B.Tech. *Computer Science & Engineering (Artificial Intelligence)* **SEMESTER V**

Course No.	Type	Course	L	T	P	Credits	Evaluation Scheme (Percentage weights)					Offering Dept.	AICTE COURSE TYPE	Pre-requisite	
							Theory			Practical					
							CA	MS	ES	CA	ES			Code	Title
CACSC14	CC	Principles of Compiler Construction	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	CACSC10	Theory of Automata & Formal languages
CACSC15	CC	Distributed Computing	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	CAECC12	Data Communication
CACSC16	CC	Game Theory and Applications	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	CACSC06	Design and Analysis of Algorithms
CACSC17	CC	Machine Learning	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	CAMTC13	Probability and Stochastic Processes
CACSExx	ED					4									
CACSExx	ED					4									
	EO	Elective Open	-	-	-	4	-	-	-	-	-	-	MANDATORY COURSE		
			28 2*			**28**									

2*: The actual weekly load depends upon the elective chosen by the student under FE. Maximum 28 credits.

B.Tech. *Computer Science & Engineering (Artificial Intelligence)*
SEMESTER V (Discipline Centric Electives: EDs)

Course No.	Course Name	Prerequisite
CACSE01	Semantic Web 3 1 0	CACSC05
CACSE02	Object oriented analysis and design 3 0 2	CACSC02
CACSE03	Cryptography techniques 3 1 0	CACSC01

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CACSC14	CC	Principles of Compiler Construction	3	0	2	4	15	15	40	15	15	Theory Automata and Formal Languages

COURSE OUTCOMES

1 Understand the internal organization and behavior of the compilers and other language processors.
2 Apply the formal constructs for designing a compiler.
3 Study and understand the functioning of a compiler.
4 Gain an ability to design simple domain-specific languages (DSLs) using compiler construction tools.
5 Go for the translation of languages or design the tools for online processing.

COURSE CONTENTS

Unit 1

Introduction: Language processors, structure of a compiler, compiler-construction tools, evolution of programming languages, applications of compiler technology, Transition diagrams, bootstrapping, just-in-time compilation.

Unit 2

Lexical analysis: Input buffering, specification and recognition of tokens, lexical analyzer generator.

Unit 3

Syntax analysis: Specification of syntax using grammar. Top-down parsing – recursive-descent, predictive. Bottom-up parsing – shift-reduce, SLR, CLR, LALR. Parser generator.

Unit 4

Intermediate-code generation: Syntax-directed translation. Three-address code. Translation of declarations, expressions, control flow. Backpatching. Runtime environment: Activation trees and records.

Unit 5

Code optimization: Sources of optimization, basic blocks, optimization of basic blocks, data-flow analysis, loop optimizations. Code generation: Issues, register allocation and assignment, peephole optimization

Practical:

1. Develop simple language processors like desk calculator and assembler.
2. Design a small high-level language.
3. Develop a lexical analyzer and a syntax analyzer for the same using the LEX and YACC tools. Also implement the bookkeeper module.
4. Design a small high-level language and implement a compiler for the same. If the target machine of the compiler is a hypothetical machine, then implement a simulator for it.
5. Develop a simple calculator using LEX and YACC tools.

6. Implement a program for symbol table using hashing
7. Implement a two-pass assembler
8. Implement a bottom-up parser using YACC tool.
9. Represent 'C' language using Context Free Grammar
10. Add assignment statement, If then else statement and while loop to the calculator and generate the three address code for the same.

SUGGESTED READINGS

1. Aho, A. V., Lam, M. S., Sethi, R. and Ullman J. D., "Compilers – Principles, Techniques and Tools (2nd ed.)", Pearson.
2. Chattopadhyay, S. 2005, "Compiler Design, PHI".
3. Appel, A. W. 200, "Modern Compiler Implementation in C", Cambridge University Press.
4. Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.

