Total No. of Questions :8]

SEAT No. :

P3637

[Total No. of Pages :3

[4959] - 1126

B.E. (Information Technology)

C - MODERN COMPILERS

(End - Sem - Semester - I) (Elective - I) (2012 Pattern)

Time : 2½ Hours] *[Max. Marks :70*

Instructions to the candidates:

1) ***Answer Q1 or Q2, Q3 or Q4, Q5 or Q6, Q7 or Q8.***
2) ***Figures to the right indicate full marks.***
3) ***Assume suitable data if necessary.***
4) ***Neat diagrams must be drawn wherever necessary.***

Q1) a) Explain with a neat diagram the two layers of abstraction with respect to Tiger compiler. **[6]**

b) Draw control - flow graph for the given code. Find the live ranges of a, b, c. **[6]**

```
a = 0
L1 : b=a+1
c  =c+b
a=b*2
if a <N goto L1
return c
```

c) What is incremental garbage collection? Explain Baker's algorithm and comment on its cost of collection. **[8]**

OR

P.T.O.

Q2) a) When we say that a "veriable escapes"? For each variable a, b, c, d in the given program, find whether itescape and why? **[6]**

```
int f(int a, int *b)
{     int c[3], d;
      d=a+1;
      b=g(c, &b);
      return c[1]+b;
}
```

b) Define canonical trees. Describe any two identities for transformation on ESEQ. **[6]**

c) Explain copying garbage collection with a neat diagram. Write Cheney's algorithm and comment on its cost. **[8]**

Q3) a) What are the facilities for testing class membership in Java? Explain type coercions and type cases in brief. **[6]**

b) What is Closure? How it can be implemented using Heap-allocation? **[6]**

c) What is meant by private field in programming language? What are various ways to support it in programming language? **[6]**

OR

Q4) a) Define inline expansion. Explain the rules for inline expansion. **[6]**

b) Explain different techniques for optimization of lazy functional programming. **[6]**

c) Explain strictness analysis. **[6]**

Q5) a) Explain Inter - procedural data-flow analysis in brief. Describe different functions for flow-insensitive side effect analysis. **[8]**

b) What are possible caches in a system? Describe different approaches for instruction-cache optimization. **[8]**

OR