Teaching Scheme of B.Tech.-III (CSE) (Semester VI)

Sr. No.	Course	Code	Credit	Teaching Scheme			Examination Scheme			Total
				L	**T**	**P**	**L**	**T**	**P**	
1	Principles of Programming Languages (Core-11)	CS302	5	3	1	2	100	25	50	175
2	Distributed Systems (Core-12)	CS304	5	3	1	2	100	25	50	175
3	System Software (Core-13)	CS306	5	3	1	2	100	25	50	175
4	Artificial Intelligence (Core-14)	CS308	4	3	0	2	100	0	50	150
5	Institute Elective-2	-	3	3	0	0	100	0	0	100
6	Core Elective-2	CS3BB	3	3	0	0	100	0	0	100
	Total		**25**	**18**	**3**	**8**	**600**	**75**	**200**	**875**
	Total Contact Hours per week			**29**						

Practical Examination Scheme (Internal 50% and External 50%)

Institute Elective-2 (CS3YY):

1	Cryptography (CS362)	4	Image Processing (CS368)
2	Digital Forensics (CS364)	5	Adaptive Signal Processing (CS372)
3	Embedded Systems (CS366)		

Core Elective-2 (CS3BB):

1	Data Visualization (CS322)	4	Wireless Networks (CS328)
2	Natural Language Processing (CS324)	5	Optimization Methods (CS332)
3	Cloud Computing (CS326)		

B.Tech. III (CSE) Semester – VI
PRINCIPLES OF PROGRAMMING LANGUAGES (CORE-11)
CS302

	L	T	P	Credit
Scheme	**3**	**1**	**2**	**05**

1. Course Outcomes (COs): **At the end of the course, students will be able to**	
CO1	understand language features of current programming languages.
CO2	program in different language paradigms and evaluate their relative benefits.
CO3	analyze object oriented constructs in different programming languages.
CO4	evaluate the programming solutions of different problems.
CO5	design programs in Functional and Logical Languages.

2. Syllabus

- **INTRODUCTION** **(06 Hours)**

  Introduction: Role of Programming Languages: Why Programming Languages, Towards Higher-Level Languages, Programming Paradigms, Programming Environments Language Description: Syntactic Structure, Language Translation Issues: Programming Language Syntax, Stages in Translation, Formal Translation Models.

- **BASICS OF PROGRAMMING LANGUAGE** **(08 Hours)**

  Data, Data Types, and Basic Statements: Names, Variables , Binding, Type Checking, Scope, Scope Rules , Lifetime and Garbage Collection, Primitive Data Types, Strings, Array Types, Associative Arrays ,Record Types, Union Types, Pointers and References , Arithmetic Expressions , Overloaded Operators, Type Conversions , Relational and Boolean Expressions, Assignment Statements, Mixed Mode Assignments, Control Structures, Selection ,Iterations, Branching, Guarded Statements.

- **SUBPROGRAMS** **(08 Hours)**

  Subprograms and Implementations: Subprograms, Design Issues, Local Referencing, Parameter Passing, Overloaded Methods, Generic Methods, Design Issues for Functions, Semantics of Call and Return, Implementing Simple Subprograms, Stack and Dynamic Local Variables, Nested Subprograms, Dynamic Scoping.

- **OBJECT-ORIENTED PROGRAMING** **(10 Hours)**

  Object-Orientation, Concurrency, and Event Handling: Grouping of Data and Operations - Constructs for Programming Structures, Abstraction Information Hiding, Program Design with Modules, Defined Types, Object Oriented Programming - Concept of Object, Inheritance, Derived Classes and Information Hiding - Templates, Semaphores, Monitors, Message Passing, Threads, Statement Level Concurrency Exception Handling (Using C++ and Java as Example Language).

- **FUNCTIONAL AND LOGIC PROGRAMMING LANGUAGES:** **(10 Hours)**

Introduction to Lambda Calculus, Fundamentals of Functional Programming Languages, Programming with Programming with ML, Introduction to Logic and Logic Programming - Programming with Prolog. Types of Logic, WFF, Symbolic Logic, Facts, Clauses, Predicates, Unification, Backtracking, Cut, Fail & Built-In Predicates, Recursion in Prolog, Arithmetic Operators & Relational Operators, LIST Processing, String manipulation & Built-In Predicates, Compound Objects, Dynamic Database.

