Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

M. E. - SEMESTER – II • EXAMINATION – WINTER 2012

Subject code: 1720202 **Date: 31-12-2012**

Subject Name: Design of Language Processors

Time: 10.30 am – 01.00 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)**	List phases of a compiler and give examples of errors detected by each phase. Explain need of intermediate code generator phase.	**07**
	(b)	What is hashing? Explain hashing & rehashing functions and clustering. Also justify suitability of hash table as the appropriate data structure for symbol table.	**07**
Q.2	**(a)**	Explain use and fields of following tables of an assembler OPTABLE, SYMTAB, LITTAB, POOLTAB.	**07**
	(b)	List task performed by each pass of a two pass assembler. Also explain following directives for an assembler: ASSUME, EQU, EXTERN, ORIGIN	**07**
		OR	
	(b)	Explain advanced macro facilities.	**07**
Q.3	**(a)**	Write unambiguous production rules for arithmetic expression consisting of following operators: +, - (binary), - (unary), ( ), *, /, ^ (exponent). Draw parse tree for following : id * id + (id ^ id ^ id) * id * id	**07**
	(b)	Explain working of shift reduce parser. Parse following string using unambiguous production rules: id * id / id ^ id – id + id	**07**
		OR	
Q.3	**(a)**	What is left factoring? Give example. Write unambiguous production rules for if then else construct.	**07**
	(b)	Explain working of an operator precedence parser. Construct precedence graph and precedence table for operators id, +, *, /, $. Parse following string : $ id + id * id / id $	**07**
Q.4	**(a)**	Find first & follow to construct parse table for terminal symbols consisting of id, +, - , /, ( ), $ for unambiguous production rules for non-recursive predictive parser. Parse following string: id + id / id / (id + id).	**07**
	(b)	Construct DFA with out constructing NFA for following regular expression: (a | b) * (b | c) * a*# Write production rules from constructed DFA.	**07**
		OR	
Q.4	**(a)**	Construct NFA and then DFA for following regular expression: (a | b) (b * | c*) a*#	**07**
Q.4	**(b)**	Explain synthesized and inherited attributes by giving two examples.	**07**
Q.5	**(a)**	Explain various parameter passing techniques for functions. Also explain various dynamic memory allocation strategies.	**07**
	(b)	Explain various optimization techniques.	**07**
		OR	
Q.5	**(a)**	Explain activation record and tool yacc.	**07**
	(b)	List and justify your suggested design considerations for a two pass compiler.	**07**

****