CS6660-COMPILER DESIGN

UNIT I –INTRODUCTION TO COMPILER

PART-A

1. Define compiler.(Nov/Dec09)
2. What are the two types of analysis of the source programs by compiler? (April/May 10)
3. What is a language processing system?(Nov/Dec 2008)
4. Describe the error recovery schemes in a lexical phase of a compiler.(April/May 2015)
5. Illustrate diagrammatically how a language is processed. MAY/JUNE-2016
6. What are the phases of a compiler? APR-2018 DEC-2018
7. What is a symbol table? (Nov/Dec 2007) (Nov/Dec 2016)
8. Mention some of the cousins of a compiler. [May/June 2012]
9. What is grouping of phases?[Nov/Dec 2013][May/June-14]
10. Define compiler-compiler.
11. What are compiler construction tools? (Nov/Dec 04,05) (Nov/Dec 2016)
12. Define translator.
13. What do you mean by assembler?
14. What do you mean by preprocessor? (May/June07)
15. Define linker.
16. Define macro processor.
17. Define loader.
18. Differentiate compiler and interpreter. (May/June2008)
19. Define pretty printers?
20. Explain with diagram how a statement is compiled. DEC-2018
21. Define Interpreters. [April/May 2011]
22. What are the functions of preprocessor? [Nov/Dec 2007]
23. Define concrete and abstract syntax with example. [May/June 2009]
24. What are the phases included in front end of a compiler? What does the front end produce? May -11
25. Define the term cross compiler. Nov/Dec 05
26. What are the factors affecting number of passes in compiler?
27. What are the Error Recovery actions in Lexical Analyzer? (Nov/Dec 2008) APR-2018
28. State any two reasons as to why phases of compiler should be grouped. ? (May-June 2014)
29. Why is buffering used in lexical analysis? What are the commonly used buffering methods? (May-June 2014)
30. What are the issues in lexical analysis?(May/June2007)

PART-B

1. What is a compiler? Explain about the different types of software tools available for analyzing. (8)(June-14)
2. Explain language processing system with neat diagram. (8) May/June 2016 APR-2018
3. Describe the various phases of compiler and trace it with the program segment (position := initial + rate*60). NOV/DEC-2016 APR-2018 (13) DEC-2018
4. Explain various Errors encountered in different phases of compiler. (4) (8) May/June2016
5. Explain the need for grouping of phases. (4) (8) (May/June 2016) (Nov/Dec 2016)
6. Explain construction tools in compiler.(8) Nov/Dec2016] DEC-2018
7. Explain briefly about Programming Language Basics.(13)

UNIT-II (LEXICAL ANALYSIS)

PART-A

1. Write a regular expression for an identifier and whitespace. (Nov/Dec 2013)
2. List the various error recovery strategies for a lexical analysis. [May June 2012]
3. What are roles and tasks of a lexical analyzer? [Nov/Dec 2011]
4. Define terminal symbol and non-terminal symbol.
5. What do you mean by LEX?
6. What are the reasons for separating the analysis phase to lexical analysis and parser? [May/June 2013] [May/June 2009]
7. Write down the possible error recovery actions taken by lexical analyzer.
8. Define character class and language
9. What are the operations supported by a string?
10. What do you mean by subsequence?
11. What is the purpose of the following notations ✳ , + and ?
12. Define Look ahead operator.
13. Define NFA. May/June-14
14. Define DFA. May/June-14
15. Define concrete and abstract syntax with example. [May/June 2009]
16. Give the transition diagram for an identifier. [Nov/Dec 2011]
17. Define LEXEME.(May/June 2014)
18. Define tokens, patterns and lexemes. (Nov/Dec 2016)
19. List the operations on languages. May/June2016
20. Write a grammar for branching statements. May/June2016

PART-B

1. Explain in detail about the role of the lexical analyzer with the possible error recovery actions.(10)
2. Differentiate between lexeme, token and pattern. (6) May/June2016
3. What are the issues in lexical analysis? (4) May/June2016
4. Discuss how finite automata are used to represent tokens and perform lexical analysis with examples. (10)(Nov/Dec 2016)
5. Write notes on regular expressions. (6) May/June2016
6. Explain briefly about converting of an NFA into a DFA(8).
7. Write an algorithm for minimizing the number of states of a DFA.(8) Nov/Dec 2016
8. Convert the regular expression (a+b)*abb into NFA – ε and find the equivalent minimum state DFA. (10)[NOV/DEC 2008]
9. Discuss about LEX tool.(8) (APR-2018)
10. Discuss about Lexical errors.(4)
11. Explain briefly about Design of Lexical Analyzer for a sample Language.(13)
12. Write notes on regular expression to NFA. Construct Regular expression to NFA for the sentence (a/b)*a. (10) May/June 2016
13. Construct DFA to recognize the language (a/b)*ab. (7) (May/June 2016)