Tutorials will be based on topics discussion in the class (14 Hours)

Practicals will be based on topics discussion in the class (28 Hours)

(Total Contact Time: 42 Hours+14 Hours+28 Hours= 84 Hours)

---

3. Tutorials:

1 Programming languages paradigm.

2 Study of programming language and its benefits. Success and Failure of language.

3 Prolog programming.

4 Object oriented programming constructs.

5 Mapping complex problems with available technologies and evaluate its usefulness.

4. Practicals:

1 Convert prolog predicates into semantic net.

2 Implement travelling salesman problem using prolog.

3 Implement 8 puzzle problem using prolog.

4 Implement N-Queens problem using prolog.

5 Implement C++/Java program for class & object, constructor & destructor.

6 Implement C++/Java programs for operator overloading, inheritance, and polymorphism, file operation.

7 Implement of string operation using prolog.

8 Implement of artificial intelligence based application using prolog.

5. Books Recommended:

1. Terrance W.Pratt, Marvin V. Zelkowitz, T.V.Gopal, “Programming Languages: Design and Implementations”, Fourth ed., Prentice Hall, Sep 7, 2000.
2. David A. Watt, “Programming Language Design Concept”, 1st Edition, Willey India, Jan 1, 2009.
3. Ravi Sethi, “Programming languages: Concepts and Constructs”, Second Ed., Pearson, Jan 7, 1996.
4. Benjamin C. Pierce, “Types and programming Languages”, The MIT Press Cambridge, Massachusetts, London, England, Jan 4, 2002.
5. Robert W. Sebesta, Concepts of Programming Languages, 11th Ed., Pearson, Feb 16, 2015.

B.Tech. III (CSE) Semester – VI
DISTRIBUTED SYSTEMS (CORE-12)
CS304

	L	T	P	Credit
Scheme	**3**	**1**	**2**	**05**

1. Course Outcomes (COs): **At the end of the course, students will be able to**	
CO1	understand the concepts of distributed System and design and implementation issues.
CO2	define key mechanism for designing distributed algorithms for different primitives like mutual exclusion, deadlock detection, agreement etc.
CO3	analyze different types of faults and fault handling techniques in order to implement fault tolerant systems.
CO4	correlate different election algorithm, file system, time synchronization and naming services.
CO5	design and develop distributed programs subject for specific design and performance constraints.

2. Syllabus

- **INTRODUCTION TO DISTRIBUTED SYSTEMS** **(04 Hours)**

  Review of Networking Protocols, Point to Point Communication, Operating Systems, Concurrent Programming, Characteristics and Properties of Distributed Systems, Goals of Distributed Systems, Multiprocessor and Multicomputer Systems, Distributed Operating Systems, Network Operating Systems, Middleware Concept, The Client-Server Model, Design Approaches-Kernel Based-Virtual Machine Based, Application Layering.

- **COMMUNICATIONIN DISTRIBUTED SYSTEMS** **(04 Hours)**

  Layered Protocols, Message Passing-Remote Procedure Calls-Remote Object Invocation, Message Oriented Communication, Stream Oriented Communication, Case Studies.

- **PROCESS MANAGEMENT** **(04 Hours)**

  Concept of Threads, Process, Processor Allocation, Process Migration and Related Issues, Software Agents, Scheduling in Distributed System, Load Balancing and Sharing Approaches, Fault Tolerance, Real Time Distributed System.

- **SYNCHRONIZATION** **(06 Hours)**

  Clock Synchronization, Logical Clocks, Global State, Election Algorithms-The Bully algorithm-A Ring algorithm, Mutual Exclusion-A Centralized Algorithm-A Distributed Algorithm-A token ring Algorithm, Distributed Transactions.

- **CONSISTENCY AND REPLICATION** **(06 Hours)**

  Introduction to Replication, Object Replication, Replication as Scaling Technique, Data Centric Consistency Models-Strict-Linearizability and Sequential-Causal-FIFO-Weak-release-Entry, Client Centric Consistency Models-Eventual Consistency-Monotonic Reads and Writes-Read your Writes-Writes Follow Reads, Implementation Issues, Distribution Protocols-Replica Placement-Update Propogation-Epidemic Protocols, Consistency Protocols.

