INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad - 500 043

INFORMATION TECHNOLOGY

DEFINITIONS AND TERMINOLOGY QUESTION BANK

Course Name	:	**COMPILER DESIGN**
Course Code	:	**AIT004**
Program	:	**B.Tech**
Semester	:	**V**
Branch	:	**Computer Science and Engineering**
Section	:	**A,B,C,D**
Academic Year	:	**2019 - 2020**
Course Faculty	:	**Dr. K Srinivasa Reddy, Professor** **Y Harika Devi, Assistant Professor**

COURSE OBJECTIVES:

The course should enable the students to:	
I	Apply the principles in the theory of computation to the various stages in the design of compilers.
II	Demonstrate the phases of the compilation process and able to describe the purpose and operation of each phase.
III	Analyze problems related to the stages in the translation process.
IV	Exercise and reinforce prior programming knowledge with a non-trivial programming project to construct a compiler.

DEFINITIONS AND TERMINOLOGY QUESTION BANK

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
UNIT-I						
1	Define Compiler	Compiler is a computer program that translates computer code written in one programming language (the source language) into another programming language (the target language).	Remember	CO 1	CLO 1	AIT004.01
2	Define interpreter	An interpreter is a computer program that is used to directly execute program instructions written using one of the many high-level programming languages.	Remember	CO 1	CLO 1	AIT004.01
3	Define Translator	A translator is a program that takes as input a program written in one language and produces as output a program in another language.	Remember	CO 1	CLO 1	AIT004.01
4	What are the types of translators	There are three types a) Interpreter b)Compiler	Remember	CO 1	CLO 1	AIT004.01

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
		c)preprocessor				
5	What is a Token	Token is a sequence of characters that can be treated as a single logical entity.	Remember	CO 1	CLO 1	AIT004.01
6	Define Pattern	A set of strings in the input for which the same token is produced as output. This set of strings is described by a rule called a pattern associated with the token.	Remember	CO 1	CLO 3	AIT004.03
7	List Some Compilers	a)Ada compilers b)ALGOL compilers c) BASIC compilers d)C# compilers e) C compilers f) C++ compilers g) COBOL compilers h) Java compilers	Remember	CO 1	CLO 1	AIT004.01
8	Define Lexeme	A lexeme is a sequence of characters in the source program that is matched by the pattern for a token.	Remember	CO 1	CLO 1	AIT004.01
9	Define Left Recursion	Left recursion is a special case of recursion where a string is recognized as part of a language by the fact that it decomposes into a string from that same language (on the left) and a suffix (on the right).	Understand	CO 1	CLO 4	AIT004.04
10	What are specifications of tokens	There are 3 specifications of tokens: a) Strings b)Language c) Regular expression	Understand	CO 1	CLO 1	AIT004.01
11	Define Language	A language is a set of strings, chosen form some Σ* or A language is a subset of Σ*. A language which can be formed over 'Σ'can be Finite or Infinite. Language that contains strings over ∑ = {a, b} are {ε, a, b, aa, ab}	Remember	CO 1	CLO 3	AIT004.03
12	What is a regular expression	Regular expression is an Algebraic way to represent a language.	Understand	CO 1	CLO 3	AIT004.03
13	Define Automata	Automation is defined as a system where information is transmitted and used for performing some functions without direct participation of man.	Understand	CO 1	CLO 2	AIT004.02
15	What is the difference between compiler and interpreter	A compiler converts the high level instruction into machine language while an interpreter converts the high level instruction into an intermediate form.	Remember	CO 1	CLO 1	AIT004.01
16	What are the functions of parser	a)It checks if the tokens from lexical analyzer, occur in pattern that are permitted by the	Understand	CO 1	CLO 4	AIT004.04

