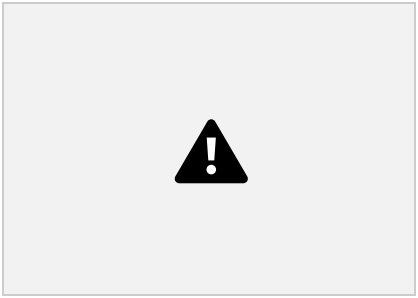
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai - 600089

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



QUESTION BANK

DEGREE / BRANCH: B.TECH-CSE

VI SEMESTER

18CSC304J – COMPILER DESIGN

2018 Regulation

Academic Year 2021-2022 EVEN SEMESTER

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai-600089

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTIONBANK

SUBJECT : COMPILER DESIGN

SEM/YEAR : VI/III

Course Outcomes

CLR-1: Utilize the mathematics and engineering principles for the Design of Compilers
CLR-2: Acquire knowledge of Lexical Analyzer from a specification of a language's lexical rules
CLR-3: Acquire knowledge of Syntax Analyzer for parsing the sentences in a compiler grammar
CLR-4: Gain knowledge to translate a system into various intermediate codes
CLR-5: Analyze the methods of implementing a Code Generator for compilers
CLR-6: Analyze and Design the methods of developing a Code Optimizer

UNIT- IV			
Intermediate Code Generation, Intermediate Languages - prefix – postfix, Quadruple - triple - indirect triples, Representation- Syntax tree- Evaluation of expression –three address code, Synthesized attributes – Inherited attributes- Intermediate languages – Declarations, Assignment Statements- Boolean Expressions, Case Statements, Back patching – Procedure calls- Code Generation, Issues in the design of code generator- The target machine – Runtime Storage management, A simple Code generator- Code Generation Algorithm, Register and Address Descriptors- Generating Code of Assignment Statements, Cross Compiler – T diagrams- Issues in Cross compilers.			
PART-A (Multiple Choice Questions)			
Q. No	**Questions**	**Course Outcome**	**Competence BT Level**
1	**Consider the intermediate code given below:** **1. i = 1** **2. j = 1** **3. t1 = 5 * i** **4. t2 = t1 + j** **5. t3 = 4 * t2** **6. t4 = t3**	CO3	BT1

	7. a[t4] = –1 **8. j = j + 1** **9. if j <= 5 goto(3)** **10. i = i + 1** **11. ifi< 5 goto(2)** **The number of nodes and edges in the control-flow-graph constructed for the above code, respectively, are** a. 5 and 7 b. 6 and 7 c. 5 and 5 d. 7 and 8 **Answer: b**		
2	**Generation of intermediate code based on a abstract machine model is useful in compilers because** a. Implementation of lexical analysis and syntax analysis is made easier b. Writing for intermediate code generation c. Portability of the front end of the compiler d. None of the mentioned **Answer: a**	CO3	BT3
3	**Some code optimizations are carried out on the intermediate code because** a. they enhance the portability of the compiler to other target processors b. program analysis is more accurate on intermediate code than on machine code c. the information from dataflow analysis cannot otherwise be used for optimization d. the information from the front end cannot otherwise be used for optimization **Answer: a**	CO3	BT1
4	**Input to code generator** a. Source code b. Intermediate code c. Target code d. All of the above **Answer: b**	CO3	BT1
5	**The code optimization is carried out on the intermediate code**	CO3	BT1

	a. Because for optimization information from the front end cannot be used. b. Because program is more accurately analysed on intermediate code than on machine code. c. Because for optimization information from data flow analysis cannot be used. d. Because they enhance the portability of the compiler to the other target processor. **Answer: b**		
6	**Which of the following is an infix expression?** a. (a+b)*(c+d) b. ab+c* c. +ab d. abc+* **Answer: a**	CO3	BT4
7	**What is the postfix expression for the corresponding infix expression?** a. abc*+de*+ b. abc+*de*+ c. a+bc*de+* d. abc*+(de)*+ **Answer: a**	CO3	BT4
8	**In infix to postfix conversion algorithm, the operators are associated from?** a. right to left b. left to right c. centre to left d. centre to right **Answer: b**	CO3	BT4
9	**From the given Expression tree, identify the correct postfix expression from the list of options.**  a. ab*cd*+ b. ab*cd-+ c. abcd-*+ d. ab*+cd- **Answer: b**	CO3	BT4

