

--	--	--	--	--	--	--	--

## B.Tech. Degree VI Semester Regular Examination April 2022

### CS 19-202-0602 COMPILER CONSTRUCTION (2019 Scheme)

Time: 3 Hours

Maximum Marks: 60

#### Course Outcome

On successful completion of the course, the students will be able to:

- CO1: Summarize the functionality of each phase involved in compilation process.  
 CO2: Develop scanner and parser using lex and yacc tools.  
 CO3: Design top down parsers including recursive descent parser and non-recursive predictive parser for CFGs.  
 CO4: Design bottom up parsers including shift reduce, operator precedence and LR parsers (SLR, CLR and LALR).  
 CO5: Explain Syntax directed translation using S-attributed definition and L-attributed definition.  
 CO6: Familiarize specification for a type checker and run time environment.  
 CO7: Comprehend different representations of intermediate code.  
 CO8: Describe various code optimization techniques to improve the performance of a program and learn code generation techniques.

Bloom's Taxonomy Levels (BL): L1 – Remember, L2 – Understand, L3 – Apply, L4 – Analyze,

L5 – Evaluate, L6 – Create

PO – Programme Outcome



#### PART A

(Answer *ALL* questions)

I.		(8 × 3 = 24)	Marks	BL	CO	PO
(a)	The scanner fits in between the input interface and the parser. Comment the statement with three valid points.		3	L3	1	1
(b)	Develop the regular expression for an identifier which starts with an alphabet and may contain alphabets and/or digits. Also develop a Lex program to count the number of identifiers.		3	L4	2	4
(c)	Distinguish between recursive descent parser and non-recursive predictive parser.		3	L2	3	1
(d)	With a suitable example briefly explain handle and handle pruning?		3	L1	4	1
(e)	What do you understand by syntax direct definition?		3	L1	5	1
(f)	Mention three phases of storage management process.		3	L1	6	1
(g)	Translate the following expression to quadruple and triples $a = b * -c + b * -c$		3	L4	7	2
(h)	Describe any three issues in the design of a code generator.		3	L1	8	1

#### PART B

(4 × 12 = 48)

II.	Summarize the functionality of each phase involved in compilation process.	12	L1	1	1
<b>OR</b>					
III.	Explain input buffering techniques with neat diagrams.	12	L1	1	1
IV.	Consider the following grammar: $S \rightarrow AA$ $A \rightarrow aA$ $A \rightarrow b$ And construct the LALR parsing table.	12	L3	4	3

**OR**

(P.T.O)

BTS-VI(R)-04-22-0286

V.	Construct the operator precedence table for the following grammar: $E \rightarrow E + T / T$ $T \rightarrow T * F / F$ $F \rightarrow id$	12	L3	4	3
VI.	Explain the specification of a simple type checker.	12	L1	6	1
<b>OR</b>					
VII.	(a) Differentiate the storage allocation strategies - static, stack and heap allocation.	6	L2	6	1
	(b) With a neat sketch briefly describe the typical subdivision of run time memory.	6	L1	6	1
VIII.	Explain with suitable example the following optimization: (i) Common sub expression elimination (ii) Copy propagation (iii) Dead-code elimination (iv) Constant folding.	12	L1	8	1
<b>OR</b>					
IX.	Describe three techniques that are used for loop optimization.	12	L1	8	1

Blooms's Taxonomy Levels

L1 = 58.82%, L2 = 11.76%, L3 = 17.65%, L4 = 11.76%

\*\*\*