Course No.	Title of the Course	Course Structure	Pre-Requisite
CACSC15	**Distributed Computing**	**3L-0T-2P**	**Computer Networks**

COURSE OUTCOMES (CO)

1. Study software components of distributed computing systems. Know about the communication and interconnection architecture of multiple computer systems.
2. Recognize the inherent difficulties that arise due to distributedness of computing resources.
3. Understand basic problems in distributed computing, especially in relation to concurrency, parallelism, synchronization, deadlocks, safety and liveness properties.
4. Understand differences between various distributed computing models and widely used distributed computing schemes
5. Understanding communication mechanism among the distributed entities

COURSE CONTENT:

UNIT-I

CHARACTERIZATION OF DISTRIBUTED SYSTEMS: Introduction, Examples of distributed systems, Trends in distributed systems

SYSTEM MODELS: Physical models, Architectural models, Fundamental models.

UNIT-II

INTERPROCESS COMMUNICATION : The API for the Internet protocols, External data representation and marshalling, Multicast communication, Network virtualization: Overlay networks.

REMOTE INVOCATION: Request-reply protocols, Remote procedure call, Remote method invocation.

INDIRECT COMMUNICATION: Group communication, Publish-subscribe systems, Message queues, Shared memory approaches

UNIT-III

DISTRIBUTED FILE SYSTEMS: File service architecture

TIME AND GLOBAL STATES: Clocks, events and process states, Synchronizing physical clocks, Logical time and logical clocks, Global states, Distributed debugging

UNIT-IV

COORDINATION AND AGREEMENT: Distributed mutual exclusion Elections, Coordination and agreement in group communication, Consensus and related problems

TRANSACTIONS AND CONCURRENCY CONTROL: Transactions, Nested transactions, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control.

UNIT-V

DISTRIBUTED TRANSACTIONS: Flat and nested distributed transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery.

REPLICATION: System model and the role of group communication, Fault-tolerant services, Transactions with replicated data.

SECURITY: Overview of security techniques, Cryptographic algorithms, Digital signatures, Cryptography pragmatics.

Recommended Books:

1. G. Coulouris, J. Dollimore, "Distributed Systems Concepts and Design," Addison Wesley.
2. Hwang & Dongarra & Fox, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things,"
3. M. Singhal, N.G. Shivarathri , "Advanced Operating Systems,", McGraw Hill.
4. Randy Chow, T. Johnson, "Distributed Operating Systems and Algorithms," Addison Wesley.
5. A.S. Tanenbaum, "Distributed Operating Systems," Prentice Hall.
6. M. Tamer Ozsu, PatrickValduriez, "Principles of Distributed Database Systems," Prentice Hall International

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CACSC16	CC	Game Theory and Applications	3	0	2	4	15	15	40	15	15	algortihms

COURSE OUTCOMES: The course outcomes of game theory and its applications are defined as follows:

1. The aim of this course is to introduce students to the novel concepts of Game.
2. Introduce the theory with special emphasis on its applications in diverse fields and current research.
3. To get familiar with perfect and imperfect information
4. To get familiar with the extensive and strategic games
5. To study various applications on auction and designing mechanisms

COURSE CONTENTS

Unit I: Introduction:

Introduction, overview, uses of game theory, some applications and examples, and formal definitions of: the normal form, payoffs, strategies, pure strategy Nash equilibrium, dominant strategies.

Unit II: Games with Perfect Information:

Normal form games, Dominant strategy equilibrium, Iterated strict dominance, Nash Equilibrium in pure strategies, Cournot duopoly, Prisoner's dilemma, Battle of Sexes, Problem of commons, Final offer arbitration, Strategic Voting, Non-existence of pure strategy Nash equilibrium pareto optimality

Unit III:

Extensive Form Game with Perfect Information: Theory, Stackelberg Model of Duopoly, Buying Votes, Committee Decision-Making, Repeated games: The Prisoner's Dilemma, General Result

Unit IV:

Strategic Games with Imperfect Information: Bayesian Games, Cournot's Duopoly with Imperfect Information, Radio Spectrum, With Arbitrary Distribution of Valuations

Unit V:

Auction and Mechanism Design with Applications: Revenue Equivalence, Risk Averse Bidders, Asymmetries among Bidders, Mechanism, Optimal Mechanism.