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
		specification for the source language. b)It also imposes on tokens a tree-like structure that is used by the sub-sequent phases of the compiler.				
17	Define Lexical Analyzer	Lexical analyzer (the "lexer") parses individual symbols from the source code file into tokens.	Remember	CO 1	CLO 3	AIT004.03
18	Define Phase	A phase is a logically interrelated operation that takes source program in one representation and produces output in another representation.	Remember	CO 1	CLO 4	AIT004.04
19	What is a Loader	A loader is a program that places programs into memory and prepares them for execution.	Understand	CO 1	CLO 1	AIT004.01
20	Identify the properties under which regular languages are not closed	Subset, superset, infinite union and infinite intersection.	Remember	CO 1	CLO 3	AIT004.03
21	Identify Regular expression for strings without two consecutive one's 11?	Regular expression for all strings which may begin with either 0 or 1 and without consecutive one's (1+ €)(0+01)*	Understand	CO 1	CLO 3	AIT004.03
22	What is an Identifier	Identifiers are the set or string of letters and digits beginning with a letter.	Remember	CO 1	CLO 1	AIT004.01
23	Explain Recursive Decent Parsing	Recursive descent is a top-down parsing technique that constructs the parse tree from the top and the input is read from left to right.	Understand	CO 1	CLO 5	AIT004.05
24	Define ambiguous grammar	A grammar is said to be ambiguous if it has more than one derivation trees for a sentence or in other words if it has more than one leftmost derivation or more than one rightmost derivation	Remember	CO 1	CLO 2	AIT004.02
25	Define pass	The transversal of a compiler through the entire program is known as a Pass.	Remember	CO 1	CLO 1	AIT004.01
26	What is Left factoring	Left factoring is a grammar transformation technique used for removing the common left factor that appears in two productions of the same non-terminal.	Remember	CO 1	CLO 4	AIT004.04
27	What is LL parser	LL parser is a top down parser which parses the input from Left to right, performing Leftmost derivation of the sentence	Understand	CO 1	CLO 5	AIT004.05
28	What is the role of Top-down Parser	In Top down parser parsing starts with the starting symbol S. and moves down from root node to leaf nodes using productions	Remember	CO 1	CLO 5	AIT004.05

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
29	List the properties of Derivation tree	a)The root node is always a node indicating start symbol b)The derivation is read from left to right c)The leaf nodes always terminals nodes d)The interior nodes are always non terminal nodes	Remember	CO 1	CLO 4	AIT004.04
30	Define sub tree?	A subtree of a derivation tree is a particular vertex of the tree together with all its descendants ,the edges connecting them and their labels. The label of the root may not be the start symbol of the grammar.	Remember	CO 1	CLO 4	AIT004.04
		UNIT-II				
1	Define Parser.	A parser takes input in the form of sequence of tokens and produces output in the form of parse tree.	Remember	CO 2	CLO 4	AIT004.05
2	List the types of parsers.	There are two types of parsers a) Topdown parsing b) Bottom parsing	Remember	CO 2	CLO 5	AIT004.05
3	What is bottom up parsing?	In the bottom up parsing, the parsing starts with the input symbol and construct the parse tree up to the start symbol by tracing out the rightmost derivations of string in reverse.	Understand	CO 2	CLO 5	AIT004.05
4	What is top-down parsing?	Top-down parsing constructs parse tree for the input string, starting from root node and creating the nodes of parse tree in pre-order. It is done by leftmost derivation for an input string.	Understand	CO 2	CLO 5	AIT004.05
5	What are the tasks performed by parser.	a)Helps you to detect all types of Syntax errors b)Find the position at which error has occurred c)Clear & accurate description of the error. d)Recovery from an error to continue and find further errors in the code. e)The parse must reject invalid texts by reporting syntax errors	Understand	CO 2	CLO 4	AIT004.04
6	What are the common errors occur in parser.	Common Errors that occur in Parsing a)Lexical: Name of an incorrectly typed identifier b)Syntactical: unbalanced parenthesis or a missing semicolon c)Semantical: incompatible	Remember	CO 2	CLO 4	AIT004.04

