Question Bank

Class: B.Sc [ECS]-III Sem-VI Sub: Compiler Construction (Paper ECS604)

Q.No.1	Choose correct alternatives.	10
1	The compiler process can be considered as a series of sub process is called_____. a) series b) sub process c) phases d) none of these	c
2	The _____ is used to recognition of tokens in lexical anyzer. a) parser b) symbol table c) Finite automata d) none of these	c
3	A compiler for a high level language that runs on one machine and produce code for different machine is called …… . a) a) Optimizing compiler b) One pass compiler c) Multipass compiler d) Cross compiler	d
4	_____ is the process where the stream of characters making up the source program is read from left to right and grouped into tokens. a)Lexical analysis b)Diversion c) Modeling d) None of the above	a
5	In some programming languages, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denotes the sets of letters and digits respectively, which of the following expressions define an identifier ? A. (L∪ D) * B. L (L ∪ D)* C. (L . D)* **D.** L . (L . D)*	b
6	A language L allows declaration of arrays whose sizes are not known during compilation. It is required to make efficient use of memory. Which one of the following is true ? A. a compiler using static memory allocation can be written for L B. a compiler cannot be written for L ; an interpreter must be used C. a compiler using dynamic memory allocation can be written for L D. none of these	c
7	A_____ compiler is also called as residential compiler. a) self b) cross c) both a and b d) none of these	a
8	In some programming languages, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denote the sets of letters and digits respectively, which of the following expressions define an identifier? a) (L∪ D) * b) L(L∪ D) * c) (L. D) * d) L(L∪ D) *	b

9	…….. is a compiler which performs the recompilation of only modified source rather than compiling the whole source program. a)Cross compiler b)Boot strapping compiler c)One pass compiler d)Incremental compiler	d
10	Which of the following is used for grouping of characters into tokens? a) Parser b) code optimizer c) code generator d) scanner	d
11	A compiler that runs on one machine and produces code for a different machine is called_____. a) One pass compilation b) Two pass compilation c) Cross compilation d) None of these	c
12	The _____ should be able to catch syntactic errors. a) Lexical analyzer b) syntax analyzer c) both a and b d) none of these	b
13	The errors comes due to undefined variable, incompatible operands to operator is called _____ errors. a) lexical b) syntactic c) semantic d) logical	c
14	The output of a lexical analyzer is …… a) Machine code b) Intermediate code c) A stream of tokens d)A parse tree.	c
15	Grammar of the programming is checked at _____ phase of compiler. a) Semantic analysis b) code generation c) Syntax analysis d) code optimization.	c
16	…… can perform functions like deleting comments, extra blanks spaces & extra blank lines, keeping track of line numbers. a)Lexical analyzer b) Syntax Analyzer c)Semantic analyzer d)none of these	a
17	Type checking is normally done during_____phase. a)Lexical analysis b)Syntax analysis c)Syntax directed translation d)code optimization	c
18	To recover from an error, the operator precedence parser may A. insert symbols onto the stack and onto the input B. delete symbols from the stack C. delete symbols from the input D. all of these	d

19	Advantage of panic mode o error recovery is that A. it is simple to implement B. it never gets into an infinite loop C. both (a) and (b) D. none of these	c
20	Reduction in strength means A. replacing run-time computation by compile time computation B. replacing a costly operation by a relatively cheaper one C. Both (a) & (b) D. removing loop invariant computation	c
21	The errors comes due to undefined variable, incompatible operands to operator is called _____ errors. b) lexical b) syntactic c) semantic d) logical	c
22	_____ is the activity of filling up unspecified information of labels using appropriate semantic actions in during the code generation process. a) dangling reference b) symbol table c) backtracking d) backpatching	d
23	In _____the parser discards enough number of tokens to reach a descent state on the declaration of errors. a) Panic mode recovery b) parser level recovery c) both a and b d) None of these.	a
24	…… can perform functions like deleting comments, extra blanks spaces & extra blank lines, keeping track of line numbers. a)Lexical analyzer b) Syntax Analyzer c)Semantic analyzer d)none of these	a
25	_____ is a grammar transformation that is useful for producing a grammar suitable for predictive parsing. a) Left recursive b) left factoring c) both a and b d) none of these	b
26	Canonical parser is more powerful in LR parser. a) True b) False	a
27	In _____the parser discards enough number of tokens to reach a descent state on the declaration of errors.	a