SUGGESTED READINGS

1. Martin Osborne, An Introduction to Game Theory, Oxford University Press, 2003
2. Vijay Krishna, Auction Theory, Academic Press.
3. Prajit Dutta, Strategies and Games, MIT Press
4. http://www.ece.stevens-tech.edu/~ccomanic/ee800c.html
5. Allan MacKenzie, Game Theory for Wireless Engineers, Synthesis lectures on Communications, 2006

Course	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-

No.												requisites
CACSC17	CC	Machine Learning	3	0	2	4	15	15	40	15	15	Algorithms

COURSE OUTCOMES

1. To develop an understanding of the fundamentals of machine learning.
2. To develop an understanding of statistical pattern recognition.
3. To gain an insight into the various components of machine learning such as supervised learning, unsupervised learning, learning theory, reinforcement learning and adaptive control.
4. To acquire skills that can be applied to various components of machine learning to applications like robotic control, data mining, autonomous navigation, bioinformatics, speech recognition, and text and web data processing.
5. To apply the knowledge gained to the projects

COURSE CONTENTS

Unit I

Introduction: Definition of learning systems. Goals and applications of machine learning.
Inductive Classification: The concept learning task. Concept learning as search through a hypothesis space. General-to-specific ordering of hypotheses. Finding maximally specific hypotheses. Version spaces and the candidate elimination algorithm. Learning conjunctive concepts.

Unit II

Ensemble Learning: Using committees of multiple hypotheses. Bagging, boosting, and DECORATE. Active learning with ensembles.
Experimental Evaluation of Learning Algorithms: Measuring the accuracy of learned hypotheses. Comparing learning algorithms: cross-validation, learning curves, and statistical hypothesis testing.

Unit III

Computational Learning Theory: Models of learnability: learning in the limit; probably approximately correct (PAC) learning. Sample complexity: quantifying the number of examples needed to PAC learn. Computational complexity of training.
Sample complexity for finite hypothesis spaces. PAC results for learning conjunctions, kDNF, and kCNF. Sample complexity for infinite hypothesis spaces, Vapnik- Chervonenkis dimension.

Unit IV

Rule Learning: Propositional and First-Order, Translating decision trees into rules. Heuristic rule induction using separate and conquer and information gain. First-order Horn-clause induction (Inductive Logic Programming) and Foil. Learning recursive rules. Inverse resolution, Golem, and Progol.
Artificial Neural Networks: Neurons and biological motivation. Linear threshold units. Perceptrons: representational limitation and gradient descent training.
Multilayer networks and backpropagation. Hidden layers and constructing intermediate, distributed representations. Overfitting, learning network structure, recurrent networks.

Unit V

Support Vector Machines: Maximum margin linear separators. Quadractic programming solution to finding maximum margin separators. Kernels for learning non-linear functions.
Bayesian Learning: Probability theory and Bayes rule. Naive Bayes learning algorithm. Parameter smoothing. Generative vs. discriminative training. Logisitic regression. Bayes nets and Markov nets for representing dependencies.

Guidelines for project based work: Semester long projects, presentations, research work, term papers

based on the above topics.

SUGGESTED READINGS

1. Richard Duda, Peter Hart and David Stork, Pattern Classification, 2nd ed. John Wiley & Sons, 2001.
2. Tom Mitchell, Machine Learning. McGraw-Hill, 1997.
3. Richard Sutton and Andrew Barto, Reinforcement Learning: An introduction. MIT Press, 1998
4. Trevor Hastie, Robert Tibshirani and Jerome Friedman, The Elements of Statistical Learning. Springer, 2009

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CACSE01	**ED**	**Semantic Web**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	**Algorithms**

COURSE OUTCOMES

1. Understand the rationale behind Semantic Web.
2. Model ontologies using Resource Description Framework (RDF).
3. Design RDF Schemas for ontologies, model and design ontologies using Web Ontology Language (OWL).
4. Query ontologies using SPARQL.
5. Understand and reflect on the principles of Ontology Engineering, make an association between Semantic web and Web 2.0. And apply Semantic web technologies to real world applications.