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
		value assignment d)Logical: Infinite loop and not reachable code				
7	Define handle.	A handle of a string is a substring that matches the right side of a production and whose reduction to the non-terminal on the left side of the production represents one step along the reverse of a rightmost derivation.	Remember	CO 2	CLO 5	AIT004.05
8	What is Handle pruning?	If A -->β is a production then reducing β to A by the given production is called handle pruning i.e., removing the children of A from the parse tree. A rightmost derivation in reverse can be obtained by handle pruning.	Remember	CO 2	CLO 5	AIT004.05
9	Define shift reduce parser.	Shift Reduce parsing is a bottom-up parsing that reduces a string w to the start symbol of grammar.It scans and parses the input text in one forward pass without backtracking.	Remember	CO 2	CLO 4	AIT004.04
10	What are the actions of shift reduce parser.	A shift-reduce parser can make four possible actions a) shift b) reduce c) accept d) error.	Remember	CO 2	CLO 4	AIT004.04
11	What are the conflicts occurs in shift reduce parser	There are two conflicts occur in shift-Reduce parser. a)Shift-Reduce conflict b)Reduce-Reduce conflict	Understand	CO 2	CLO 4	AIT004.04
12	What is shift step.	The input pointer advances to the next input symbol by the shift step and the next input symbol is known as shifted symbol and the symbol is pushed upon stack. The shifted symbol is considered as a single node of the parse tree.	Understand	CO 2	CLO 5	AIT004.05
13	What is reduce step.	When a complete grammar rule RHS is replaced by LHS it is termed as reduce-step. The stack performs a pop function which facilitates in popping off the handle and replacing with the LHS non-terminal symbol.	Understand	CO 2	CLO 5	AIT004.05
14	What is LR Parser	The LR parser is a non-recursive, shift-reduce, bottom-up parser. It uses a wide class of context-free grammar which makes it the most efficient syntax analysis technique.	Remember	CO 2	CLO 5	AIT004.05
15	What is LR(K) parser.	LR parsers are also known as LR(k) parsers, where L stands for left-to-right scanning of the	Remember	CO 2	CLO 5	AIT004.05

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
		input stream; R stands for the construction of right-most derivation in reverse, and k denotes the number of lookahead symbols to make decisions.				
16	List the different types of LR Parsers.	LR parsing is divided into four parts: a)LR (0) parsing, b) SLR parsing, c)CLR parsing d)LALR parsing.	Remember	CO 2	CLO 4	AIT004.04
17	What is LR algorithm.	The LR algorithm requires stack, input, output and parsing table. In all type of LR parsing, input, output and stack are same but parsing table is different.	Remember	CO 2	CLO 4	AIT004.04
18	What is Input buffer?	Input buffer is used to indicate end of input and it contains the string to be parsed followed by a $ Symbol.	Remember	CO 2	CLO 4	AIT004.04
19	What is parsing table.	Parsing table is a two dimensional array. It contains two parts: a)Action part b)GoTo part.	Understand	CO 2	CLO 4	AIT004.04
20	What is Augmented Grammar?	Augmented grammar G` will be generated if we add one more production in the given grammar G. It helps the parser to identify when to stop the parsing and announce the acceptance of the input. for example S`→ S S → AA A → aA | b	Understand	CO 2	CLO 5	AIT004.04
21	What is SLR (1) Parser.	a) It is Simple LR Parser b)It works on smallest class of grammar c)Few number of states, hence very small table c)Simple and fast construction	Remember	CO 2	CLO 5	AIT004.05
22	What is LR (1) Parser.	a)It is LR Parser b)It works on complete set of LR(1) Grammar c)Generates large table and large number of states d)Slow construction	Remember	CO 2	CLO 5	AIT004.05
23	What is LALR (1) Parser.	a) It is Look-Ahead LR Parser b) Works on intermediate size of grammar c)Number of states are same as in SLR(1)	Remember	CO 2	CLO 5	AIT004.05

