Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

M.E –IIst SEMESTER–EXAMINATION – JULY- 2012

Subject code: 1720202 **Date: 09/07/2012**

Subject Name: Design of Language Processors

Time: 10:30 am – 13:00 pm **Total Marks: 70**

Instructions:

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**

Q.1 **(a)** Consider following statement : **07**

z = a + b * c * a;

Explain translational output of above statement after each phase of the compiler. Where, a and b are of type int and c is float. Assume appropriate data type for z.

(b) Explain: symbol table, hashing & rehashing functions and heap. **07**

Q.2 **(a)** Explain activation record and various parameter passing techniques. **07**

(b) Construct NFA using Thompson's notation & construct DFA from it for following regular expression: a* b* (a | c) c*c# **07**

OR

(b) Construct DFA for following regular expression with out constructing NFA and then minimize it: (a | b)* a*b*abc# **07**

Q.3 **(a)** Construct an operator precedence graph and parse table for following arithmetic operators: /, - , (,), id and $. Parse the following string using the same. $ id / ((id - id) / id)$. **07**

(b) Write production and semantic rules for producing following types of statements of programming language 'C': **07**

char * cp , c;

int i , *ip;

float f , *fp, f1;

Specify the type of the attributes used in semantic rules.

OR

Q.3 **(a)** Explain working of simple LR parser with the help of an example. **07**

(b) Consider following production rules: **07**

X = Xab | Y

Y = YZ | d

Z = eZ | epsilon

Apply non-recursive predictive parsing method to parse following string: deabab

Q.4 **(a)** Explain two variants of forms of intermediate code for an assembler. **07**

(b) Construct a syntax tree and a DAG for the following: **07**

a = b + c*d + c *d + e ^ c *d ^ (b + c)

Write three address codes for the both. The operator ^ represents exponent.

OR

Q.4 **(a)** Explain use of OPTAB, SYMTAB and LITTAB for an assembler. Also explain assembler directives – ASSUME and EQU. **07**

Q.4 **(b)** (i) Write unambiguous production rules for producing statements like **07**

a == b

c >= d

e < f

(ii) Explain peep-hole optimization technique.

Q.5 **(a)** Explain in detail and with examples - Advanced macro facilities of an assembler. **07**

(b) Explain program relocation. **07**

OR

Q.5 **(a)** Explain with examples - keyword parameters and default specification of parameters. Give an example of nested macro. Write definition and expansion of it. **07**

(b) Explain in detail how is linking carried out by a linker. **07**

****