10	**Which statement is an abstract form of intermediate code?** a. 3- address b. 2-address c. Address d. Intermediate code **Answer: a**	CO3	BT2
11	**In the compiler, the function of using intermediate code is:** a. to improve the register allocation b. to increase the error reporting & recovery. c. to make semantic analysis easier. d. to increase the chances of re-using the machine-independent code optimizer in other compilers. **Answer: d**	CO3	BT2
12	**Synthesized attribute can be easily simulated by a** a. LR grammar b. LL grammar c. Ambiguous grammar d. None of these **Answer: a**	CO3	BT2
13	**Consider the following C code segment.** **for (i = 0, i<n; i++)** **{** **for (j=0; j<n; j++)** **{** **if (i%2)** **{** **x += (4*j + 5*i);** **y += (7 + 4*j);** **}** **}** **}** Which one of the following is false? a. The code contains loop invariant computation b. There is scope of common sub-expression elimination in this code c. There is scope of strength reduction in this code d. There is scope of dead code elimination in this code **Answer: d**	CO4	BT4
14	The language that needs heap allocation in the runtime environment are a. Those that use global variables b. Those that use dynamic scoping c. Those that support recursion d. Those that allow dynamic data structure	CO4	BT1

	Answer: d		
15	Some code optimizations are carried out on the intermediate code because a. They enhance the portability of the compiler to other target processors b. Program analysis is more accurate on intermediate code than on machine code c. The information from dataflow analysis cannot otherwise be used for optimization d. The information from the front end cannot otherwise be used for optimization **Answer: a,b**	CO4	BT4
16	One of the purposes of using intermediate code in compilers is to a. make parsing and semantic analysis simpler. b. improve error recovery and error reporting. c. increase the chances of reusing the machine-independent code optimizer in other compilers. d. improve the register allocation. **Answer: c**	CO4	BT1
17	Which of the following is not performed during compilation a. Dynamic memory allocation b. Type checking c. Symbol table Management d. Inline Expansion **Answer: a**	CO4	BT1
18	For a C program accessing X[i] [j] [k], the following intermediate code is generated by a compiler. Assume that the size of an integer is 32 bits and the size of a character is 8 bits. t0 = i * 1024 t1 = j * 32 t2 = k * 4 t3 = t1 + t0 t4 = t3 + t2 t5 = X[t4] Which one of the following statements about the source code for the C program is **CORRECT**? a. X is declared as "int X[32] [32] [8]" b. X is declared as "int X[4] [1024] [32]" c. X is declared as "char X[4] [32] [8]" d. X is declared as "char X[32] [16] [2]"	CO4	BT4
19	In activation record, Which of the following Stores the address of activation record of the caller procedure? a. Access Link b. Actual Parameters	CO4	BT1

	c. Control Link d. Temporaries **Answer: c**		
20	Which of the following known as the text part of a program that does not change at run time. Its memory requirements are known at the compile time? a. Code b. Procedures c. Variables d. All of the above **Answer: a**	CO4	BT2
	PART - B (4 Marks)		
1	**Construct triples of an expression a * - (b + c)**	CO4	BT 5
2	**What is back patching?**	CO4	BT 1
3	**Write three address code for x = *y; a = &x;**	CO3	BT 5
4	**What is the idea behind generating three address code during compilation?**	CO3	BT 1
5	**Why are quadruples preferred over triples in an optimizing compiler?**	CO3	BT 2
6	**What are the notations used to represent an intermediate language?**	CO3	BT 1
7	**How would you solve the issues in the design of code generators?**	CO3	BT 2
8	**Discuss in detail about the four principle uses of registers in code generation.**	CO5	BT2
9	**Translate the given assignment statement d:= (a-b) + (a-c) + (a-c) into the intermediate code.**	CO3	BT 5
	PART – C (12 Marks)		
1	**What is three address code? What is its type? How is it implemented?**	CO4	BT 1
2	**Discuss the various methods for translating the Boolean expression.**	CO4	BT 2
3	**Explain the following grammar for a simple procedure call statement. S -> call id (Elist).**	CO4	BT 5
4	**Using Backpatching, generate an intermediate code for the following expression A < B OR C < D AND P < Q**	CO4	BT 5
5	**Illustrate in detail about the code generation algorithm with an example.**	CO5	BT3
6	**Create following assignment statement into three address code D:=(a-b)*(a-c)+(a-c). Apply code generation algorithm to generate a code sequence for the three address statement.**	CO5	BT6
7	**Generate code for the following sequence assuming that n is in a memory location** **s=0** **i=0** **L1 : if I > n goto L2**	CO5	BT6

	s=s+i **i=i+1** **goto L1** **L2 :**		
8	**Explain in detail about the issues in code generation with examples**	CO5	BT1
9	**Generate the code sequence for the assignment statement d:= (a-b) + (a-c) + (a-c).**	CO5	BT6

Note:

1. **BT Level –** Blooms Taxonomy Level
2. **CO –** Course Outcomes

BT1 – Remember BT2 – Understand BT3 – Apply BT4 – Analyze
BT5 – Evaluate BT6 – Create