- **FAULT TOLERANCE** **(04 Hours)**

  Introduction, Failure Models, Failure Masking, Process Resilience, Agreem in Faulty Systems, Reliable Client Server communication, Group communication, Distributed Commit, Recovery.

- **DISTRIBUTED OBJECT BASED SYSTEMS** **(06 Hours)**

  Introduction to Distributed Objects, Compile Time Vs Run Time Objects, Persistent and Transient Objects, Enterprise JAVA Beans, Stateful and Stateless Sessions, Global Distributed Shared Objects, Object Servers, Object Adaptors, Implementation of Object References, Static And Dynamic Remote Method Invocations, Replica Framework.

- **DISTRIBUTED FILE SYSTEMS** **(04 Hours)**

  Introduction, Architecture, Mechanisms for Building Distributed File Systems-Mounting-Caching-Hints-Bulk Data Transfer-Encryption, Design Issues-Naming and Name Resolution-Caches on Disk or Main Memory-Writing Policy-Cache consistency-Availability-Scalability-Semantics, Case Studies, Log Structured File Systems.

- **DISTRIBUTED WEB BASED SYSTEMS** **(04 Hours)**

  Architecture, Processes, Communication, Naming, Synchronization, Web Proxy Caching, Replication of Web Hosting Systems, Replication of Web Applications.

Practicals will be based on the coverage of the above topics. **(28 Hours)**

Tutorials will be based on the coverage of the above topics. **(28 Hours)**

(Total Contact Time 42 Hours + 14 Hours + 28 Hours = 84 Hours)

---

3. <u>Practicals:</u>

1 Implementation of concepts of communication protocols using UDP and TCP IP.

2 Implement the remote procedure call with an application.

3 Implementation of object based system using RMI or CORBA.

4 Implementation of distributed system for file sharing and message passing.

5 Implementation of Socket programming.

6 Implementation of distributed client-server application.

7 Implementation of client-server application with scheduling in distributed environment.

8 Implementation of distributed load balancing and resource sharing.

4. Tutorials:

1 Concepts of communications (UDP and TCP IP).

2 Concepts of fault tolerance.

3 Concept of time Synchronization.

4 Concepts of process management.

5 Concepts of replication and consistency.

6 Object based system (RMI and CORBA).

5. Books Recommended:

1. Andrew S Tanenbaum, “Distributed systems: Principles and Paradigms”, Second Edition, Pearson Education. Inc 2007.
2. Mukesh Singhal and Niranjan G. Shivaratri, “Advanced Concepts in Operating Systems”, TMH, McGraw-Hill, Inc. New York, USA 1994.
3. Pradeep K. Sinha, “Distributed Operating System: Concept and design”, PHI, New Delhi 2019.
4. W Richard Stevens, “Unix Network Programming: Vol 1, Networking APIS: Sockets & XTI”, Second Edition E, Pearson Education, 1998.
5. Colouris, Dollimore, Kindberg, "Distributed Systems Concepts & Design", Fourth Edition, Pearson Ed. 2005.

B.Tech. III (CSE) Semester – VI
SYSTEM SOFTWARE (CORE – 13)
CS306

	L	T	P	Credit
Scheme	**3**	**1**	**2**	**05**

1. Course Outcomes (COs): **At the end of the course, students will be able to**	
CO1	understand systems software components, finite automata, regular expression and context free grammar.
CO2	apply the knowledge of assembler and macro processors to convert assembly language into machine code.
CO3	analyse working phases of Compiler, various parsing techniques, semantic analysis, Error handling, code generation and code optimization techniques to undertake meaningful language translation.
CO4	evaluate Linkers, Loaders, interpreters and debugging methods to manages system memory and provide a portable runtime environment.
CO5	create a language translator application and mimic a simple compiler.

2. Syllabus

- **INTRODUCTION** **(04 Hours)**

  Introduction to System Software, Utility Software, Systems Programming, Recent Trends in Software Development, Programming Languages and Language Processors, Data Structures for Language Processing.

- **ASSEMBLERS** **(06 Hours)**

  Overview of the Assembly Process, Cross Assembler, Micro Assembler, Meta Assembler, Single Pass Assembler, Two Pass Assembler, Design of Operation Code Table, Symbol Table, Literal Table, Advanced Assembly Process .

