3. Bernd Jahne, **Digital Image Processing**, (5th revised and extended edition), Springer, 2002

4. S. Annadurai and R. Shanmugalakshmi, **Fundamentals of Digital Image Processing**, Pearson Education, 2007

5. M.A. Joshi, **Digital Image Processing: An Algorithmic Approach**, Prentice-Hall of India, 2006

6. B. Chanda and D.D. Majumder, **Digital Image Processing and Analysis**, Prentice-Hall of India, 2007

MCSE202: COMPILER DESIGN

Course Objectives: The course aims to develop the ability to design, develop, and test a functional compiler/ interpreter for a subset of a popular programming language.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe how different phases of a compiler work.

CO2: implement top-down and bottom-up parsing algorithms.

CO3: use tools like Lex and Yacc to implement syntax-directed translation.

Syllabus:

Unit- I Lexical and Syntactic Analysis: Review of regular languages, design of a lexical analyzer generator, context-free grammars, syntactic analysis: top-down parsing: recursive descent and predictive parsing, LL(k) parsing; bottom-up parsing: LR parsing, handling ambiguous in bottom-up parsers.

Unit-II Syntax directed translation: Top-down and bottom-up approaches, data types, mixed mode expression; subscripted variables, sequencing statement, subroutines and functions: parameters calling, subroutines with side effects.

Unit-III Code generation, machine dependent and machine-independent optimization techniques.

Readings:

1. Alfred V. Aho, Ravi Sethi, D. Jeffrey Ullman, Monica S. Lam, **Principles, Techniques and Tools**, Pearson Education India, 2nd edition,, 2013.
2. A.V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, **Compilers,** Pearson, 2016.
3. Dick Grune, Kees van Reeuwijk, Henri E .Bal, Ceriel J.H. Jacobs, K Langendoen, **Modern Compiler Design,** Springer, 2012.

MCSE 203: NATURAL LANGUAGE PROCESSING [3-0-1]