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
24	What are the differences between LL and LR	LL: a)leftmost derivation is done b) Parse tree is built top-down. c) Starts with the root non-terminal on the stack. LR: a) Rightmost derivation is done. b) Parse tree is built bottom-up. c) Ends with the root non-terminal on the stack.	Understand	CO 2	CLO 5	AIT004.05
25	What is CLR(1) parsing.	CLR refers to canonical lookahead. CLR parsing use the canonical collection of LR (1) items to build the CLR (1) parsing table. CLR (1) parsing table produces the more number of states as compare to the SLR (1) parsing.	Remember	CO 2	CLO 5	AIT004.05
26	What is LR(1) item.	LR (1) item is a collection of LR (0) items and a look ahead symbol. LR (1) item = LR (0) item + look ahead The look ahead is used to determine that where we place the final item.The look ahead always add $ symbol for the argument production.	Understand	CO 2	CLO 5	AIT004.05
27	What is YACC tool?	a)YACC stands for Yet Another Compiler Compiler. b)YACC provides a tool to produce a parser for a given grammar. c)YACC is a program designed to compile a LALR (1) grammar.	Understand	CO 2	CLO 6	AIT004.06
28	What is panic-mode error recovery?	In the case when the parser encounters an error, this mode ignores the rest of the statement and not process input from erroneous input to delimiter, like a semi-colon. This is a simple error recovery method.	Remember	CO 2	CLO 6	AIT004.06
29	What is phrase-level error recovery?	Compiler corrects the program by inserting or deleting tokens. This allows it to proceed to parse from where it was. It performs correction on the remaining input. It can replace a prefix of the remaining input with some string this helps the parser to continue the process.	Remember	CO 2	CLO 6	AIT004.06
30	What is canonical collection of LR(0) items.	An LR (0) item is a production G with dot at some position on the right side of the production. LR(0) items is useful to indicate that how much of the input has been scanned up to a given point in the process of parsing.	Understand	CO 2	CLO 5	AIT004.05

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
UNIT-III						
1	Define Syntax Directed Translation	Syntax Directed Translations are augmented rules to the grammar that facilitate semantic analysis.	Remember	CO 3	CLO 7	AIT004.07
2	Define Syntax tree	A Syntax tree is a graphical depiction of a string derivation.	Remember	CO 3	CLO 7	AIT004.07
3	Define Attribute grammar	Attribute grammar is a special form of context-free grammar where some additional information (attributes) is appended to one or more of its non-terminals in order to provide context-sensitive information.	Remember	CO 3	CLO 7	AIT004.07
4	Define Synthesized attributes.	These attributes get values from the attribute values of their child nodes.	Remember	CO 3	CLO 7	AIT004.07
5	Define Inherited attributes.	Inherited attributes can take values from parent and/or siblings.	Remember	CO 3	CLO 7	AIT004.07
6	List the types of Attribute Grammar.	Synthesized attributes and inherited attributes.	Remember	CO 3	CLO 7	AIT004.07
7	What is S-attributed SDT ?	If an SDT uses only synthesized attributes, it is called as S-attributed SDT.	Remember	CO 3	CLO 8	AIT004.08
8	What is L-attributed SDT?	This form of SDT uses both synthesized and inherited attributes with restriction of not taking values from right siblings.	Remember	CO 3	CLO 8	AIT004.08
9	What is abstract syntax tree?	An abstract syntax tree, or just syntax tree, is a tree representation of the abstract syntactic structure of source code written in a programming language.	Remember	CO 3	CLO 7	AIT004.07
10	What is polish notation?	Prefix notation is also known as Polish Notation. In this operator is prefixed to operands.	Remember	CO 3	CLO 11	AIT004.11
11	What is three address code?	Three address code is a type of intermediate code which is easy to generate and can be easily converted to machine code.It makes use of at most three addresses and one operator to represent an expression and the value computed at each instruction is stored in temporary variable generated by compiler.	Remember	CO 3	CLO 11	AIT004.11
12	List the Intermediate forms of source programs	Abstract syntax tree, polish notation and three address code	Remember	CO 3	CLO 11	AIT004.11
13	List the types of three address codes.	a)Quadruple b)Triples c) Indirect Triples	Remember	CO 3	CLO 11	AIT004.11

