be computed from the values of the attributes at the parent's of that node in the parse tree is called as

a) Inherited b) Synthesized
c) Both a and b d) None of these

4) ________________ will try different productions, backing up when a parse fails.

a) backtracking b) predictive
c) intermediate code d) back patching

P.T.O.

5) In ________________ a node in a flow graph dominates another if every path to the latter must go through the former.

a) inner loops b) reducible
c) pre-header d) dominators

6) In ________________ however, the position of an activation record for a procedure is not known until run time.

a) stack allocation b) heap allocation
c) static allocation d) both b and c

7) This parser uses ________________ derivation process.

a) Topdown parser b) Bottom up Parser
c) Either a or b d) Both a and b

8) ________________ begin at the source string and try to apply reductions to arrive at the start symbol.

a) LL parser b) LR parser

c) None of these d) All of above

9) All the items whose (period operator) are not at the leftmost end of the RHS of the rule

a) kernel items b) non kernel items

c) both a and b d) none of these

10) ________________ is a parser with single lookahead terminal.

a) SLR b) CLR c) LALR d) Shift reduce

11) The LR-parsing method is the most general ________________ shift-reduce parsing method.

a) backtracking b) non backtracking

c) both a and b d) none of these

12) ________________ engines that produce collections of routines for walking a parse tree and generating intermediate code.

a) Scanner generators b) Syntax-directed translation

c) Data-flow analysis d) Code-generator generators

13) For the grammar S->(L)|a L->L,S|S the First (S)=

a) {(,a} b) {(,),a} c) {),a} d) {(,)}

14) The pattern [a-zA-Z] [a-zA-Z0-9]* is used for representing

a) identifier b) keyword

c) operator d) number

2. Solve **any seven** from following questions. **14**

1) Define cross compilers.

2) Convert the following grammar in left factored form.
S->aAb|aBC|aAbG

3) Role of lexical analyzer.

4) Define Regular Definition.

5) Construct the DAG for the following string.
X:=-a*b+-a*b

6) Define the terms a) dominators d) natural loops

7) Explain Backtracking with example.

8) Design the dependency graph for the following grammar.
E->E1+E2
E->E1*E2

9) What is Global Register Allocation ?

2) What is runtime storage ? Explain the runtime storage allocation in detail.

3) Explain compiler construction tools.

B) Construct recursive descent parser for the following grammar. 4

E->num T

T->∗num T|ε

4. Solve **any two** from the following questions. 14

1) Check whether following grammar is Canonical LR(1) grammar or not.

S->AaAb|BbBa, A->ε , B->ε

2) Write the Intermediate Representation like postfix notation, syntax tree and three address code for the following expression

(a – b)∗(c + d) – (a + b)

3) Write a note on Peephole Optimization.

5. Solve **any two** from the following questions. 14

1) Construct LL(1) parse table for following grammar :

S->aBDh B->cC C->bC|ε D->EF E->g|ε F->f|ε

2) Define Code optimization. Explain principle sources of code optimization.

3) Explain Activation Record. Draw the Activation Record for the factorial program.

Set **P**

Seat No.

B.Sc. (E.C.S.) – III (Semester – VI) (CGPA Pattern) Examination, 2018
COMPUTER SCIENCE (Paper – IV)
Compiler Construction

Day and Date : Tuesday, 3-4-2018 Max. Marks : 70
Time : 2.30 p.m. to 5.00 p.m.

Instructions : *1)* ***All*** *questions are* ***compulsory****.*
2) Figures to the ***right*** *indicate* ***full*** *marks.*

1. Choose the correct alternative : **14**

1) In compilers generation of intermediate code based on an abstract machine model is useful because

A) Syntax-directed translations can be written for intermediate code generation

B) To generate code for real machines directly from high-level language programs is not possible

C) Portability of the front end of the compiler is enhanced

D) Implementation of lexical and syntax analysis is easier

2) We have the grammar $E \rightarrow E+n|E \times n|n$. The handles in the right-sentential form of the reduction for a sentence $n + n \times n$ are

A) $n, n + n$ and $n + n \times n$

B) $n, E + n$ and $E \times n$

C) $n, E + n$ and $E + E \times n$

D) $n, E + n$ and $E + n \times n$

3) The languages that need heap allocation in the runtime environment are

A) Those that use global variables

B) Those that use dynamic scoping

C) Those that support recursion

D) Those that allow dynamic data structure

P.T.O.