	b) Panic mode recovery b) parser level recovery c) both a and b d) None of these.	
28	A right most derivation in reverse is obtained by _____. a) handle b) handle pruning c) grammar d) None of these	b
29	SLR parser is more powerful in LR parser. a) True b) False	b
30	A right most derivation in reverse is obtained by _____. a) handle b) handle pruning c) grammar d) None of these	b
31	Shift reduce parsers are …….. a) Top down parser b) Bottom up parser c) May be top down or bottom up d) None of the above	b
32	Which of the following feature is related with bottom up parsing? a) Shift b) Reduce c) Precedence relation d) All of the above. a) definition b) declaration c) activation d)All of the above	d
33	Shift reduce parsers are ……... a)Top down parser b)Bottom up Parser c)Either a or b d)Both a and b	b
34	Type checking is normally done during_____phase. a)Lexical analysis b)Syntax analysis c)Syntax directed translation d)code optimization	c
35	Shift reduce parsers are ……... a)Top down parser b)Bottom up Parser c)Either a or b d)Both a and b	b
36	SLR parser is more powerful in CLR parser. b) True b) False	b
37	The …… determines the structure of the source string by grouping the tokens together. a)Lexical analyzer b) Syntax Analyzer c)Semantic analyzer d)none of these	b
38	The--------- parser uses reduction process. a)Topdown parser b)Bottom up Parser c)Either a or b d)Both a and b	b
39	……. parsers begin at the start symbol and try to apply productions to arrive at the target string. a)LL parser b)LR parser c)none of these d)all of above	a
40	A bottom up parser generates_____. a) Right most derivations b) Right most derivations in reverse c) Leftmost derivations d) Leftmost derivations in reverse	b

41	An important component of semantic analysis is _____. a) code checking b) type checking c) flush checking d) All of the above	b
42	Which of the following parser is most powerful? a) Operator precedence b) Canonical LR c) LALR d) SLR	c
43	If optimization is over small program segments then it is called as _____ optimization. a) local b) global c) simple d)None of these	a
44	Synthesized attribute can be easily simulated by a…… . a)LL grammar b)Ambiguous grammar c)LR grammar c)None of these	c
45	In a syntax directed translation scheme, if values of an attribute of a node are a function of the attributes of its children, then attribute is called_____. a) canonical attribute b) synthesized attribute c) inherited attribute d) None of these	b
46	A parse tree showing the values of attributes at each node is called an_____. a) derivation tree b) directed acyclic graph c) annotated parse tree d) All of the above	c
47	The activation records are represented by_____. a) three address code b) activation c) activation tree d) none of these	c
48	A computer uses a _____ to keep track of scope and binding information about names. a) phases b) symbol table c) heap allocation d) none of these	b
50	The _____ is technique in compiler used to determine if storage location may be accessed more than one way. a) Symbol table b) alias analysis c) syntax analysis d) Semantic analysis	b
51	In run time environment each node represents _____ of a procedure. a) definition b) declaration c) activation d)All of the above	c
52	In run time environment each node represents _____ of a procedure. a) definition b) declaration c) activation d)All of the above	c
53	The ……. is a machine code that contains reference to actual addresses within programs address. a)absolute code b)relocatable code c)assembler code d)all of the above	a

54	……. is a data structure used by compiler to keep track of semantics of variable. a) Dependency Graph b) Tree c) Symbol table d) None of these.	c
55	……. allocation is for all the data objects at compile time. a)stack b)static c)heap d)all of above	b
56	The Specific task storage manager performs are_____ a) allocation /de-allocation storage to program b) protection of storage area allocated to a program from illegal access by other programs in the system c) the status of each program d) both a and b	d
57	Function of the storage assignment is A. assign storage to all variables referenced in the source program B. assign storage to all temporary locations that are necessary for intermediate results C. assign storage to literals, and to ensure that the storage is allocated and appropriate locations are initialized D. all of these	d
58	A non relocatable program is the one which A. cannot be made to execute in any area of storage other than the one designated for it at the time of its coding or translation B. consists of a program and relevant information for its relocation C. can itself perform the relocation of its address sensitive portions D. all of these	a
59	Running time of a program depends on A. the way the registers and addressing modes are used B. the order in which computations are performed C. the usage of machine idioms D. all of these	d
60	A computer uses a _____ to keep track of scope and binding information about names. a) phases b) symbol table c) heap allocation d) none of these	b