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
14	What is Quadruple notation of three address code?	Each instruction in quadruples presentation is divided into four fields: operator, arg1, arg2, and result.	Remember	CO 3	CLO 11	AIT004.11
15	What is Triples notation of three address code?	Each instruction in triples presentation has three fields : op, arg1, and arg2.The results of respective sub-expressions are denoted by the position of expression.	Remember	CO 3	CLO 11	AIT004.11
16	What is Indirect Triples notation of three address code?	This representation is an enhancement over triples representation. It uses pointers instead of position to store results.	Remember	CO 3	CLO 11	AIT004.11
17	What is Reversed polish notation?	Postfix notation is also known as Reversed Polish Notation. In this the operator is postfixed to the operands.	Remember	CO 3	CLO 11	AIT004.11
18	Write postfix notation of (a+b)*c	ab+c*	Understand	CO 3	CLO 11	AIT004.11
19	Write postfix notation of a*(b+c)	abc+*	Understand	CO 3	CLO 11	AIT004.11
20	Write postfix notation of (a+b)*(c+d)	ab+cd+*	Understand	CO 3	CLO 11	AIT004.11
21	What is Annotated Parse Tree?	A parse tree showing the values of attributes at each node is called an Annotated parse tree.	Remember	CO 3	CLO 11	AIT004.11
22	What is High Level IR?	High-level intermediate code representation is very close to the source language itself. They can be easily generated from the source code and we can easily apply code modifications to enhance performance.	Remember	CO 3	CLO 9	AIT004.09
23	What is Low Level IR?	Low-level intermediate code representation is close to the target machine, which makes it suitable for register and memory allocation, instruction set selection, etc.	Remember	CO 3	CLO 9	AIT004.09
24	Write three address code for the expression a*b+c	t1= a*b t2=t1+c	Understand	CO 3	CLO 9	AIT004.09
25	Write three address code for the expression (a+b)*c	t1= a+b t2=t1*c	Understand	CO 3	CLO 9	AIT004.09
26	Write the SDT for following statement E→E+T	{E.value:=E.value+T.value}	Understand	CO 3	CLO 8	AIT004.08
27	Write the SDT for following statement T→T*F	{T.value:=T.value*F.value}	Understand	CO 3	CLO 8	AIT004.08

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
28	What are the functions used for constructing syntax trees.	a)mknode(op,left,right) b)mkleaf(id,entry) c)mkleaf(num,val)	Remember	CO 3	CLO 8	AIT004.08
29	Write the SDT for following statement F→digit	{F.value:=digit.lexicalvalue}	Understand	CO 3	CLO 8	AIT004.08
30	Write the SDT for following statement T→T/F	{T.value:=T.value/F.value}	Understand	CO 3	CLO 8	AIT004.08
UNIT-IV						
1	Define Type Checking?	Type checking is simply testing for type errors in given program, either by the compiler or during program execution.	Remember	CO4	CLO 13	AIT004.13
2	What is Static checking?	Static checking includes the syntax checks performed by the parser and semantic checks such as type checks, flow-of- control checks, uniqueness checks, and name-related checks.	Remember	CO4	CLO 14	AIT004.14
3	What is Dynamic checking?	Dynamic type checking is the process of verifying the type safety of a program at runtime. Common dynamically-typed languages include Groovy, JavaScript, Lisp, Objective-C, PHP, Prolog, Python, Ruby, Small talk .	Remember	CO 4	CLO 14	AIT004.14
4	Explain Type Expression?	Type expressions are built from basic types and constructors, a natural concept of equivalence between two type expressions is structural equivalence. i.e., two expressions are either the same basic type or formed by applying the same constructor to structurally equivalent types.	Understand	CO 4	CLO 13	AIT004.13
5	Define Function Overloading?	Function overloading or method overloading is the ability to create multiple functions of the same name with different implementations.	Remember	CO 4	CLO 13	AIT004.13
6	Explain structural equivalence of type expression?	Type expression are built from basic types and constructors ,a natural concept of equivalence between two type expressions is structural equivalence i.e two expressions are either the same basic type or formed by applying the same constructor to same equivalent type.	Understand	CO 4	CLO 13	AIT004.13
7	Explain any two uses of type checking?	Depending on Language Type checker can prevent 1)Application of a function to wrong number of arguments	Understand	CO 4	CLO 15	AIT004.15