4) In some programming language, L denotes the set of letters and D denotes the set of digits. An identifier is permitted to be a letter followed by any number of letters or digits. The expression that defines an identifier is

A) $(L.D)^*$ B) $(L + D)^*$ C) $L(L.D)$ D) $L(L + D)^*$

5) Which one of the following statement is true ?

A) Canonical LR parser is more powerful than LALR parser

B) SLR parser is more powerful than LALR

C) LALR parser is more powerful than canonical LR parser

D) SLR parser, canonical LR parser and LALR parser all have the same power

6) Consider the following C program :

```
int main (){/*line1*/
int i, n;/*line 2*/
for (i=0,i
```

While creating the object module, the compiler's response about Line No..

A) Only syntax error B) No compilation error

C) Only lexical error D) Both lexical and syntax error

9) Which one of the following statements holds true for a bottom-up evaluation of syntax directed definition ?

A) Inherited attributes can always be evaluated

B) Inherited attributes can never be evaluated

C) Inherited attributes can be evaluated only if the definition is L-attributed

D) Inherited attributes can be evaluated only if the definition has synthesized attributes

10) For predictive parsing, the grammar $A \rightarrow AA \mid (A) \mid \epsilon$ is not suitable because

A) The grammar is right recursive

B) The grammar is left recursive

C) The grammar is ambiguous

D) The grammar is an operator grammar

11) Assuming that the input is scanned in left to right order, while parsing an input string the top-down parser use

A) Rightmost derivation

B) Leftmost derivation

C) Rightmost derivation that is traced out in reverse

D) Leftmost derivation that is traced out in reverse

12) ________ is a top-down parser.

A) Operator precedence parser B) An LALR (k) parser

C) An LR (k) parser D) Recursive descent parser

13) Why is the code optimizations are carried out on the intermediate code ?

A) Because for optimization information from the front end cannot be used

B) Because program is more accurately analyzed on intermediate code than on machine code

C) Because for optimization information from data flow analysis cannot be used

D) Because they enhance the portability of the compiler to the other target processor

14) In a compiler, when is the keyboards of a language are recognized ?

A) During the lexical analysis of a program

B) During parsing of the program

C) During the code generation

D) During the data flow analysis

2. Answer the following (**any seven**) : 14

   1) List the phases that constitute the front end of a compiler.
   2) What is meant by Handle and Handle Pruning ?
   3) Why lexical and syntax analyzers are separated out ?
   4) What is operator precedence parser ?
   5) What are the problems with top down parsing ?
   6) What is phrase level error recovery ?
   7) Mention the functions that are used in back-patching.
   8) What is a flow graph ?
   9) What is code motion ?

3. A) Answer the following (**any two**) : 10

   1) Consider the following Context Free Grammar G = ({S, A, B}, S, {a, b}, P) where P is

      S → AaAb

      S → Bb

      A → ∈

      B → ∈

      a) Compute the FIRST sets for A, B and S.
      b) Compute the FOLLOW sets for A, B and S.
      c) Is the CFG G LL(1) ? Justify ?

   2) Define string. Give commonly used string related terms with example.
   3) What are the types of Parser ? Give some common programming errors with example which can occur at different levels.

   B) Consider the expression a + a * (b – c) + (b – c) * d. 4

   a) Draw the Syntax Tree.
   b) Draw the DAG.
   c) Give the postfix notation for same.
   d) Give the code sequence for the same.

4. Answer the following (**any two**) : 14

   1) Explain an Activation Record.
   2) Construct a table-based LL(1) predictive parser for the following grammar :

      G = {bexpr, {bexpr, bterm, bfactor}, {not, or, and, (,), true, false}, P} with P given below.

      bexpr → bexpr or bterm | bterm

      bterm → bterm and bfactor | bfactor

      bfactor → not bfactor | (bexpr) |true| false

      For this grammar, answer the following questions :

      a) Remove left recursion from G.
      b) Left factor the resulting grammar in (a).
      c) Compute the FIRST and FOLLOW sets for non-terminals.
      d) Construct the LL parsing table.

   3) Explain the primary structure preserving transformations and algebraic transformations on basic block with example.

