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

# T 3185

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2008.

Sixth Semester

(Regulation 2004)

Computer Science and Engineering

CS 1352 — PRINCIPLES OF COMPILER DESIGN

(Common to B.E. (Part-Time) Fifth Semester Regulation 2005)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — ($10 \times 2 = 20$ marks)

1. Differentiate compiler and interpreter.

2. Write short notes on buffer pair.

3. Construct a parse tree of $(a+b)^*c$ for the grammar $E \rightarrow E+E/E^*E/(E)/id$.

4. Eliminate immediate left recursion for the following grammar $E \rightarrow E+T/T, T \rightarrow T^*F/F, F \rightarrow (E)/id$.

5. Write short notes on global data flow analysis.

6. Define back patching with an example.

7. Give syntax directed translation for the following statement Call p1 (int a, int b).

8. How can you find the leaders in basic block?

9. Define Code motion.

10. Define basic block and flow graph.

PART B — (5 × 16 = 80 marks)

11. (a) (i) Explain the phases of compiler, with the neat schematic. (12)

(ii) Write short notes on compiler construction tools. (4)

Or

(b) (i) Explain grouping of phases. (8)

(ii) Explain specification of tokens. (8)

12. (a) Find the SLR parsing table for the given grammar and parse the sentence $(a+b)^*c$ $E \rightarrow E+E/E^*E/(E)/id$ .

Or

(b) Find the predictive parser for the given grammar and parse the sentence $(a+b)^*c$ $E \rightarrow E+T/T, T \rightarrow T^*F/F, F \rightarrow (E)/id$ .

13. (a) Generate intermediate code for the following code segment along with the required syntax directed translation scheme : (8)

(i) if $(a > b)$

$x = a + b$

else

$x = a-b$

where a & $x$ are of real and $b$ of int type data

(ii) int $a$, $b$ ; (8)

float $c$ ;

$a$ = 10 ;

swith $(a)$

{case 10 : $c$ = 1 ;

case 20 : c = 2 ;

}

Or

(b) (i) Generate intermediate code for the following code segment along with the required syntax directed translation scheme : (8)

$i$ = 1 ; $s$ = 0 ;

while $(i <= 10)$

$s = s + a[i]$ [i] [i]

$i = i + 1$

(ii) Write short notes on back-patching. (8)

14. (a) (i) Explain the various issues in the design of code generation. (6)

(ii) Explain code generation phase with simple code generation algorithm. (10)

Or

(b) (i) Generate DAG representation of the following code and list out the applications of DAG representation : (12)

```
i = 1 ; s = 0 ;
    while (i <= 10)
    s = s + a [i] [i]
    i = i + 1
```

(ii) Write short notes on next-use information with suitable example. (4)

15. (a) (i) Explain – principle sources of optimization. (8)

(ii) Write short notes on : (8)

(1) Storage organization

(2) Parameter passing.

Or

(b) (i) Optimize the following code using various optimization technique : (12)

```
i = 1 ; s = 0 ;
for (i = 1 ; i <= 3 ; i++)
    for (j = 1 ; j <= 3 ; j++)
    c[i][j] = c[i][j] + a[i][j] + b[i][j]
```

(ii) Write short notes on access to non-local names. (4)