Code No: R1631051 **R16** SET - 1

III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019

COMPILER DESIGN

(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A (14 Marks)

1.	a)	Differentiate between token, lexeme and pattern with examples.	[3M]
	b)	Name the three techniques for constructing LR parsing table.	[2M]
	c)	What are the actions performed by Shift reduce parser?	[2M]
	d)	Define abstract Syntax tree.	[2M]
	e)	Write about the sub-division of run-time memory.	[3M]
	f)	List the characteristics of peephole optimization.	[2M]

PART –B (56 Marks)

2.	a)	Explain various data structures used for lexical analysis.	[7M]
	b)	Discuss in brief about Bootstrapping process with suitable diagram.	[7M]
3.	a)	Define Context Free Grammar. Explain how it is suitable for parsing?	[7M]
	b)	Write an algorithm to find LR(0) items and give an example.	[7M]
4.	a)	Design LALR(1) parser for the following grammar: S -> aAd | bBd | aBc |bAc A -> e B -> e where a, b, c, d, e are terminals.	[7M]
	b)	Give the SDT scheme for desk calculator.	[7M]
5.	a)	Explain the construction of syntax tree for expressions.	[7M]
	b)	Discuss the concept of back patching with an example.	[7M]
6.	a)	What is a flow graph? Explain how flow graph can be constructed for a given program.	[7M]
	b)	Discuss the advantages and disadvantages of heap storage allocation strategy.	[7M]
7.	a)	Distinguish between machine dependent and machine independent optimization.	[7M]
	b)	Explain the algebraic transformations of local machine independent optimization.	[7M]

*****
~~~~~~

Code No: R1631051 **R16** SET - 2

III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019

COMPILER DESIGN

(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A (14 Marks)

1.	a)	Give the types of a language processing system.	[2M]
	b)	What are the problems in top-down parsers?	[2M]
	c)	Write the applications of SDTs.	[3M]
	d)	What is Static Checking? List out some examples of static checks.	[3M]
	e)	Mention the issues in design of code generation.	[2M]
	f)	Write the criteria for achieving machine independent code optimization.	[2M]

PART –B (56 Marks)

2.	a)	Define compiler. Describe the phases of a compiler with a neat sketch.	[7M]
	b)	Explain the recognition of keywords and identifiers with a suitable transition diagram.	[7M]
3.	a)	Eliminate ambiguities in the following grammar: S -> iEtS| iEtSeS| a E -> b| c | d	[7M]
	b)	Define LR(k) parser. Draw and explain the model of LR parser.	[7M]
4.	a)	How to detect and reduce handle in LR parsers? Explain.	[7M]
	b)	Construct LALR parsing table for the following grammar: S -> CC C -> cC | d	[7M]
5.	a)	What is the role of type checking in error detection and error recovery? Explain.	[7M]
	b)	Give the translation scheme to convert an expression grammar into three address code.	[7M]
6.	a)	Explain the activities of *caller* and *callee* in stack allocation strategy with an example. What is the role of parameter passing in it?	[7M]
	b)	How to access non-local data? Explain with example.	[7M]
7.	a)	Discuss about the principal sources of optimization with examples.	[7M]
	b)	Explain in brief about the DAG based local optimization.	[7M]

*****
~~~~~~

Code No: R1631051 **R16** SET - 3

III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019

COMPILER DESIGN

(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A (14 Marks)

1.	a)	What are the functions of preprocessor?	[2M]
	b)	Differentiate between top-down parser and bottom-up parser.	[3M]
	c)	Give the usage of look ahead symbol in LALR parsing.	[2M]
	d)	How to generate polish notation using translation schemes?	[2M]
	e)	What are the advantages and disadvantages of heap storage allocation strategies for records?	[3M]
	f)	Write the factors that affects the target code generation.	[2M]

PART –B (56 Marks)

2.	a)	Describe the functionality of compilers in a typical language processing system.	[7M]
	b)	Explain how input buffering helps lexical analyzer in compilation process.	[7M]
3.	a)	Design a non-recursive predictive parser for the following grammar: S -> AaAb | BbBb A -> e B -> e where a, b, e are terminals.	[7M]
	b)	What kinds of source program errors would be detected during lexical analysis? Explain.	[7M]
4.	a)	State and explain the rules used to construct the LR(1) items.	[7M]
	b)	Discuss the evolution order of SDTs. Also write its applications.	[7M]
5.	a)	What is a three address code? What are its types? How it is implemented?	[7M]
	b)	What are the one-pass code generation methods? Explain any one.	[7M]
6.	a)	Generate the flow-graphs for the following expressions: S -> id: = E | S;S | if E then S else S | do S while E. E -> id + id | id	[7M]
	b)	Why garbage collection is important for code optimization? Explain garbage collection by using reference counting.	[7M]
7.	a)	Prove that simple code generation algorithms allocate the registers efficiently.	[7M]
	b)	Discuss the transformations that are characteristic of peephole optimizations.	[7M]

*****
~~~~~~

Code No: R1631051 **R16** SET - 4

III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2019

COMPILER DESIGN

(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A (14 Marks)

1. a) Define Boot strapping. [2M]
   b) Give the rules to find the first function. [2M]
   c) List the properties of LR parser. [2M]
   d) How is object code different from intermediate code generation? [3M]
   e) Write the limitations of access links. [3M]
   f) What is peephole? What peephole optimizations can be performed on code? [2M]

PART –B (56 Marks)

2. a) Consider a hypothetical programming language that has only integer and floating point's constants, data declaration and assignment statements. Describe in detail the steps involved in design of lexical analyzer for this language. [7M]
   b) Describe the need and functionality of linkers, assemblers and loaders. [7M]

3. a) What is an LL(1) grammar? Can you convert every context free grammar into LL(1). [7M]
   b) Consider the following grammar [7M]
   E -> T + E|T
   T -> V*T|V
   V -> id
   Write down the procedures for the non-terminals of the grammar to make a recursive descent parser.

4. a) Justify how LALR parsing is efficient over SLR parsing. [7M]
   b) Write the procedure for eliminating left recursion from SDTs. [7M]

5. a) What is the role of type system in type checker? Write the syntax directed definition for type checker. [7M]
   b) Explain the steps involved in partitioning a sequence of three address statements into basic blocks. [7M]

6. a) What is runtime stack? Explain the storage allocation strategies used for recursive procedure calls. [7M]
   b) Discuss about register allocation and assignment in target code generation. [7M]

7. a) Explain the structure preserving transformations of local machine independent optimization. [7M]
   b) What is Data flow Analysis? Explain its role in code optimization. [7M]

*****
~~~~~~