- **MACRO PROCESSORS** **(06 Hours)**

  Introduction of Macros, Macro Processor Design, Forward Reference, Backward Reference, Positional Parameters, Keyword Parameters, Conditional Assembly, Macro Calls within Macros, Implementation of Macros Within Assembler. Designing Macro Name Table, Macro Definition Table, Kew Word Parameter Table, Actual Parameter Table, Expansion Time Variable Storage.

- **COMPILERS** **(14 Hours)**

  Phases of Compiler, Analysis-Synthesis Model of Compilation, Interface with Input, Parser and Symbol Table, Token, Lexeme, Patterns and Error Reporting in Lexical Analysis, Programming

Language Grammars, Classification of Grammar, Ambiguity in Grammatical Specification, Top Down Parsing, Recursive Descent Parsing, Transformation on The Grammars, Predictive Parsing, Bottom Up Parsing, Operator Precedence Parsing, LR Parsers, Language Processor Development Tools – LEX & YACC, Semantic Gap, Binding and Binding Times, Memory Allocation, Compilation of Expression, Intermediate Representations, Basic Code Optimization.

- **LINKERS AND LOADERS** **(06 Hours)**

  Design of a Linker, Program Relocation, Linking of Overlay Structured Programs, Dynamic Linking, General Loader Schemes, Absolute Loader, Relocating Loader, Dynamic Loader, Bootstrap Loader, Linking Loader, other Loading Schemes, Linkers v/s Loaders.

- **INTERPRETERS & DEBUGGERS** **(06 Hours)**

  Overview of Interpretation and Debugging Process, Types of Errors, Classification of Debuggers, Dynamic/Interactive Debugger, The Java Language Environment, Java Virtual Machine and Recent Developments.

- **Tutorials will be based on the coverage of the above topics separately** **(14 Hours)**

(Total Contact Time: 42 Hours + 14 Hours + 28 Hours = 84 Hours)

---

3. **Practicals:**

1 Study, install and setup various system software tools.

2 Implementation of single pass and two pass assembler.

3 Design and implement scanner using lexical analyzer (LEX) tool.

4 Design and implement parser using YACC tools.

5 Design and configure a compiler application using modern tools and softwares.

6 Implementation of different stages of compiler.

7 Implementation of interpreter and debugger.

8 Implementation of optimization based compiler design.

4. **Tutorials**

1 Problem solving on the basics of assembler.

2 Problem solving on the basics of macro processor.

3 Problem solving on the basics of lexical analysis.

4 Problem solving on the basics of parsing.

5 Problem solving on the basics of linkers and loaders.

6 Problem solving on the basics of interpreters & debuggers.

BOOKS RECOMMENDED

1. D. M. Dhamdhere, "Systems Programming", 1/E, McGraw Hill, 2011.
2. Leland L. Beck, "System Software - An Introduction to System Programming", 3/E, Pearson Education, 2002.
3. John Donovan, “Systems programming”, 1/E, McGraw Hill, 2017.
4. Santanu Chattopadhyay, “System Software” 1/E, Prentice-Hall India,2007.
5. A.V.Aho, R.Sethi & J D.Ullman, "Compilers-Principles, Techniques and Tools", 2/E, Pearson India, 2013.

ADDITIONAL REFERENCE BOOKS

1. Allen. Holub, "Compiler Design in C", 1/E, Pearson India, 2015.
2. Ronald Mak, “Writing Compilers and Interpreters: A Software Engineering Approach”, 3/E, Wiley, 2009.

B.Tech. III (CSE) Semester – VI
ARTIFICIAL INTELLIGENCE (CORE-14)
CS308

	L	T	P	Credit
Scheme	**3**	**0**	**2**	**04**

1. Course Outcomes (COs): At end of the program, students will be able to	
CO1	understand the role of agents and how it is related to environment and the way of evaluating it and how agents can act by establishing goals.
CO2	apply various knowledge representation technique, searching techniques, constraint satisfaction problem and example problems- game playing techniques.
CO3	analyse the current scope, potential, limitations, and implications of intelligent systems.
CO4	evaluate the AI techniques suitable for recent areas of applications like expert systems, neural networks, fuzzy logic, robotics, natural language processing, and computer vision.
CO5	design a real world problem for implementation and understand the dynamic behaviour of a system.