COURSE CONTENTS

Unit I

Introduction to the Semantic Web, ontologies and description logic

Unit II

Overview and Introduction: Knowledge Representation, Semantic Web in Depth: RDF and RDF Schema, Semantic Web in Depth: OWL. Resource description framework, lightweight ontologies, a query language for Resource description framework (RDF) - SPARQL

Unit III

Writing OWL ontologies: Protégé, Semantic Web Methodologies and Design Patterns, Semantic Web in Depth: SPARQL, Semantic Web in Depth: Rules.

Unit IV

Publishing on the Semantic Web: Linked Data, Semantic Web Vocabularies and Applications, Semantic Web vs Web2.0, Trust and Community.

Unit V

Applications: Information Integration, Ontology Alignment, Scalable Reasoning and Knowledge Acquisition.

SUGGESTED READINGS

1. A Semantic Web Primer, third edition, MIT Press, 2012, Grigoris Antoniou, Paul Groth, Frank van Harmelen and Rinke Hoekstra
2. Grigoris Antoniou, Frank Van Harmelen, A Semantic Web Primer, MIT Press, 2008.
3. Pascal Hitzler, Markus Krotzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies, CRC Press, 2009.

4. Dean Allemang, James Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, Morgan Kauffmann, ISBN-10: 0-12-373556-4.
5. Geroimenko, Vladimir; Chen, Chaomei (Eds.) 2nd ed., 2006, XIV, 248 p. 108 illus., Hardcover ISBN: 978- 1-85233-976-0, Visualizing the Semantic Web XML-based Internet and Information Visualization, SpringerVerlag London Ltd; 2Rev Ed edition (Oct 2005).
6. Michael C. Daconta, Leo J. Obrst, Kevin T. Smith, The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management: A Guide to the Future of XML, Web Services and Knowledge Management, John Wiley & Sons (20 Jun 2003).
7. S Powers, Practical RDF (Paperback) , OReilly (1 Aug 2003).
8. Thomas B. Passin, Explorer's Guide to the Semantic Web (Paperback), Manning Publications (8 Jul 2004)

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CACSE02	**ED**	**Object Oriented Analysis and Design**	**3**	**0**	**2**	**4**	**15**	**15**	**40**	**15**	**15**	DBMS, data structures

COURSE OUTCOMES

1. To appreciate the fact that software development cannot be done in an adhoc fashion and has to follow a disciplined systematic approach for timely development of software within budget using suitable Process model and techniques
2. To learn various techniques for Requirements Elicitation and Specification in order to develop SRS for a problem domain
3. To model a problem domain using Object oriented analysis and design using UML
4. To learn Different techniques for software project management like Feasibility Analysis, Cost and Effort Estimation, Scheduling a project
5. To learn about different Software Quality frameworks, OO metrics, Configuration Management etc.

COURSE CONTENTS

Unit 1

Introduction: Introduction to software engineering, Importance of software, The Software evolution, Software characteristics, Software components, Software applications, Crisis-Problem and causes. Difference between software engineering and system engineering

Software Process Models: Waterfall model, Evolutionary Models, prototyping, V Model, Spiral model Incremental Model, RAD Model etc. Introduction to Agile models like Scrum, Extreme Programming, Feature Driven Development, Crystal etc., Comparison between Traditional and Agile models

Unit 2

Requirement Engineering: Different Types of Requirements: Functional, Non Functional and Domain Requirements in detail, Requirem**ent** elicitation Techniques like interviews, questionnaire, brainstorming, JAD, Scenario, Mind mapping, Requirement workshop, Prototyping, CRC Cards etc. Requirements Management, Writing SRS as per IEEE standard, Quality characteristics of SRS

Unit 3

Requirements Specification: Difference between structured and Object Oriented Analysis, Different views of modeling, Quick review of ER diagram, Data flow diagrams, State Transition Diagrams, data Dictionary,

Introduction to Unified Modeling language Conceptual Model of Unified Modeling Language (UML), Use case Diagram, Activity Diagram, Class Diagram, Object Diagram,

