YMCA UNIVERSITY OF SCIENCE AND RECHNOLOGY, FARIDABAD

BTECH EXAMINATION (Under CBS)

SYSTEM SOFTWARE DESIGN (CE-311)

Time 3hrs. M.Marks:60

Note: There are two parts in paper: PART I and PART II. PART I consist of ten questions (2 marks each) and all are compulsory. Part II contains six questions (10 marks each) from which students have to attempt any four questions.

Q1.

a) What do you mean by bootstrapping?

b) Distinguish between parse tree and syntax tree.

c) Identify the lexemes that make up the tokens for following program fragment. Give reasonable attribute values for the tokens.

```
Void swap(int i, int j)
{       int temp;
        Temp=i;
        I=j;
        J=temp;
}
```

d) How are finite automata useful to lexical analysis?

e) Write the algorithm to remove left recursion from the grammar.

f) Do the left factoring on the following grammar:

$D \rightarrow$ Type List;

Type $\rightarrow$ int/float

List $\rightarrow$ id,List/id

g) How can we remove the ambiguity from a context free grammar?

h) Write the syntax directed definition for while loop.

i) Find the first and follow set for each nonterminal in the following grammar:

$E \rightarrow TE'$

$E' \rightarrow +E/\epsilon$

$T \rightarrow FT'$

$T' \rightarrow T/\epsilon$

$F \rightarrow PF'$

$F' \rightarrow *F'/\epsilon$

$P \rightarrow (E)/a/b/\epsilon$

j) Briefly discuss the functioning of absolute loader.

PART II

Q2. Write regular expression and construct NFA for the following:

a) A real number with optional integer and factional part.

b) A real number with exponent part.

c) A comment string in c language.

Q3. What is the significance of number of pass of compiler? Briefly describe how do various system programs facilitate the execution of program.

Q4. Consider the grammar:

S→*L=R/R

L→**R/id

R→L

Construct CLR parsing table for the above grammar.

Q5. Explain machine dependent and machine independent code optimization. Write the postfix notation of following program fragment:

If x then if a+b then c+d else c-d else c*d

Q6. Explain the need of operator precedence parsing in detail. Also construct the operator precedence parsing table for arithmetic grammar ?

Q7. Write short note on following:

a) Problems in code generation

b) Types of three address code

c) Elimination of useless symbol from the grammar

d) LL(1) parsing (2.5 each)