2. Syllabus

- **INTRODUCTION TO AI** **(03 Hours)**

  Intelligent Agents, AI Techniques, AI-Problem formulation, AI Applications, Production Systems, Control Strategies.

- **KNOWLEDGE REPRESENTATION** **(06 Hours)**

  Knowledge Representation Using Predicate Logic, Introduction to Predicate Calculus, Resolution, Use of Predicate Calculus, Knowledge Representation Using other Logic-Structured Representation of Knowledge.

- **PRODUCTION SYSTEM** **(06 Hours)**

  Defining the Problems as a State Space Search, Production Systems, Production Characteristics, Production System Characteristics, Forward and Backward, State-Space Search, Problem Solving Methods – Problem Graphs, Matching, Indexing.

- **PROBLEM-SOLVING THROUGH SEARCH** **(06 Hours)**

  Generate and Test, BFS, DFS, Blind, Heuristic, Problem-Reduction, A, A*, AO*, Minimax, Constraint Propagation, Neural, Stochastic, and Evolutionary Search Algorithms, Sample Applications, Measure of Performance and Analysis of Search Algorithms, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis, Issues in the Design of Search Programs.

- **KNOWLEDGE INFERENCE** **(06 Hours)**

  Knowledge Representation -Production Based System, Frame Based System. Inference – Backward Chaining, Forward Chaining, Rule Value Approach, Fuzzy Reasoning – Certainty Factors, Bayesian Theory-Bayesian Network-Dempster – Shafer Theory. Symbolic Logic Under Uncertainty : Non-Monotonic Reasoning, Logics for Non-Monotonic Reasoning, Statistical Reasoning : Probability and Bayes Theorem, Certainty Factors, Probabilistic Graphical Models, Bayesian Networks, Markov Networks, Fuzzy Logic.

- **GAME PLAYING AND PLANNING** **(06 HOURS)**

  Overview and Example Domain: Overview, Minimax, Alpha-Beta Cut-Off, Refinements, Iterative Deepening, The Blocks World, Components of a Planning System, Goal Stack Planning, Nonlinear Planning Using Constraint Posting, Hierarchical Planning, Reactive Systems, Other Planning Techniques.

- **NATURAL LANGUAGE PROCESSING** **(04 Hours)**

  Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking.

- **EXPERT SYSTEMS** **(05 Hours)**

  Expert Systems – Architecture of Expert Systems, Roles of Expert Systems – Knowledge Acquisition – Meta Knowledge, Heuristics, Typical Expert Systems – MYCIN, DART, XOON, Expert Systems Shells.

- **Practicals will be based on the coverage of the above topics using prolog.** **(28 Hours)**

(Total Contact Time: 42 Hours + 28 Hours = 70 Hours)

---

3. Practicals:

1 Practical assignment to understanding basic concepts of prolog.

2 Practical assignment to implement various search strategies.

3 Practical assignment to implement various algorithm based on game theory.

4 Implementation of heuristic based search techniques.

5 Implementation of neural network based application.

6 Implementation of fuzzy logic based application.

7 Implementation of fuzzy inference engine for an application.

8 Implementation of neuro-fuzzy based system.

4. Books Recommended:

1. Elaine Rich and Kevin Knight, “Artificial Intelligence”, 2nd Edition, Tata McGraw-Hill, 2003.

2. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.
3. Nils Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, 1998,
4. W. Patterson, 'Introduction to Artificial Intelligence and Expert Systems', Prentice Hall of India, 2010.
5. I. Bratko, "Prolog Programming for Artificial Intelligence", 3/E, Addison-Wesley, 2001,
   0-201-40375-7.

B.Tech. III (CSE) Semester – VI
CRYPTOGRAPHY (INSTITUE ELECTIVE-2)
CS362

	L	T	P	Credit
Scheme	**3**	**0**	**0**	**03**

1. <u>Course Outcomes (COs):</u> **At the end of the course, students will be able to**	
CO1	understands the key concept and mathematical background of cryptography.
CO2	apply the concept of security mechanisms from the application developer's perspective.
CO3	analyse security mechanisms while trying to satisfy the required security services.
CO4	evaluate different information hiding and authentication techniques.
CO5	design and develop the security solution depending on the organisation's requirements.

