INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING
ASSIGNMENT

Course Name	**COMPILER DESIGN**
Course Code	A50514
Class	III B. Tech I Semester
Branch	Computer Science and Engineering
Year	2017-2018
Course Coordinator	Ms. B Ramyasree, Assistant Professor.
Course Faculty	Ms. N Mamatha, Assistant Professor, Mr. N PoornaChandra Rao, Assistant Professor.

OBJECTIVES

To meet the challenge of ensuring excellence in engineering education, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education. The major emphasis of accreditation process is to measure the outcomes of the program that is being accredited.

In line with this, Faculty of Institute of Aeronautical Engineering, Hyderabad has taken a lead in incorporating philosophy of outcome based education in the process of problem solving and career development. So, all students of the institute should understand the depth and approach of course to be taught through this question bank, which will enhance learner's learning process.

ASSIGNMENT – I & II

S. No	Question	Blooms Taxonomy Level	Course Outcome
	UNIT – I		
1.	**Convert** 01*+1 regular expression to finite automata?	Remember	2
2.	**Explain** phases of a compiler. Also write down the output for the following expression after each phase x= y*z+10?	Understand	1
3.	**Explain** the general format of a LEX program with example?	Apply	3
4.	**Define** regular expression? State the rules, which define regular expression?	Apply	2
5.	**Explain** the role lexical analyzer and issues of lexical analyzer?	Understand	1
6.	**Explain** the specification of tokens?	Apply	1
7.	**Define** Symbol table?	Apply	1
8.	**Explain** lexeme? Define a regular set?	Remember	2
9.	**Explain** the differences between pass and phase in detail and explain bootstrapping?	Understand	1
10.	**Consider** the grammar $S \rightarrow 0A\|1B\|0\|1$ $A \rightarrow 0S\|1B\|1$ $B \rightarrow 0A\|1S$ **Construct** left most derivations for parse trees for the sentence. i) 1100101 ii) 0101	Apply	4

S. No	Question	Blooms Taxonomy Level	Course Outcome
11.	**Write** FIRST & FOLLOW, construct predictive parsing table for the following grammar E → T E' E' → +TE' / Ɛ T → FT ' T ' → *FT ' /Ɛ F → (E) / id	Analysis	4
12.	**Check** the following grammar is LL(1) or not and construct parsing table. S->AaAb/BbBa A-> Ɛ B-> Ɛ	Analysis	4
13.	**Explain** elimination of left recursion in the grammar E->E+T/T T->T*F/F F->(E) /id	Analysis	4
14.	**Explain** top down parsing methods with example?	Understand	4
15.	**Analyze** whether the following grammar is LL(1) or not. Explain your answer with reasons. S → L, R S → R L → * R L → id R → L	Analysis	4
16.	**For** the operators given below, calculate the operator-precedence relations and operator precedence function. id, +, *, $	Apply	4
17.	**Check** whether the following grammar is a LL(1)grammar S→iEtS|iEtSeS|a E →b Also define the FIRST and FOLLOW procedures.	Apply	4
18.	**Define** the necessary conditions to be carried out before the construction of predictive parser?	Remember	4
19.	**Prepare** the predictive parser for the following grammar: S → a|b|(T) T → T, S|S Write down the necessary algorithms and define FIRST and FOLLOW. Show the behavior of the parser in the sentences, i. (a,(a,a)) ii. (((a,a),a,(a),a)	Apply	4

20.	Consider the following fragment of C code:
	float i, j;
	i = i*70+j+2;
	Write the output at all phases of the compiler for above „C‟ code.
		Apply	1

	UNIT – II		
1.	**Construct** SLR parsing table for S → CC C → aC / b	Apply	5
2.	**Construct** SLR parsing table for S → CC C → aC / b	Apply	5

S. No	Question	Blooms Taxonomy Level	Course Outcome
3.	**Explain** Bottom up parsing method	Understand	5
4.	**Explain** shift reduce parsing method for the following grammar D → type tlist ; Tlist → tlist,id/id type → int/float with input string int id, id;	Apply	5
5.	**Explain** the error recovery in parsing.	Understand	5
6.	**State** shift-reduce parsing? Explain in detail the conflicts that may occur during shift-reduce parsing.	Understand	5
7.	**Prepare** a canonical parsing table for the grammar given below S → CC C → cC|d	Apply	5
8.	For the grammar given below, **calculate** the operator precedence relation and the precedence functions E →E + E|E- E|E * E|E / E|E .E|(E)|-E|id	Understand	5
9.	**Consider** the grammar given below. E→ E + T E →T T→ T * F T →F F→ (E) F→ id Prepare LR parsing table for the above grammar .Give the moves of LR parser on id * id + id.	Apply	5
10.	**Analyze** whether the following grammar is SLR(1) or not. Explain your answer with reasons. S → L,R S → R L → * R L → id R → L	Analysis	5
	UNIT – III		
1.	**State** L – attributed grammars and S- attributed grammars with an example?	Apply	6
2.	**Define** triple, Indirect triple, quadruples with examples?	Remember	8
3.	**Explain** Intermediate code representations?	Understand	8
4.	**Brief** about Syntax Directed Translator?	Apply	7
5.	**Explain** Abstract syntax trees with an example?	Understand	8
6.	**Define** type expression? Explain the equivalence of type expressions with an appropriate example?	Analysis	6

