Register No: ……

**B.E. DEGREE EXAMINATIONS: APRIL/MAY 2011**

Sixth Semester

**COMPUTER SCIENCE AND ENGINEERING**

U07IT505: Principles of Compiler Design

**Time: Three hours** **Maximum Marks: 100**

**Answer all questions:-**

**PART A (10 x 1 = 10 Marks)**

1. Hierarchical analysis is called as _____

a) Semantic analysis b) Syntactic Analysis c) Lexical analysis d) Code analysis

2._____special character is not a part of the source program

a) Blank space b) Back space c) Literal d) Sentinel

3. A grammar that produces more than one parse tree is called as _____

a) Unique b) Ambiguous c) Context Free Grammar d) Regular Expression

4. General form of top-down parsing is known as_____

a) Predictive parsing b) LR parsing c) Recursive Descent parsing d) LL parsing

5._____ is a linearized representation of a syntax tree

a) Infix b) Prefix c) Regular expression d) Postfix

6. Subsequent filling of labels is called_____

a) Back tracking b) Back patching c) Left factoring d) Calling sequences

7._____ data structure is used for implementing transformations on basic blocks

a) Graph b) Syntax tree c) DAG d) Link list

8._____ is a small moving window on the target program

a) Machine idioms b) Peephole c) Peephole Optimization d) Code motion

9. A compiler use a _____ to keep track of scope and binding information about names.

a) Symbol Table b) Hash table c) Lookup table d) Name table

10._____ transformation can be implemented by constructing a dag for a basic block

a) Structure preserving b) Function preserving

c) Local Transformation d) Global transformation

**PART B (10 x 2 = 20 Marks)**

11. Write the need of static checkers analysis tool.

12. Draw NFA that recognizes the language $(a/b)^*$ abb.

13. Why use regular expressions to define the Lexical syntax of a language?

14. List the disadvantages of operator precedence parsing.

15. Define Backpatching.

16. Translate the arithmetic expression a=b * -c + b * -c into three address code.

17. List the issues faced in the design of a code generator.

18. State the applications of DAGS.

19. State code motion.

20. Illustrate dangling references.

**PART C (5 x 14 = 70 Marks)**

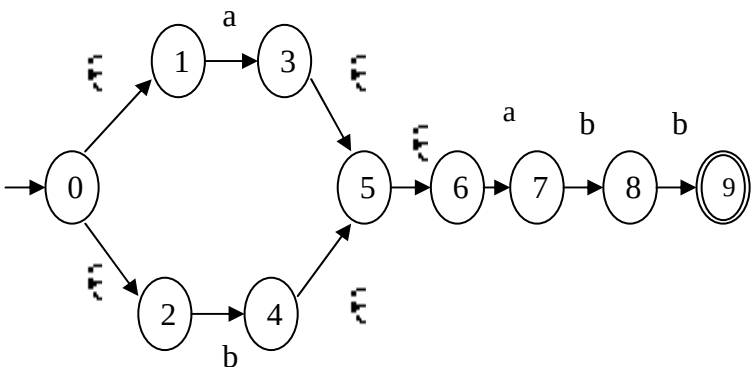
21. a) (i) Describe the phases of a compiler with neat diagram. (10)

(ii) Explain different steps involved in language processing system. (4)

**(OR)**

b) (i) Illustrate an algorithm to construct DFA from NFA. (7)

(ii) Convert the following NFA into corresponding DFA. (7)




22. a) (i) Explain the algorithm to construct a predictive parsing table. (7)

(ii) Construct predictive parsing table for the following grammar (7)

$E \longrightarrow E+T \mid T$

$T \longrightarrow T*F \mid F$

$F \longrightarrow (E) \mid id$

**(OR)**

b) (i) Write an algorithm to eliminate left recursion from a grammar with a suitable example.

(ii) Explain shift-reduce parsing with an example.

23. a) Write a translation scheme to produce three address code for assignment statement.

**(OR)**

b) Describe the translation scheme for Boolean expression.

24. a) Illustrate how to manage activation records at run time.

**(OR)**

b) Explain how to improve the quality of the intermediate representation of a target code.

25. a) Write different code optimization techniques.

**(OR)**

b) Explain different storage allocation strategy used in run time environment.

**************