2. <u>Syllabus</u>

- **CLASSICAL CRYPTOGRAPHY** **(04 Hours)**

  Shift Cipher, Substitution Cipher, Affine Cipher, Vigenere Cipher, Hill Cipher, Permutation Cipher, Stream Ciphers, Cryptanalysis of Classical Ciphers.

- **SHANNON'S THEORY** **(04 Hours)**

  Elementary Probability Theory, Computational, Provable and Perfect Secrecy, Entropy, Huffman Encodings, Properties of Entropy, Spurious Key and Unicity Distance, Product Cryptosystems.

- **NUMBER THEORY** **(04 Hours)**

  Modular Arithmetic, Algebraic Structures-Group, Ring, Fields, Galois Fields GF(P), GF(2^n), Euclidean Algorithm, Polynomials and its Operations, Chinese Remainder Theorem, Euler's Phi Function, Fermat's Theorem.

- **MODERN BLOCK CIPHERS** **(04 Hours)**

  The Substitution-Permutation Networks, Linear Cryptanalysis-Piling-up Lemma, Linear Approximation of S-Boxes, Linear Attack on SPN, Differential Cryptanalysis, The Data Encryption Standard (DES), The Advanced Encryption Standard (AES), Block Cipher Modes of Operation.

- **CRYPTOGRAPHIC HASH FUNCTIONS** **(04 Hours)**

  Hash Functions and Data Integrity, Security of Hash Functions-The Random Oracle Model, Iterated Hash Functions- Merkel Damgard Construction, Secure Hash Algorithm (SHA), Message Authentication Codes (MAC), HMAC, CBC-MAC and Authenticated Encryption, Unconditionally Secure MACs.

- **PUBLIC KEY CRYPTOGRAPHY AND DISCRETE LOGARITHMS** **(08 Hours)**

  The El-Gamal Cryptosystem and its Security, Algorithms for Discrete Logarithm Problem-Shank, Pollard-rho, Pohlig- Hellman, Index Calculus, Finite Fields, The Diffie-Hellman Problems, Elliptic Curves-Elliptic Curves over Real numbers and Finite Fields, Properties, Point Compression and ECIES, Point Addition, Scalar Multiplication.

- **RSA CRYPTOSYSTEM AND FACTORING INTEGERS** **(06 Hours)**

  RSA Key Generation, Encryption, Decryption, The Integer Factorization Problem, Primality Testing-Legendre and Jacobi Symbols, The Solovay-Strassen Algorithm, The Miller-Rabin Algorithm, Square root modulo a composite, Factoring Algorithm, Attacks on RSA-Computing ϕ(n), Low Decryption Exponent Attack, Optimal Asymmetric Encryption Padding.

- **DIGITAL SIGNATURE SCHEMES** **(04 Hours)**

  Security Requirements, Signature and Hash Functions, ElGamal Digital Signature Scheme and its Security, Variants of ElGamal Digital Signature-Schnorr, Digital Signature Algorithm(DSA), Elliptic Curve DSA, Provably Secure Signature Schemes, One Time Signatures, Full Domain Hash, Undeniable Signatures, Blind Signatures, Fail-Stop Signatures.

- **IDENTIFICATION SCHEMES AND ENTITY AUTHENTICATION** **(02 Hours)**

  Challenge Response Protocols, Password Based Authentication, Zero Knowledge Schemes.

- **ADVANCED TOPICS** **(02 Hours)**

(Total Contact Time: 42 Hours)

---

3. <u>Books Recommended:</u>

1. Stinson, Douglas R., “Cryptography: theory and practice”, 3rd Edition, Chapman and Hall/CRC, 2005.
2. Stallings, William, “Cryptography and network security: principles and practice”, 7th Edition, Upper Saddle River: Pearson, 2017.
3. Forouzan, Behrouz A., “Cryptography & network security”, 3rd Edition, McGraw-Hill, Inc., 2007.
4. Schneier, Bruce, “Applied cryptography: protocols, algorithms, and source code in C”, 2nd Edition, john wiley & sons, 2007.
5. Patel, Dhiren R. “Information security: theory and practice”, 1st Edition, PHI Learning Pvt. Ltd., 2008.