Unit 4

Sequence Diagram, Communication Diagram, State Machine Diagram, Timing Diagram, Composite Structure diagram, Package Diagram, Component Diagram, Deployment Diagram, Introduction to Rational Unified Process(RUP)

Unit 5

Software project Management: Project Management Process, System Request, Feasibility Analysis in detail Project scheduling, Finding Critical Path, Effort Estimation for OO systems using Use case diagram, Introduction to Object Oriented Metrics, Configuration Management,
Software Quality : Software Quality Models like McCall's Quality model, Quality frameworks like Capability Maturity Model, ISO9001

Practical:

1. Choose a problem domain and textually write the Functional and Non Functional requirements of the domain. While writing make use of elicitation techniques discussed in the class
2. For a Problem domain develop a Mind-Map
3. Draw the static View and functional view of the Problem domain of exercise 1 using ER diagram and DFD
4. Draw the Dynamic view of working of an ATM Machine or Microwave Oven
5. Develop a Use Case Model(use Case diagram and Use case Narratives) for a problem domain
6. For the problem of Q5, draw activity diagram, sequence diagrams and Class diagram
7. Draw the state machine diagram for a CD player
8. Perform Effort estimation activity and Implement Critical path method using an open source Tool for a case study
9. Write SRS as per IEEE std-830 for the problem domain of Q5

SUGGESTED READINGS

1. Sommerville, "Software Engineering", 10th Edition Published by Pearson
2. Sangeeta Sabharwal, "Software Engineering: Principles and Techniques", Second Edition, Published by New Age International Publishers, 2020
3. R . S. Pressman, "Software Engineering – A practitioner's approach", McGraw Hill Int. Ed.
4. Object Oriented Modeling and Design with UML by Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson
5. UML distilled, Third Edition by Martin Fowler, Addison Wesley .
6. The Unified Software Development Process by I*var* Jacobson, Booch and Rumbaugh,Addison Wesley, 2007

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CACSE03	**ED**	**Cryptography techniques**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	Networking
COURSE OUTCOMES												

1. Explain common attacks against network assets, the associated threats and vulnerabilities, and what network security personnel do to secure assets,
2. Explain how to use cryptography to help protect information and how to choose an appropriate encryption method for an organization
3. Help protect information in an organization by using authentication and access control and deploy and manage certificates.
4. Help protect transmission of data by identifying threats to network devices and implementing security for common data transmission, remote access, and wireless network traffic
5. Identify common security threats and vulnerabilities to directory services and DNS, and then apply security methods to help protect them

COURSE CONTENTS

Unit I

Foundation of Security & Cryptography: OSI security architecture, Security attack, security services and mechanisms, model of security, Classical encryption techniques: Substitution Techniques, Transposition Techniques and Steganography.

Unit II

Block Ciphers and Public key cryptography: Design Principle of Block Ciphers: DES, AES, Multiple Encryption, Block Cipher modes of operation, stream ciphers, RC4, Public Key Cryptography: RSA, Key management, Diffie-Hellman Key exchange, Elliptic Curve Cryptography

Unit III

Hashes & Digital Signatures: Authentication functions, Message authentication codes, Hash functions and their security, HMAC, CMAC, Secure hash algorithms, Digital Signature: Certificates & standards, authentication protocols

Unit IV

Authentication Applications: Kerberos, X.509 Authentication service, public key infrastructure, electronic Mail Security: pretty good privacy, S/MIME

Unit V

IP and Web Security Protocols: IPsec, Secure socket layer and transport layer security, secure e-transaction, System Security: Computer Virus, Firewall & Intrusion Detection, Trusted systems

SUGGESTED READINGS

1. Cryptography & Network Security by Stallings, William (Fourth Edition or later)
2. Foundations of Cryptography (Basic Tools), Oded Goldreich Cambridge 2001.
3. Cryptography: Theory and Practice, by Douglas R. Stinson, First Edition, second edition: first volume
4. An Introduction to Cryptology, Henk C.A. van Tilborg, Kluwer Academic Publishers, 1987