

ST.ANN'S COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
LESSON PLAN

Subject: COMPILER DESIGN
Name of Faculty: M. Lakshmi Bai

Year/ Sem: III-I Semester 'B'
Academic Year: 2017-2018

S.No.	UNIT	TOPICS	No.of Classes Required
1	I	Overview of language processing – pre-processors – compiler – assembler – interpreters, pre-processors, – linkers & loaders - structure of a compiler – phases of a compiler . Lexical Analysis – Role of Lexical Analysis– Lexical Analysis Vs. Parsing – Token, patterns and Lexemes – Lexical Errors – Regular Expressions – Regular definitions for the language constructs – Strings, Sequences, Comments – Transition diagram for recognition of tokens, Reserved words and identifiers, Examples.	10
2	II	Syntax Analysis – discussion on CFG, LMD,RMD, parse trees, Role of a parser – classification of parsing techniques – Brute force approach, left recursion, left factoring, Top down parsing – First and Follow- LL(1) Grammars, NonRecursive predictive parsing – Error recovery in predictive parsing.	07
3	III	What is bottom up parsing approach , Types of Bottom up approaches; Introduction to simple LR – Why LR Parsers – Model of an LR Parsers – Operator Precedence- Shift Reduce Parsing – Difference between LR and LL Parsers, Construction of SLR Tables. More powerful LR parses , construction of CLR (1), LALR Parsing tables, Dangling ELSE Ambiguity, Error recovery in LR Parsing. Comparison of all bottoms up approaches with all top down approaches.	11
4	IV	Semantic analysis , SDT Schemes, evaluation of semantic rules. Intermediate code, three address code, quadruples, triples, abstract syntax trees. Types and declarations, type Checking.	05
5	V	Symbol tables: use and need of symbol tables. Runtime Environment: storage organization, stack allocation, access to non-local data, heap management, parameter passing mechanisms, introduction to garbage collection. Reference counting garbage collectors. Code generation: Issues, target language, Basic blocks & flow graphs, Simple code generator, Peephole optimization, Register allocation and assignment.	08
6	VI	Machine independent code optimization – semantic preserving transformations, global common sub expression elimination, copy propagation, dead code elimination, constant folding, strength reduction, loop optimization. Instruction scheduling, inter procedural optimization.	07
TOTAL CLASSES			48

Text Books:

1. Compilers, Principles Techniques and Tools- Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D. Ullman, 2nd ed, Pearson, 2007.
2. Compiler Design, K. Muneeswaran, Oxford.

Reference Books:

1. Engineering a compiler, 2nd edition, Keith D.Cooper & Linda Torczon, Morgan Kaufman.
2. <http://www.nptel.iitm.ac.in/downloads/106108052/>
3. Principles of compiler design, V. Raghavan, 2nd ed, TMH, 2011.
4. Compiler construction, Principles and Practice, Kenneth C Loudon, CENGAGE.
5. Implementations of Compiler, A new approach to Compilers including the algebraic methods, Yunlinsu, SPRINGER.

FACULTY

HEAD OF THE DEPARTMENT

SACRET