61	The interdependencies among the inherited and synthesized attributes at the nodes in a parse tree can be depicted by a directed graph is called a_____. a) Color graph b) dependency graph c) graph d) acyclic graph	a
62	The attributes that can be computed from the values of the attributes at the siblings and parent of that node is called as…….. . a)synthesized b) inherited c) both a & b d) none of these	b
63	In run time environment each node represents _____ of a procedure. a) definition b) declaration c) activation d)All of the above	c
64	The attributes that can be computed from the values of the attributes at the children's of that node in the parse tree is called as…….. . a)inherited b)synthesized c) both a & b d) none of these	b
65	In case of …….. the size of data objects is known at compile time. a)stack allocation b)heap allocation c)static allocation d)both b and c	c
66	A memory allocates and deallocates storage as needed at runtime from data areas known as_____. a) heap b) stack c) static c) All of these	a
67	_____ values of actual parameters are passed to caller procedure in call by value. a) R b) L c) Both (a) & (b) d) None of these	a
68	The_____ is used to eliminate common sub expression. a) Syntax tree b) annoted parse tree c) DAG d) none of these	c
69	_____ is the sequence of statements in compiler. a) Three address code b) syntax errors c) both a and b d) none of these	a
70	The_____ occurs when there is reference to storage that has been deallocated. a) self reference b) dangling reference c) both a and b d) None of these	b
71	Which of the following is not an intermediate code form? a) Postfix notation b) Syntax trees c) Three address codes d) Quadruples.	d
72	Three address codes are implemented using ……. a) Indirect triples b)Triples c) Quadruples d)All of the above.	d

73	The attributes that can be computed from the values of the attributes at the siblings and parent of that node is called as…….. . c) synthesized b) inherited c) both a & b d) none of these	b
74	…….. is not the form of intermediate representation. a)syntax tree b)postfix c)Three address code d)Quadruple	d
75	The identification of common sub-expression and replacement of run-time computations by compile-time computations is A. local optimization B. loop optimization C. constant folding D. data flow analysis	c
76	An intermediate code form is_____ a) Postfix notation b) Syntax tree c) three address code d) all of above	d
77	Three address codes are implemented using ……. a) Indirect triples b)Triples c) Quadruples d)All of the above.	d
78	The_____ is a flow graph in which there are two types of edges forward edges and backward edges. a) inner loops b) pre-header c) reducible d) all of above	c
79	…… is the activity of filling up unspecified information of labels using appropriate semantic actions in during code generation process. a)backtracking b)triplets c)intermediate code d)backpatching	d
80	…….. is a translation scheme in which the type of each expression is obtained from the types of subexpressions. a)Type checking b)Type analysis c)both a & b d)none of these	a
81	_____ is the activity of filling up unspecified information of labels using appropriate semantic actions in during the code generation process. a) dangling reference b) symbol table c) backtracking d) backpatching	d
82	The_____ is a flow graph in which there are two types of edges forward edges and backward edges. a) inner loops b) pre-header c) reducible d) all of above	c
83	The …… is a sequence of consecutive statements in which flow of control enters at the beginning and leaves at the end without halt or possibility of branching. a)basic blocks b)flow graphs c)directed acyclic graph d)all of the above	a

84	The …… is a new block created such that successor of this block is the header block. a) inner loops b) reducible c) pre-header d) dominators	c
85	The _____ is optional phase of compiler. a) Lexical analyzer b) syntax analyzer c) code optimization d) code generation	c
86	The interdependencies among the inherited and synthesized attributes at the nodes in a parse tree can be depicted by a directed graph is called a_____. b) Color graph b) dependency graph c) graph d) acyclic graph	b
87	The graph that shows basic blocks and their successor relationship is called A. DAG B. Flow graph C. control graph D. Hamiltonion graph	b
88	What are x and y in the following macro definition? macro Add x, y Load y Mul x Store y end macro A. Variables B. Identifiers C. actual parameters D. formal parameters	d
89	What are x and y in the following macro definition? macro Add x, y Load y Mul x Store y end macro a) Variable b) identifier	d