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
		2)use of undeclared variables in expressions				
8	What is flow of control checks?	Statements that cause flow of control to leave a construct must have some place to which to transfer the flow of control.	Remember	CO 4	CLO 15	AIT004.15
9	What are Type systems?	A type system is a set of rules for assigning type expressions to the syntactic constructs of a program.	Remember	CO 4	CLO 14	AIT004.14
10	Define strongly typed language?	A strongly typed language is one in which the compiler can guarantee that the programs it accepts will run without type errors	Remember	CO 4	CLO 14	AIT004.14
11	Define Type Inference?	Type Inference are rules that determine the type of a language construct based on how it is used.	Remember	CO 4	CLO 14	AIT004.14
12	What is Dynamic Typed language?	A dynamically typed language is one in which some of the constructs of a language can only be typed at run time. Perl, Python, and Lisp are dynamically typed.	Remember	CO 4	CLO 15	AIT004.15
13	Define Operator overloading?	Operator overloading is a technique by which operators used in a programming language are implemented in user-defined types with customized logic that is based on the types of arguments passed	Understand	CO 4	CLO 13	AIT004.13
14	Define Activation trees.	A program consist of procedures, a procedure definition is a declaration that, in its simplest form, associates an identifier (procedure name) with a statement (body of the procedure). Each execution of procedure is referred to as an activation of the procedure.	Remember	CO 4	CLO 14	AIT004.14
15	What are the properties of Activation trees?	a)Each node represents an activation of a procedure. b)The root shows the activation of the main function. c)The node for procedure ‘x’ is the parent of node for procedure ‘y’ if and only if the control flows from procedure x to procedure y.	Understand	CO 4	CLO 14	AIT004.14
16	What is Control stack?	Control stack or runtime stack is used to keep track of the live procedure activations i.e the procedures whose execution have not been completed. A procedure name is pushed on to the stack when it is called (activation begins) and it is popped when it returns (activation ends).	Understand	CO 4	CLO 14	AIT004.14

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
17	What is Activation record?	Activation records consist of a)local variables b)Temporary values c)Machine status d)Access link e)Control link f)Return value	Remember	CO 4	CLO 14	AIT004.14
18	List the various storage allocation strategies.	There are three allocation strategies. a) Static Storage Allocation b) Stack Storage Allocation c) Heap Storage Allocation	Remember	CO 4	CLO 13	AIT004.13
19	What is Static storage allocation?	Any program if we create memory at compile time, memory will be created in the static area and memory is created only once.	Remember	CO 4	CLO 13	AIT004.13
20	What is Stack storage allocation	Storage is organised as a stack and activation records are pushed and popped as activation begin and end respectively. Locals are contained in activation records so they are bound to fresh storage in each activation.	Remember	CO 4	CLO 13	AIT004.13
21	What is Heap storage allocation	Heap allocation is used to dynamically allocate memory to the variables and claim it back when the variables are no more required. Memory allocation and deallocation can be done at any time and at any place depending on the requirement of the user.	Remember	CO 4	CLO 13	AIT004.13
22	What is Symbol table?	Symbol table is an important data structure created and maintained by compilers in order to store information about various entities such as variable names, function names, objects, classes, interfaces, etc.	Remember	CO 4	CLO 15	AIT004.15
23	List the various implementation of symbol table.	A symbol table can be implemented in one of the following ways: a)Linear list b)Binary search tree c)Hash table	Remember	CO 4	CLO 15	AIT004.15
24	What are the operations in symbol table?	There are two operations. a)insert() b)lookup()	Remember	CO 4	CLO 15	AIT004.15
25	What is lookup() operation.	Lookup() operation is used to search a name in the symbol table to determine: a)if the symbol exists in the table. b) if the name is used in the scope. c) if the symbol is initialized. d)if the symbol declared multiple times.	Understand	CO 4	CLO 15	AIT004.15

