Course Title	**CS-850,Advanced Theory of Computation**
Credit Hours	3+0
Prerequisites:	▪ Software Construction ▪ Data Structures ▪ Computing Algorithms
Course Description:	Structure of programming languages and their implementation. Basic language design principles; abstract data types; functional languages; type systems; object-oriented languages. Basics of lexing, parsing, syntax-directed translation, semantic analysis, and code generation.
Tools and Technologies:	▪ OCaml and parser generators such as ANTLR
Learning Outcomes:	Students taking this course can expect to acquire the following: 1. an understanding of the major classes of high-level programming languages, language features, and programming styles, with an emphasis on applying concepts from programming language theory; 2. formal methods of specifying the syntax and semantics of programming languages; 3. and the knowledge needed to write parsers, interpreters, and simple compilers for the major classes of programming languages.
Text Books:	*Compilers: Principles, Techniques, and Tools*, also known as "The Dragon Book"; by Aho, Sethi, and Ullman. Published by Addison-Wesley.
Reference Books:	▪ The Objective Caml system, release 3.11 Documentation and user's manual by Xavier Leroy (with Damien Doligez, Jacques Garrigue, Didier Vouillon), from the official INRIA website for OCAML. ▪ Essentials of Programming Languages, 2nd Edition; by Friedman, Wand, and Haynes. Published by MIT Press 2001. ISBN: 0-262-06217-8. ▪ Advanced Programming Language Design, by Raphael A. Finkel. Addison Wesley Publishing Company, 1996. ▪ Programming Language Pragmatics, by Michael L. Scott. Morgan Kaufman Publishers, 2000.
Course Contents:	▪ Automata theory ▪ Language Semantics, Compilers, and Applications ▪ Regular Expressions and Lexical Analysis ▪ Recursive Descent Parsing (LL Parser), LR Parser ▪ Formal Grammars ▪ Translation and Code Generation ▪ Data Flow Analysis and Code Optimization