

Total No. of Questions : 10]

SEAT No. :

P3848

[5561]-276

[Total No. of Pages : 3

B.E. (Computer Engineering)

PRINCIPLES OF MODERN COMPILER DESIGN

(2012 Course) (410442) (Semester - I)

Time : 2½ Hours]

[Max. Marks : 70

Instructions to the candidates:

- 1) *Attempt Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8, Q.9 or Q.10.*
- 2) *Neat diagrams must be drawn wherever necessary.*
- 3) *Figures to the right indicate full marks.*
- 4) *Assume suitable data, if necessary.*

Q1) a) Draw and explain model of LR parser. **[4]**

b) Find FIRST and FOLLOW of the following : **[6]**

$S \rightarrow v \mid XUW \mid UV \mid uWV$

$U \rightarrow w \mid WvX \mid SUVW \mid SWX \mid \epsilon$

$V \rightarrow x \mid wX \mid \epsilon$

$W \rightarrow w \mid UXW$

$X \rightarrow yz \mid SuW \mid SW$

OR

Q2) a) Generate annotated parse tree for following expression : $a*b-c/e + f$. **[4]**

b) Construct LALR items for following grammar : **[6]**

$S \rightarrow AA, A \rightarrow aA|b.$

Q3) a) What is top down parsing? What are the problems with top down parsing? **[4]**

b) Write the quadruple, triple, indirect triple for the statement **[6]**

$a: = b * -c + b * -c.$

OR

Q4) a) Write short note on type checking & type conversion. **[4]**

b) Generate three address code for following code, **[6]**

While (a – b) do

If(cod) then

x = y + 2

else

x = y – 2.

P.T.O.

- Q5) a)** Write a note on a simple code generator. **[4]**
b) What are the principle sources of optimization? Give example of each. **[6]**
c) Write 3-Address code for following code and then perform optimization technique **[8]**

```

for (i = 0; i <= n; i++)
{
    for (j = 0; j < n; i++)
    {
        c[i, j] = 0;
    }
}

for (i = 0; i <= n; i++)
{
    for (j = 0; j <= n; i++)
    {
        for (k = 0; k <= n; k++)
        {
            c[i, j] = c[i, j] + a[i, k]*b[k, j];
        }
    }
}

```

OR

- Q6) a)** Define induction variable. Give examples. **[4]**
b) Explain the use of algebraic transformations with an example. **[6]**
c) Optimize and develop this code by eliminating common sub expression, performing reduction in strength on induction variables and eliminating all the induction variable you can **[8]**

```

t6 = 4*i           B2 : i = i + 1
x = a[t6]         t2 = 4*i
t7 = 4*I         t3 = a[t2]
t8 = 4*j         if t3 < a[t1] goto B2
t9 = a[t8]
a[t7] = t9
t10 = 4*j
a[t10] = x
goto B2.

```

- Q7)** a) Explain identification w.r.t. context handling. [6]
b) What are different types of routine. Explain each. [6]
c) List and Explain the Key features of functional programming language. [4]

OR

- Q8)** a) Explain Object Oriented Source language Issues. [6]
b) Write short note on Java CC. [6]
c) Give which aspect of Haskell corresponds to which compiler phase. [4]

- Q9)** a) Explain message passing and its issues related to parallel programming model. [6]
b) Describe the object replication, migration and location w.r.t. to Object Oriented languages. [6]
c) Give details of Just In Time compiler. [4]

OR

- Q10)** a) Differentiate between : [6]
i) Parallel and distributed systems.
ii) Multicomputers and multiprocessors.
b) Describe the compilation framework LLVM. [6]
c) What is interpreter? Explain JVM as an interpreter. [4]