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
26	What is scope management?	A compiler maintains two types of symbol tables: a global symbol table which can be accessed by all the procedures and scope symbol tables that are created for each scope in the program.	Remember	CO 4	CLO 15	AIT004.15
27	Difference between static and Dynamic allocation.	Static: a)Memory is allocated before the execution of the program begins. b)Implemented using stacks. c) In this type of allocation Memory cannot be resized after the initial allocation. Dynamic: a)Memory is allocated during the execution of the program. b)Implemented using heap. c)In this type of allocation Memory can be dynamically expanded and shrunk as necessary.	Remember	CO 4	CLO 14	AIT004.14
28	What is control link and access link?	Control link points to activation record of the caller. Access Link is used to refer to non-local data held in other activation records.	Understand	CO 4	CLO 17	AIT004.14
		UNIT-V				
1	Define Code Optimization.	The code optimization in the synthesis phase is a program transformation technique, which tries to improve the intermediate code by making it consume fewer resources (i.e. CPU, Memory) so that faster-running machine code will result.	Remember	CO 5	CLO 16	AIT004.16
2	What are the main objectives of code optimization?	a)The optimization must be correct, it must not, in any way, change the meaning of the program. b)Optimization should increase the speed and performance of the program. c)The compilation time must be kept reasonable. d)The optimization process should not delay the overall compiling process.	Understand	CO 5	CLO 16	AIT004.16
3	List the types of code optimization	The optimization process can be broadly classified into two types a)Machine-Independent Optimization b)Machine-Dependent Optimization	Remember	CO 5	CLO 16	AIT004.16
4	What are the code	Techniques are: a)Compile Time Evaluation	Remember	CO 5	CLO 16	AIT004.16

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
	optimization techniques?	b)Common sub-expression elimination c)Dead Code Elimination Code Movement d)Strength Reduction				
5	Define Machine independent optimization	In this optimization, the compiler takes in the intermediate code and transforms a part of the code that does not involve any CPU registers and/or absolute memory locations.	Understand	CO 5	CLO 17	AIT004.17
6	Define Machine dependent optimization	Machine-dependent optimization is done after the target code has been generated and when the code is transformed according to the target machine architecture. It involves CPU registers and may have absolute memory references rather than relative references.	Remember	CO 5	CLO 20	AIT004.20
7	List the different loop optimization techniques.	There are three techniques: a)Code motion b)Induction-variable elimination c)Reduction in strength	Understand	CO 5	CLO 16	AIT004.16
8	What are function preserving transformations?	The transformations are Sources of Optimization a)Common Subexpression Elimination b)Copy Propagation c)Dead-Code Elimination d) Constant Folding	Understand	CO 5	CLO 17	AIT004.17
9	What is common sub expression elimination?	An occurrence of an expression E is called a common sub expression .if E was previously computed, and the values of variables in E have not changed since the previous computation we can avoid recomputing the expression if we can use the previously computed value.	Remember	CO 5	CLO 17	AIT004.17
10	Define copy propagation.	Assignments of the form f : = g called copy statements, or copies for short. The idea behind the copy-propagation transformation is to use g for f, whenever possible after the copy statement f: = g. Copy propagation means use of one variable instead of another.	Remember	CO 5	CLO 17	AIT004.17
11	What is Dead code elimination?	A variable is live at a point in a program if its value can be used subsequently; otherwise, it is dead at that point. A related idea is dead or useless code, statements that compute values that never get used.	Remember	CO 5	CLO 17	AIT004.17