B.Tech. III (CSE) Semester – VI
NATURAL LANGUAGE PROCESSING (CORE ELECTIVE-2)
CS324

	L	T	P	Credit
Scheme	**3**	**0**	**0**	**03**

1. Course Outcomes (COs): **At the end of the course, students will be able to**	
CO1	understand basics principles of natural language processing.
CO2	apply machine learning techniques for NLP based different tasks.
CO3	perform statically analysis and classification, recognition using NLP knowledge acquired.
CO4	evaluate the performance of machine translation solutions through statistical parameters.
CO5	design efficient solution for parser, translator and different applications based on NLP for day to day usage.

2. Syllabus

- **INTRODUCTION** **(04 Hours)**

  Human Languages, Language Models, Computational Linguistics , Ambiguity and Uncertainty in Language, Processing Paradigms, Phases in Natural Language Processing, Basic Terminology, Overview of Different Applications, Regular Expressions and Automata, Finite State Transducers and Morphology, Automata, Word Recognition, Lexicon, Morphology, Acquisition Models, Linguistics Resources, Introduction to Corpus, Elements in Balanced Corpus.

- **SYNTAX AND SEMANTICS** **(08 Hours)**

  Natural Language Grammars, Lexeme, Phonemes, Phrases and Idioms, Word Order, Tense, Probabilistic Models of Spelling, N-grams, Word Classes and Part of Speech Tagging using Maximum Entropy Models, Transformation Based Tagging (TBL), Context Free Grammars for English, Features and Unification, Lexicalized and Parsing, Treebanks, Language and Complexity, Representing Meaning, Semantic Analysis, Lexical Semantics, Word Sense Disambiguation.

- **PROBBILISTIC LANUAGE MODELING** **(08 Hours)**

  Statistical Inference, Hidden Markov Models, Probabilistic (weighted) Finite State Automata, Estimating the Probability of a Word, and Smoothing, Probabilistic Parsing, Generative Models of Language, Probabilistic Context Free Grammars, Probabilistic Parsing, Statistical Alignment and Machine Translation, Clustering, Text Categorization, Viterbi Algorithm for Finding Most Likely HMM Path.

- **PRAGMATICS** **(06 Hours)**

  Discourse, Dialogue and Conversational Agents, Natural Language Generation, Machine Translation, Dictionary Based Approaches, Reference Resolution, Algorithm for Pronoun Resolution, Text Coherence, Discourse Structure, Applications of NLP- Spell-Checking.

- **MACHINE TRANSLATION** **(08 Hours)**

  Probabilistic Models for Translating One to Another Language, Alignment, Translation, Language Generation, Expectation Maximization, Automatically Discovering Verb Subcategorization, Language Modelling Integrated into Social Network Analysis, Automatic Summarization, Question-Answering, Interactive Dialogue Systems.

- **ADVANCED TOPICS** **(08 Hours)**

  Summarization, Information Retrieval, Vector Space Model, Term Weighting, Homonymy, Polysemy, Synonymy, Improving User Queries, Document Classification, Sentence Segmentation, and Other Language Tasks, Automatically-Trained Email Spam Filter, Automatically Determining the Language, Speech Recognition.

(Total Contact Time: 42 Hours)

---

3. Books Recommended:

1. Daniel Jurafsky, James H. Martin: "Speech and Language Processing", 2/E, Pearson Education, 2009.
2. James Allen, "Natural Language Understanding", 2/E, Addison-Wesley, 1994.
3. Christopher D. Manning, Hinrich Schutze: "Foundations of Statistical Natural Language Processing", 1/E, MIT Press, 1999.
4. Steven Bird, “Natural Language Processing with Python”, 1st Edition, O'Reilly, 2009.
5. Jacob Perkins, “Python Text Processing with NLTK 2.0 Cookbook”, 2nd Edition, Packt Publishing, 2010.
6. Bharati A., Sangal R., Chaitanya V., “Natural language processing: A Paninian perspective”, PHI, 2000.
7. Siddiqui T., Tiwary U. S., “Natural language processing and Information retrieval”, 1st Edition, OUP,2008.