I Year II Semester	**L**	**P**	**C**
Code: 17CS253	**4**	**0**	**3**

COMPILERDESIGN
(Elective–1)

UNIT–I

Introduction Language Processing, Structure of a compiler the evaluation of Programming language, The Science of building a Compiler application of Compiler Technology. Programming Language Basics.

Lexical Analysis-: The role of lexical analysis buffing, specification of tokens. Recognitions of tokens the lexical analyzer generator lexical

UNIT–II

Syntax Analysis -: The Role of a parser, Context free Grammars Writing A grammar, top down passing bottom up parsing Introduction to Lr Parser.

UNIT–III

More Powerful LR parser (LR1, LALR) Using Armigers Grammars Equal Recovery in Lr parser Syntax Directed Transactions Definition, Evolution order of SDTS Application of SDTS. Syntax Directed Translation Schemes.

UNIT–IV

Intermediated Code: Generation Variants of Syntax trees 3 Address code, Types and Deceleration, Translation of Expressions, Type Checking. Canted Flow Back patching?

UNIT–V

Run time Environments, Stack allocation of space, access to Non Local date on the stack Heap Management code generation – Issues in design of code generation the target Language Address in the target code Basic blocks and Flow graphs. A Simple Code generation.

TEXTBOOKS:

1. Compilers, Principles Techniques and Tools. Alfred VAho, MonicalS. Lam, RaviSethi Jeffery
   D.Ullman, 2ndedition, pearson, 2007

2. Compiler Design K. Muneeswaran, OXFORD

3. Principles of compiler design, 2nd edition, Nandhini Prasad, Elsebier.

REFERENCEBOOKS:

1. Compiler Construction, Principles and practice, Kenneth CLouden, CENGAGE
2. Implementations of Compiler, A New approach to Compilers including the algebraic methods, Yunlinsu, SPRINGER