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
12	What is constant folding?	Deducing at compile time that the value of an expression is a constant and using the constant instead is known as constant folding.	Remember	CO 5	CLO 17	AIT004.17
13	What is loop optimization?	In loops, especially in the inner loops, programs tend to spend the bulk of their time. The running time of a program may be improved if the number of instructions in an inner loop is decreased, even if we increase the amount of code outside that loop.	Remember	CO 5	CLO 16	AIT004.16
14	Define code motion.	Code motion is used to decrease the amount of code in loop. This transformation takes a statement or expression which can be moved outside the loop body without affecting the semantics of the program	Remember	CO 5	CLO 16	AIT004.16
15	Define Reduction in strength.	Strength reduction is used to replace the expensive operation by the cheaper once on the target machine. Addition of a constant is cheaper than a multiplication. So we can replace multiplication with an addition within the loop.	Remember	CO 5	CLO 16	AIT004.16
16	What is Basic Block?	A number of sequences are included in the source codes, which are executed in sequence and are termed as the basic blocks of the code. When the first instruction is executed, all the instructions of the same basic block are executed in the sequence of appearance by not losing the program flow control.	Understand	CO 5	CLO 17	AIT004.17
17	What are the characteristics of basic block?	a) They do not contain any kind of jump statements in them. b) There is no possibility of branching or getting halt in the middle. c) All the statements execute in the same order they appear. d) They do not lose the flow control of the program.	Remember	CO 5	CLO 17	AIT004.16
18	Define Flow graph.	A flow graph is a directed graph with flow control information added to the basic blocks.	Remember	CO 5	CLO 17	AIT004.17
19	How to determine basic block.	a)All the statements that follow the leader (including the leader) till the next leader appears form one basic block. b)The first statement of the codeis called as the first leader. c)The block containing the first leader is called as Initial block.	Understand	CO 5	CLO 17	AIT004.16

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
20	How to determine leader statement in basic block.	a)First statement of the code. b)Statement that is a target of the conditional or unconditional goto statement. c) Statement that appears immediately after a goto statement.	Understand	CO 5	CLO 17	AIT004.17
21	Define Induction variable.	A variable x is an Induction Variable of a loop if every time the variable x changes values, it is incremented or decremented by some constant	Remember	CO 5	CLO 20	AIT004.20
22	List the types of basic block optimization.	There are two type of basic block optimization. a)Structure-Preserving Transformations b)Algebraic Transformations	Remember	CO 5	CLO 17	AIT004.17
23	List the Structure-Preserving Transformation on basic blocks.	a)Common sub-expression elimination b)Dead code elimination c)Renaming of temporary variables d)Interchange of two independent adjacent statements	Understand	CO 5	CLO 17	AIT004.17
24	What are Algebraic transformations?	In the algebraic transformation, we can change the set of expression into an algebraically equivalent set. Thus the expression x:= x + 0 or x:= x *1 can be eliminated from a basic block without changing the set of expression.	Remember	CO 5	CLO 20	AIT004.20
25	What is Directed Acyclic graph?	Directed Acyclic Graph (DAG) is a tool that depicts the structure of basic blocks, helps to see the flow of values flowing among the basic blocks, and offers optimization too. DAG provides easy transformation on basic blocks.	Remember	CO 5	CLO 20	AIT004.20
26	What is code generation	This is the final phase of compilation. which takes input as a optimized code and convert in to machine/assembly language.	Remember	CO 5	CLO 18	AIT004.18
27	List the code generator descriptors.	There are two descriptors: a)Register descriptor b)Address descriptor	Remember	CO 5	CLO 18	AIT004.18
28	What is peephole optimization?	This optimization technique works locally on the source code to transform it into an optimized code. By locally, we mean a small portion of the code block at hand. These methods can be applied on intermediate codes as well as on target codes.	Understand	CO 5	CLO 18	AIT004.18
29	What is Unreachable code?	Unreachable code is a part of the program code that is never accessed because of programming constructs.	Remember	CO 5	CLO 18	AIT004.18

S.No	QUESTION	ANSWER	Blooms Level	CO	CLO	CLO Code
		Programmers may have accidently written a piece of code that can never be reached				
30	What is register allocation and assignment?	Selecting the set of variables that will reside in registers at each point in the program Picking the specific register that a variable will reside in it.	Understand	CO 5	CLO 20	AIT004.20

Signature of the Faculty **HOD, IT**