Enrollment No: _____ Exam Seat No:_____

C.U.SHAH UNIVERSITY

Summer Examination-2016

Subject Name : System Programming

Subject Code : 4TE06SYP1 **Branch: B.Tech (IT)**

Semester : 6 **Date : 17/05/2016** **Time : 02:30 To 05:30** **Marks : 70**

Instructions:

(1) Use of Programmable calculator & any other electronic instrument is prohibited.
(2) Instructions written on main answer book are strictly to be obeyed.
(3) Draw neat diagrams and figures (if necessary) at right places.
(4) Assume suitable data if needed.

Q-1 **Attempt the following questions** **(14)**

a) The output of lexical analyzer is _____
(a). Set of Regular Expression.
(b). Syntax Tree.
(c). Set of Token.
(d). All above.

b) The gap between Application domain and Execution domain is
(a). Execution.
(b). Semantic.
(c). Specification.
(d). None of above.

c) Software that allows your computer to interact with the user, applications, and hardware is called _____
(a). application software.
(b). word processor.
(c). system software.
(d). database software.

d) In order for a computer to understand a program, it must be converted into machine language by _____
(a). operating system.
(b). utility.
(c). device driver.
(d). language translator.

e) In an absolute loading scheme which loader function is accomplished by assembler?
(a). Re-Allocation.
(b). Allocation.
(c). Linking.
(d). Loading.

f) Which of the following software tool is parser generator
(a). LEX.
(b). YACC.
(c). All Above.
(d). None of above.

g) The gap between PL domain and Execution domain is
(a). Semantic gap.
(b). Specification gap.
(c). Execution gap.
(d). None of above.

h) Which of the following is the most general phase structured grammar
(a). Regular.
(b). Context free.
(c). Context sensitive.
(d). none of these.

i) which of the following system program always reside in main memory
(a). Assembler.
(b). Compiler.
(c). Linker.
(d). Loader.

j) A parser which is a variant of top-down parsing without backtracking is
(a). Recursive Descend.
(b). Operator Precedence.
(c). LL(1) parser.
(d). LALR Parser.

k) Macro definition table is maintained to hold value of sequencing symbols. (State True/False).

l) Define : Operator Grammar.

m) Define : Language Processor.

n) Define : Ambiguous Grammar.

Attempt any four questions from Q-2 to Q-8

Q-2 **Attempt all questions**

(a) Enist and explain various types of grammar. **(04)**

(b) Compare Problem oriented and Procedure oriented languages. **(03)**

(c) Given a grammar, **(07)**
E → TA, A → □ +TA | ε
T → VB, B → □ *VB | ε
V → id | (E)
Develop an LL(1) parser table and parse id * (id + id) string using the parsing table.

Q-3 **Attempt all questions**

(a) What is bottom up parser? Explain operator precedence parser. Let a grammar for a language is E→ E+E | E*E | id. Check validity of following string using stack based operator precedence parser. id * id + id * id **(07)**

	(b)	Convert given regular expression to DFA. The expression is (a | b)*abb#	**(05)**
	(c)	Write regular expressions of a given language. The language consists of all strings of a's and b's which ends with **a** and does not contain **bb**.	**(02)**
Q-4		**Attempt all questions**	
	(a)	Explain advanced macro facilities with suitable example.	**(07)**
	(b)	Describe in detail how program relocation and linking is performed	**(07)**
Q-5		**Attempt all questions**	
	(a)	Explain & compare various intermediate code forms (representations) for an assembler.	**(07)**
	(b)	Explain pass-1 process of Two- pass Assembler with data structure.	**(07)**
Q-6		**Attempt all questions**	
	(a)	Explain pass-1 of Two –pass Macro-preprocessor with data structure.	**(07)**
	(b)	Explain following with Example 1.ORIGIN 2.EQU , DS 3.Expansion time variable 4.AIF && AGO	**(07)**
Q-7		**Attempt all questions**	
	(a)	Explain Subroutine linkage loader.	**(05)**
	(b)	Define Absolute loader.	**(02)**
	(c)	Explain Different phases of compiler with example.	**(07)**
Q-8		**Attempt all questions**	
	(a)	Explain Left factoring, Left recursion and backtracking with example.	**(06)**
	(b)	Describe different types of System software.	**(04)**
	(c)	Explain different elements of language processor.	**(04)**