**Reg. No. :** | | | | | | | | | | |

## Question Paper Code: 41261

B.E. / B.Tech. DEGREE EXAMINATION, MAY 2017

Sixth Semester

Computer Science and Engineering

14UCS601 - PRINCIPLES OF COMPILER DESIGN

(Regulation 2014)

Duration: Three hours Maximum: 100 Marks

Answer ALL Questions

PART A - (10 x 1 = 10 Marks)

1. The linker?

   (a) is same as the loader

   (b) is required to create a load module

   (c) is always used before programs are executed

   (d) none of these

2. Compiler should report the presence of _____ in the source program, in translation process.

   (a) Classes (b) Objects (c) Errors (d) Text

3. What is the action of parsing the source program into proper syntactic classes?

   (a) Lexical analysis (b) Syntax analysis
   (c) General syntax analysis (d) Interpretation analysis

4. _____ is considered as a sequence of characters in a token.

   (a) Texeme (b) Pattern (c) Lexeme (d) Mexeme

5. A grammar that produces more than one parse tree for some sentence is called as

(a) Ambiguous (b) Unambiguous (c) Regular (d) All the above

6. Which one of the following statement is false for the SLR (1) and LALR (1) parsing tables for a context free grammar?

(a) The reduce entries in both the tables may be different
(b) The error entries in both the tables may be different
(c) The go to part of both tables may be different
(d) The shift entries in both the tables may be identical

7. 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

8. When is the type checking usually done?

(a) During syntax directed translation (b) During lexical analysis
(c) During code optimization (d) During syntax analysis

9. Why is the code optimizations are carried out on the intermediate code?

(a) Because for optimization information from the front end cannot be use
(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

10. Object code form of code generation is represent by

(a) Absolute Code (b) Re locatable machine code
(c) Assembler Code (d) All the above

PART - B (5 x 2 = 10 Marks)

11. What is meant by semantic analysis?

12. Describe the error recovery schemas in lexical analyzer.

13. What is meant by left factoring?

14. Construct a syntax tree and DAG for k:=k+5?

15. What is the use of Next-use information?

PART - C (5 x 16 = 80 Marks)

16. (a) Explain in detail the process of compilation. Illustrate the output of each phase of compilation for the input : a = ( b + c ) * ( b + c ) * 2. (16)

Or

(b) (i) Mention any four compiler construction tools with their benefits and drawbacks. (8)

(ii) Describe the need for grouping of phases of compiler. (8)

17. (a) Prove that the following two regular expressions are equivalent by showing that the minimum state DFA's are same :

(i) ( a / b ) *

(ii) ( a * / b * ) * (16)

Or

(b) (i) Write in detail about the role of Lexical analyzer with the possible error recovery actions. (8)

(ii) Elaborate specification of tokens. (8)

18. (a) Find the LALR for the given grammar and parse the sentence ( a + b ) * c

$$E \rightarrow E + T \mid T ,$$

$$T \rightarrow T * F \mid F ,$$

$$F \rightarrow ( E ) / id.$$

(16)

Or

(b) Consider the following grammar S → AS | b, A → SA | a. Show the SLR parse table for the grammar. Show the actions of the parser for the input string "abab". (16)

19. (a) List out the various storage allocation strategies. (16)

Or

(b) Briefly explain about the type checking and its properties. (16)

20. (a) (i) Draw the DAG for the following three address code.

d = b * c  e = a + b  b = b * c  a = e - d. (8)

(ii) List out the issues in design of a code generator and explain it. (8)

Or

(b) Illustrate optimization of basic blocks with an example. (16)