5. Answer the following (**any two**) : **14**

1) What is Shift-Reduce Parsing ? Consider the following grammar and input string. Parse the string using shift reduce parser. Show the content of the stack, input and action taken at each stage.

   $S \rightarrow aB \mid bA$

   $A \rightarrow bAA \mid aS \mid a$

   $B \rightarrow aBB \mid bS \mid b$

   Input string : aabbab

2) Explain in detail Loops in Flow Graphs.

3) What are the various methods of implementing three address statements ? Also give the types of th

Seat No.	

Set **P**

B.Sc. (ECS) – III (Semester – VI) (CGPA) Examination, 2018
COMPILER CONSTRUCTION (Paper – IV)

Day and Date : Thursday 1-11-2018 Max. Marks : 70

Time : 10.30 a.m. to 1.00 p.m.

Instructions : *1)* ***All*** *questions are* ***compulsory****.*
2) Figures to the ***right*** *place indicate* ***full*** *marks.*

1. Choose the correct alternative : **14**

1) The source program into a sequence of atomic unit called ________

a) Identifier b) Token

c) Keywords d) None

2) A ______________ reads the input one character at a time.

a) Lexical analyzer b) Parser

c) Symbol table d) None

3) A ______________ describes the hierarchical structure of programs.

a) Lexical analyzer b) Syntax tree

c) Grammar d) None

4) ______________ are data structures that hold information about identifiers.

a) Tokens b) Parser

c) Lexical Analyzer d) None

5) The set of words or strings of characters that match a given pattern is called a

a) Language b) Lexeme

c) Regular definition d) None

6) A ______________ takes as input tokens from the lexical analyzer.

a) Parser b) Syntax directed translation

c) Code generation d) None

7) A ______________ attribute at a parse tree node is computed from attributes at its children.

a) Inherited b) Synthesized

c) Both d) None

8) Procedure calls and returns are usually managed by a run time stack called the

a) Heap b) Stack Allocation

c) Control Stack d) None

9) ______________ is the final phase of a compiler.

a) Code generation b) Run time

c) Both d) None

10) A ___________ is a graphical representation of a program in which the nodes of the graph are basic block and edges show flow.

a) Flow graph b) Loop

c) DAG d) None

11) Three address code is sequence of statements of Z = op Y.

a) True b) False

12) Code generator is not phase of compiler.

a) True b) False

13) By value is one type of parameter passing technique.

a) True b) False

4) What is L-attributed definition ?

5) Explain Stack Allocation.

6) What is the definition of Backpatching ?

7) Explain flow graph.

8) What is copy propagation ?

9) Explain left recursive.

3. A) Attempt **any two** of the following : **10**

1) Explain Predictive parser. Construct the following grammar.

$S \rightarrow AB | \epsilon$

$A \rightarrow aAB | \epsilon$

$B \rightarrow bA$

2) Write a note on Parameter passing.

3) Explain loops in flow graph with example.

B) Explain Input Buffering. **4**

4. Attempt **any two** of the following : **14**

1) Explain phases of a compiler.

2) Write a note on source language issues.

3) Design and implementation of lexical analyzer explain with suitable example.

5. Attempt **any two** of the following : **14**

1) Explain Run time storage management.

2) What is Bottom-up parser ? How to implement shift reduce parser ? Solve the given example using Handle pruning.

$S \rightarrow xPy$

$P \rightarrow xP|Qy$

$Q \rightarrow y$

3) Explain construction of Syntax tree with example.

____

Seat No.	

B.Sc. (ECS) – III (Semester – VI) Examination, 2015
COMPUTER SCIENCE (Paper – IV)
Compiler Construction

Day and Date : Monday, 16-11-2015 Max. Marks : 50
Time : 2.30 p.m. to 4.30 p.m.

Instructions : 1) *All questions are* **compulsory.**
2) *Figures to the* **right** *indicates* **full** *marks.*

1. Choose the correct alternative : 10

1) The ____________ resolves external memory addresses, where the codes in one file may refer to a location in another file.

a) loader b) pre-processor
c) linker d) translator

2) A ____________ is a description of the form that the lexemes of a token may take.

a) token b) attribute value
c) lexeme d) pattern

3) ____________ errors include type mismatches between operators and operands.

a) Semantic b) Syntactic c) Lexical d) Logical

4) A ____________ is a context-free grammar together with attributes and rules.

a) Syntax-Directed Definition (SDD)
b) S-attributed SDD
c) L-attributed SDD
d) All of above

5) Procedure calls and returns are usually managed by a runtime stack called ____________

a) Control stack b) Garbage collection
c) Activation tree d) Activation record

10) ____________ parsing is a top-down method of syntax analysis in which a set of recursive procedure is used to process the input.

a) Recursive-descent b) Predictive

c) Operator-precedence d) All of above

2. Answer **any five** of the following : **10**

1) What is the role of syntax analyzer ?