UNIT-3 SYNTAX ANALYSIS

PART-A

1. What are different error recovery strategies?
2. Define an ambiguous grammar (May/Jun -2012)
3. What is left recursion? How it is eliminated?
4. Define LL (1) grammar.
5. Define Handle. Apr/May – 2005 Nov\ Dec – 2004
6. What is handle pruning? (Apr/May -2011)
7. What are viable prefixes? Nov\ Dec – 2004
8. What are the benefits of LR parsers?

9. What is dangling reference? May/Jun -2012
10. Mention the two rules for type checking (Nov/Dec – 2011)
11. What is the signification of look ahead in LR (1) items (Apr/May – 2010)
12. Differentiate SLR parser from LALR parser (Nov/Dec – 2010)
13. List the factors to be considered for top – down parsing (May/ Jun – 2009)
14. Differentiate Top down approach from Bottom Up approach to parsing with an example (Nov/Dec – 2010)
15. What is phrase level error recovery?(May/ Jun – 2006)
16. What do you mean by viable prefixes? (Nov\ Dec – 2004)
17. Derive the string and construct a syntax tree for the input string ceaedbe using the grammar S -> SaA | A,A ->AbB | B, B -> c Sd | e (May/ Jun – 2009)
18. Define LR(0) items (Apr/May – 2004)
19. Give examples for static check.(May/June-2013)
20. Write the algorithm for FIRST and follow in parser.(May/June 2016)
21. Construct a parse tree for –(id+id). (Nov/Dec 2016)

PART-B

1. Explain the role of parser. (Nov\ Dec – 2007)(6marks)
2. Explain the error recovery strategies in syntax analysis. (6 Marks)(Apr/ May 2011)
3. Explain Context free grammar with examples.(10) May/June 2016
4. Explain the Ambiguity or Ambiguous grammar G : E → E+E | E*E | (E) | - E | id . (Nov/Dec 2016)(6 marks)
5. Construct parse tree for the input string w=cad using top down parser.(6) (Nov/Dec 2016)
6. Write an algorithm for Non recursive predictive parsing. (6) May/June 2016
7. Consider Parsing table for the grammar and find moves made by predictive parser on input id+id*id and find FIRST and FOLLOW. (13) (Nov/Dec 2016)
   a. E → E+T |T
   b. T → T*F |F
      F→ (E) |id
8. Give an algorithm for finding the FIRST and Follow positions for a given non terminal. (Nov\ Dec – 2005)(8)
9. Explain the LR parsing algorithm.(6)(Apr/May – 2005)
10. Construct the predictive parser for the following grammar (Apr/May – 2010) (Apr/May – 2005)
    a. S → a | ↑ | (T)
    b. T → T, S | S
    c. Write down the necessary algorithms and define FIRST and FOLLOW. Show the behavior of the parser in the sentences:
       i. (a, (a,a))
       ii. (((a,a), ↑ , (a), a)
11. What is Shift reducer parser? Explain in detail the conflict that may occurring Shift reducer parsing. (8)May/Jun -2012(Nov\ Dec – 2005)
12. Check whether the following grammar is SLR (1) or not . Explain Your answer with reasons (8 Marks) Apr/May – 2004
13. Write an algorithm for the construction of LR(1) or CLR items for grammar G. Nov/Dec – 2010(May/June-2013)
14. Construct SLR parsing for the following grammar: (Nov/Dec 2016)
    i. G : E → E + T | T
    ii. T → T * F | F
    iii. F→ (E) | id
15. Check whether the following grammar is LL(1) grammar (Apr/May – 2005)
    i. S→iEtS \ iEtSeS \ a
    ii. E→b
16. Briefly explain error recovery in LR parsing (Nov\ Dec – 2005)

17. Design an LALR parser for the following grammar and parse the input id=id.(16)(Nov/Dec 2013)
18. Explain briefly about the Parser Generator Yacc.
19. Construct LR (0) parsing table for the given grammar (Nov/Dec – 2010)

$E \to E*B$
$E \to E+B$
$E \to B$
$B \to 0$
$B \to 1$

UNIT-IV (SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT)

