

Roll Number: _____

Thapar Institute of Engineering & Technology, Patiala

Computer Science & Engineering Department

AUXILIARY EXAMINATION

MCA: Auxiliary Examination	Course Code: PCA511
Aug 14, 2019	Course Name: Compiler Construction
Time: 3 Hours, M. Marks: 100	Wednesday, 17:30 – 20:30 Hrs
	Name Of Faculty: Er.Sanjeev Rao

Note: Attempt all questions

Assume missing data, if any, suitably

Q.1	<p>a) Consider the following grammar :</p> $E \rightarrow E+T/T$ $T \rightarrow T*F/F$ $F \rightarrow (E)/id$ <p>i. Check whether the grammar is ambiguous or not for the sentence id+id*id</p> <p>ii. Remove the left recursion from the above grammar</p> <p>iii. Obtain the first and follow set for the above grammar.</p> <p>b) Explain various issues in design of code generation? Also, explain the role of DAG in optimization with a suitable example.</p>	10*2=20
Q.2	<p>a) Explain the classification of errors encountered during the process of program translation and execution? Discuss various types of Error Recovery strategies adopted at various phases of compiler design.</p> <p>b) Explain directed acyclic graph in detail. Also construct the DAG and sequence of instructions for the following expression :</p> $x + x * (y - z) + (y - z) * w$	10*2=20
Q.3	<p>a) Differentiate between following with suitable examples:</p> <p>i. Inherited and Synthesized attributes</p> <p>ii. Parse tree and Syntax tree</p> <p>iii. Top down and Bottom up Parsers</p> <p>b) What is local and global code optimization? Explain graph coloring algorithm in detail.</p>	10*2=20
Q.4	<p>a) Consider the following code</p> <pre>For i=1 to 10 do For j=1 to 10 do A[i,j]=0</pre>	10*2=20

	<p>For i=1 to 10 do A[i, i]=1.0 Generate intermediate code and identify the basic blocks.</p> <p>b) Consider the following expression : Position=initial +rate *60 Discuss all the phases of compiler w.r.t the above expression with a suitable diagram.</p>	
Q5	<p>a) Write a short note on following with a suitable:</p> <ol style="list-style-type: none"> Cross Compiler and Bootstrapping Regular Expressions and Context Free Grammars Tokens, Patterns and Lexemes <p>b) Consider the following C code:</p> <pre> for(i = 0; i < 10; i++) { for(j = 0; j < 10; j++) { sum = sum + (i * j); } } </pre> <p>Derive the three-address code for the above C code.</p>	10*2=20

***All the best ***