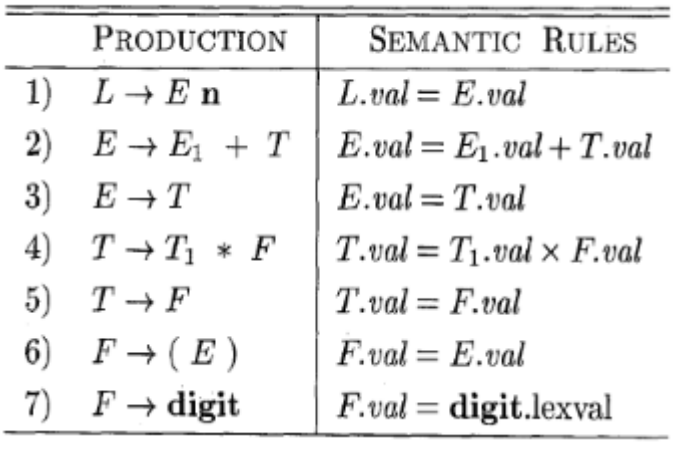
	c) actual parameter d) formal parameter	
90	Concept which can be used to identify loops is _____. a) Dominators b) Reducible graphs c) Depth first ordering d) All of these	d
Q.No.2	Solve any five.	10
1	Give the name of phases of compiler.	
2	Define: 1) Token 2) Pattern	
3	What is the role of lexical analyzer?	
4	What are the different levels of syntax error handlers?	
5	Give the types of errors handled by syntax analyzer.	
6	Write short note on input buffering.	
7	What is the role of syntax analyzer?	
8	What are difficulties with top down parsing?	
9	What are the actions available in shift reduce parser?	
10	Define handle?	
11	Explain left factoring with example.	
12	Define handle with an example.	
13	What is left recursion? How it is eliminate?	
14	Define: 1) Handle 2) Handle Pruning	
15	What is difference between CLR and LALR?	
16	Consider the grammar. E->E+E,E->E*E,E->id. Perform Shift Reduce Parsing of the input string “id-id*id”.	
17	Define an attribute. Give the types of an attributes?	
18	Role of semantic analysis.	
19	Define Short Circuit Code?	
20	What are the three kinds of intermediate representation?	

21	Construct DAG for Expression? i:= i + 10.	
22	Write the Grammar for Flow of Control Statements?	
23	Why there is need of code optimization?	
24	What are the Three Functions of Backpatching?	
25	Construct the DAG for the following blocks. a:=b*c, d:=b, e:=b*c, b:=e, f:=a+c, g:=f+d	
26	Give general form of three address code with an example.	
27	Construct DAG for Expression? x – x * (y + z) – (y + z) * w.	
28	Give the name of back patching functions.	
29	Explain case statements.	
30	What is back patching?	
31	Why there is need of code optimization?	
32	Construct DAG for Expression? a+a*(b-c)+(b-c)*d.	
33	Explain issues in register allocation.	
34	Define the terms inner loops and pre-headers.	

35	Find the basic blocks for the following program.
	prod=0;
	i=1;
	do
	{
	prod=prod+a[i]*b[i];
	i=i+1;
	} while(i<=10);	

36	Define a)dominators b)natural loops	
37	Define Dominators with example.	
Q.No.3 to 5	Solve following.	
1	Explain compiler construction tools in detail.	