PART-A

1. Define S-attribute and L-attribute definitions.
2. What do you mean by syntax directed translation?
3. What are the functions used to create the nodes of syntax trees?
4. Define an attribute. Give the types of an attribute?
5. What are the fields in an activation record?(Nov/Dec 2013)
6. Define register descriptor.
7. What are the storage allocation strategies during runtime?(Nov/Dec 2016
8. What is Activation tree?
9. What is Register Allocation?
10. What are the limitations of Static Allocation?
11. What are the limitations of Static Allocation? (Apr/May -2011)
12. Give two examples for each of top down parser and bottom up parser? (Apr/May – 2010)
13. What are the issues in static allocation? (Nov/Dec-09)
14. Define symbol table. [May/June-14]
15. What is meant by coercion? (Nov/Dec 2013)
16. Give examples for static check(May/June-2013)
17. Mention the two rules for type checking (Nov/Dec – 2011)
18. What is SDD?
19. Define annotated parse tree.
20. What is dangling reference? May/Jun -2012 ,May/Jun -2016
21. Define Type checking.
22. Define type systems.
23. What are the data structures used for designing symbol table?

PART-B

1. Explain about Syntax-Directed Definitions.(8)
2. Explain about the construction of syntax tree (or) Applications of Syntax-Directed Translation.(13)
3. Explain in detail about Bottom-Up Evaluation o f S-Attributed Definitions.(8)
4. Construction of a predictive syntax-directed translator.(16)
5. Explain the details about Type Checking with necessary diagram.(16)(Nov/Dec 2016)
6. Explain the detail about the specification of a simple type checker (16 Marks)(May/Jun -2012)
7. Explain the run time environment.(8)
8. What are the Storage Allocation Strategies? Explain (16 Marks) (Nov /Dec 2011)(May/June-2013) NOV/DEC-2018
9. Distinguish between the source test of a procedure and its activation at run time (8 Marks)(Apr/May 2011)
10. Explain in detail about dynamic storage allocation techniques.(8)
11. Explain about parameter passing. (8) NOV/DEC-2018 APR/MAY-2018
12. Explain about activation record. (7) APR/MAY-2018

UNIT-V (CODE OPTIMIZATION AND CODE GENERATION)

PART-A

1. What is meant by optimization?
2. List down the criteria for code improving transformation? Nov/Dec -2011
3. List the function preserving transformation Nov/Dec -2011
4. What is meant by loop optimization?
5. Define basic block? Nov/Dec -2004
6. What are the factors influencing the optimization?
7. What is dead code?
8. What is code motion? Apr/May -2004
9. What is loop optimization?(Apr/May -2011)
10. List out two properties of reducible flow graph (May/Jun-2012)
11. What is the use of algebraic identifies in optimizations of basic blocks? (May/Jun-2012)
12. Define dead-code elimination. (Apr/May-2011)
13. When does dangling reference occur? (Nov/Dec -2011)(May/June 2016)
14. How Would You Map Names To Values? (Nov/Dec 2010)
15. List the characteristics of peephole optimization. (Nov/Dec 2010) (Nov/Dec 2016)
16. List out the primary structure preserving transformations on basic block. (Apr/May -2011)
17. Define dead code elimination (Apr/May -2011) (Nov/Dec 2010
18. What are the fields available in activations record? Apr/May -2010
19. What is meant by constant folding (May/June-2013)
20. What is meant by copy restore? May/Jun -2009
21. What is a peehole optimization? (Nov/Dec 2010)
22. What are the properties of optimizing compilers? May/June-2016
23. Identify the constructs for optimization in basic block. (Nov/Dec-2016)

PART-B

1. Explain the principal sources of optimization in details (8 marks) May/Jun-2012)(May/June-2013)
2. Explain the DAG Representation For Basic Block with an Example.(May/June-2013)(10)
3. Construct the DAG for the following basic block. (10)(Apr/May-12) Nov/Dec 13)

   T_1:= A+B;
   T_2:= C+D;
   T_3:= E-T2;
   T_4:= $T_1 - T_3$

4. Construct DAG for the following basic block (8 marks) APR/MAY-2010)

   d := b * c
   e :=a + b
   b := b * c
   a := e – d

5. Explain in detail optimization of basic blocks with an example (8 marks Nov/Dec-2014)
6. Explain the global data flow analysis. [Nov/Dec 2013] [Nov/Dec 2016]
7. How to trace Data-flow analysis of structured programs .(6 marks) [May/June-14]
8. Write in detail about the issues in the design of a code generator (8)(Nov/Dec-16)
9. Briefly explain about simple code generator .(13)(May/June-2016)
10. Write an algorithm for constructing natural loop of a back edge.(8)Nov/Dec-2016.