7.	**Generate** the three-address code for the following C program fragment
	while(a > b)
	{
	if (c < d)
	x = y + z;
	else
	x = y - z;
	}
		Understand	8

S. No	Question	Blooms Taxonomy Level	Course Outcome
8.	**Explain** Intermediate code generation for Basic block, Control Flow and Boolean Expressions?	Apply	8
9.	**Explain** how declaration is done in a procedure using syntax directed translation?	Apply	7
10.	**List** the various ways of calling the procedures? Explain in detail?	Analysis	7
11.	**Explain** type expression, type system, simple type checker?	Understand	7
12.	**List** different data structures used for symbol table?	Remember	9
13.	**State** general activation record?	Understand	9
14.	**Explain** type checking for different expressions?	Understand	7
15.	a. **Explain** static and stack storage allocations? b. **Explain** the limitations of static allocation?	Understand	9
16.	**Write** short notes on the specification of a simple type checker?	Understand	7
17.	a. **Compare** three different storage allocation strategies? b. **Explain** symbol table organization using hashing?	Understand	9
18.	a. **List** the various attributes of a symbol table? b. **explain** symbol table organization using trees?	Understand	9
19.	**Describe** various forms of target programs?	Remember	8
20.	**Explain** heap storage allocation and static storage allocation?	Understand	9
	UNIT – IV		
1.	**Describe** 3 areas of code optimization?	Understand	10
2.	**Define** constant folding?	Understand	10
3.	**List** the advantages of the organization of code optimizer?	Understand	10
4.	**Explain** Local optimization and loop optimization in detail.	Understand	10
5.	**Define** Reduction in strength?	Understand	10
6.	**Define** Common Sub expressions?	Understand	11
7.	**Explain** runtime memory divisions?	Understand	11
8.	**Explain** peephole optimization?	Understand	10
9.	**Explain** in the DAG representation of the basic block with example.	Understand	11
10.	a. **Explain** copy propagation and Dead code elimination? **b. What** is live variable?	Remember	12
11.	a. **Explain** local and global common sub expression elimination? b. **Define** a flow graph. Explain how flow graph can be constructed for a given program?	Remember	12
12.	a. **Explain** code hoisting and elimination of loop invariant statements? b. **Explain** how? Redundant sub expression elimination? can be done at global level in a given program?	Understand	10
13.	a. **Describe** local optimization? b. **Explain** any three principal sources of code optimization?	Understand	10
14.	a. **Explain** strength reduction and code movement? b. **Define** basic block? write an algorithm for partitioning into blocks ?	Understand	11
15.	a. **Describe** peephole optimizations? b. **Explain** about loops in flow graphs ?	Understand	12
16.	a. **Explain** loop optimizations? b. **Describe** elimination of common sub expression and elimination of dead Code?	Understand	12
17.	a. **Explain** natural loops and inner loops of a flow graph with an example.	Understand	12

S. No	Question	Blooms Taxonomy Level	Course Outcome
	b. **State** purpose of data flow analysis? Explain available expression and reaching definition?		
18.	**a. Explain** strength reduction and code movement? **b. Define** basic block? write an algorithm for partitioning into blocks ?	Understand	11
19.	**a. Describe** peephole optimizations? **b. Explain** about loops in flow graphs ?	Understand	12
20.	**Explain** in detail the optimization technique “Strength Reduction”?	Understand	12
	UNIT – V		
1.	**Explain** register allocation and assignment?	Understand	13

2.	**Show** the code sequence generated by the simple code generation algorithm
	u := a – c
	v := t + u
	d := v + u//d live at the end
		Apply	13

3.	**Explain** object code forms, generic code algorithm?	Understand	14
4.	**Explain** machine dependent and machine independent optimization?	Understand	14
5.	**List** different data flow properties? Define get reg( ) function?	Apply	14
6.	**Explain** about code generation?	Understand	13
7.	**List** various machine dependent code optimization techniques?	Understand	14
8.	**Explain** the different issues in the design of a code generator?	Understand	13
9.	a. **Describe** various register allocation optimization techniques with an example. b. **generate** code sequence for the following expression using code generation algorithm K :=(a-b) + (a-c) + (a-c)	Apply	13
10.	a. **Explain** about directed acyclic graph (DAG) for register allocation? b. **Discuss** various forms of object code?	Apply	14

Prepared by : Ms. B Ramyasree, Assistant Professor,
Ms.N Mamatha, Assistant Professor,
Mr.N Poornachandra Rao, Assistant Professor.

HOD, COMPUTER SCIENCE AND ENGINEERING