2	How to specify and recognize the tokens.	
3	What is compiler? Explain phases of compiler in detail.	
4	What are the types of compiler? Explain in detail.	
5	Explain factor affecting pass structure of compiler.	
6	Write short note on reorganization of tokens.	
7	Explain the different compiler construction tools.	
8	Explain the Pass structure of compiler.Differntiate between Pass and Phase structure of compiler.	
9	Explain compiler construction tools.	
1o	Explain in details factors affecting pass structure of compiler.	
11	Explain the concept of Bootstrapping.	
12	Find first and follow of following grammar: E->E+T|T, T->T*F|F, F->(E)|id.	
13	Construct SLR(1) parsing table for following grammar: S->0S0|1S1|10	
14	Check following grammar is LL(1) grammar or not? A->AcB|cD|D, B->bB|id, D->DaB|BbB|B	
15	Explain operator precedence parser in detail.	
16	Find out triple, quadruple and indirect triple for following: a := b *-c + b *-c * d;	
17	Find out the first and follows of following grammar: S->aABb, A->c| ϵ, B->d| ϵ.	
18	Construct SLR(1) parsing table for following grammar: S->AaAb, S->BbBa, A-> ϵ, B-> ϵ	
19	Check following grammar is LL(1) grammar or not? S->BC|AB, A->aAa| ϵ, B->bAa, c-> ϵ	
20	What is handle? Explain the handle pruning with the help of example.	
21	Find out first and follows of following grammar: S->abSa|aaAb|b, A->baAb|b	
22	Prepare the following grammar for top-down parsing. E->E + E| E - E| E * E| E / E| E ^ E|(E)|id.	

23	Check following grammar is LL(1) grammar or not? S->BC|AB, A->aAa| ε, B->bAa, c-> ε	
24	Check whether following grammar is LL(1) or not. S->AaAb|BbBa, A->ε, B->ε	
25	Construct LALR(1) for following grammar: S->CC, C->aC, C->d	
26	Construct SLR(1) parsing table for following grammar: E->E+T|T, T->TF|F, F->F*|a|b	
27	Check whether following grammar is LR(1) grammar or not. S->AaAb|BbBa, A->ε, B->ε	
28	Construct LL(1) parse table for following grammar: S->aBDh B->cC C->bC|ε D->EF E->g|ε F->f|ε	
29	What is backtracking? Explain backtracking with example.	
30	Construct the predictive parsing table for the following grammar. G= ({ E, E', T, T' , F}, { +, * , (,) , id , num }, P, E) Where P: $E \rightarrow TE'$ $E' \rightarrow +TE' \mid \epsilon$ $T \rightarrow FT'$ $T' \rightarrow *FT' \mid \epsilon$ $F \rightarrow (E) \mid id \mid num$	
31	What is bottom-up parser? Explain in detail shift reduce parsing using stack implementation.	
32	Construct annotated parse tree for 3*5+4n using following grammar rules: 	
33	Explain bottom up evolution of inherited attribute with example.	
34	For the following grammar construct Syntax Directed Definition (SDD). S->EN	

	PRODUCTION	SEMANTIC RULES
1)	$L \rightarrow E$ **n**	$L.val = E.val$
2)	$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3)	$E \rightarrow T$	$E.val = T.val$
4)	$T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5)	$T \rightarrow F$	$T.val = F.val$
6)	$F \rightarrow (E)$	$F.val = E.val$
7)	$F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

	E->E+T |E-T | T T->T*F |T/F | F F->(E) | digit N-> **;**	
35	Explain storage allocation strategies in detail.	
36	Explain the parameter transmission techniques.	
37	What is runtime storage? Explain the runtime storage allocation in detail.	
38	Why symbol table is used? Explain symbol table with its operation.	
39	What does the activation record? Explain field of it.	
40	Find out quadruple, triple and indirect triple for following: `a = b + c * d;`	
41	Contract the dag for the following expression (a+b) * a * a(a+b).	
42	What are the different types of three address statement?	
43	Draw the Syntax tree and DAG for the expression (a*b)+(c-d)*(a*b)+b.	
44	Find out triple, quadruple, indirect triple for following. P=Q+R*S/T+-U*-V	
45	Write the Intermediate Representation like syntax tree and three address code for the following expression (a-b)*(c+d)-(a+b)	
46	What is three address code? Explain implementation of three address code statements.	
47	What is intermediate code generation? Explain types of three address code implementation of statements.	
48	What are issues in code generation?	
49	Explain basic block and flow graph with example.	
50	What is Code generation? What are the issues in Code generation?	
51	Explain peephole optimization in detail.	
52	What is Code Optimization? Explain principle sources of Code Optimization.	
53	Define Code optimization. Explain principle sources of code optimization.	