2) What is the compilation process ?

3) Give the advantage and disadvantage of register allocation.

4) Define :

a) Synthesized attribute

b) Inherited attribute.

5) What is the difference between actual parameter and formal parameter ?

6) Why the analysis portion of a compiler is normally separated into lexical analysis and syntax analysis ?

3. A) Answer **any two** of the following : **6**

1) Explain natural loop with example.

2) Give the common three-address statements used in the intermediate languages.

3) When the lexical errors are occurred ? How to recover it ?

B) Explain backpatching. **4**

4. Answer **any two** of the following : **10**

1) Discuss phases of compiler.

2) Explain inherited attribute on the parse stack with example.

3) What is left recursion ? How to eliminate it ? Explain with example.

5. Answer **any two** of the following : **10**

1) Discuss an activation record.

2) Explain an issues in the design of code generator.

3) Consider the grammar,

```
stmt →  if expr then stmt
      | if expr then stmt clse stmt
      | other
```

where ‘other’ stands for any other statement.

check whether the given grammar is ambiguous or not; if found ambiguous, remove the ambiguity and write the equivalent unambiguous grammar.

SLR-O – 64

Seat No.	

B.Sc. (ECS) – III (Semester – VI) Examination, 2015
COMPUTER SCIENCE (Paper – IV)
Compiler Construction

Day and Date : Thursday, 23-4-2015 Max. Marks : 50

Time : 11.00 a.m. to 1.00 p.m.

Instructions : *1)* ***All*** *questions are* ***compulsory.***

2) Figures to the ***right*** *indicate* ***full*** *marks.*

1. Choose the correct alternatives : **10**

1) The task of collecting the source program is sometimes entrusted to a separate program, called a

a) linker b) loader

c) compiler d) pre-processor

2) ________, in which the look ahead symbol unambiguously determines the flow of control through the procedure body for each non-terminal.

a) Predictive b) Recursive-descent

c) Operator-precedence d) None

3) A set of non-terminals, sometimes called

a) Semantic variables b) Syntactic variable

c) Static variable d) All above

4) Many loops have ________ variables, variables that take on a linear sequence of values each time around the loop.

a) induction b) static c) syntactic d) semantic

5) A ________ machine is an interpreter for a byte code intermediate language produced by languages such as Java and C#.

a) Server b) Client c) Virtual d) All above

6) Back patching is a technique for generating code for ________ expressions and statements in one pass.

a) boolean b) arithmetic c) logical d) none

P.T.O.

2. Answer **any five** of the following : 10

   1) List out the some useful compiler construction tools.

   2) What is the difference between panic mode recovery and phrase-level recovery ?

   3) Give several methods for evaluating semantic rules.

   4) What is control stack ?

   5) What is the role of lexical analyzer ?

   6) Define :

      a) Basic block b) Flow graph

3. A) Answer **any two** of the following : 6

      1) Explain dominator and immediate dominator with example.

      2) Define the following with example
         1) Token
         2) Pattern
         3) Lexeme

      3) Explain Boolean expression.

   B) What is three address code ? 4

      Consider, the expression

      a : = b* – c + b * – c

      Give the code for syntax tree and code for dag of expression.

4. Answer **any two** of the following : 10

   1) What is dag ? Construct the dag for the expression

      x – x * (y + z) – (y + z) * w

      Also give the instructions for the same.

   2) Define compiler. Explain the different types of compiler.

   3) Consider the expression,

      E → E + E/E * E/id

      check whether the above grammar is ambiguous or not; if found ambiguous, remove the ambiguity and write an equivalent unambiguous grammar.

5. Answer **any two** of the following : 10

   1) Explain structure-preserving transformation basic block.

   2) Give the different storage-allocation strategies. Explain any two.

   3) Explain the notational conventions with regard to grammar.

____