ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES

(An Autonomous Institution)

III B. Tech I Semester (CSE)

S.No	Category	Course Code	Title	L	T	P	Credits
1	PC	23A0551T	Introduction to Artificial Intelligence	3	0	0	3
2	PC	23A0552T	Introduction to Computer Networks	3	0	0	3
3	PC	23A0553T	Automata Theory and Compiler Design	3	0	0	3
4	PC	23A0554T	Introduction to Quantum Technologies and Applications	3	0	0	3
5	PE	23A055AT 23A055BT 23A055CT 23A055DT	**Professional Elective-I** 1. Object Oriented Analysis and Design 2. Soft Computing 3. Data Warehousing & Data Mining 4. Microprocessors and Microcontrollers	3	0	0	3
6			**Open Elective-I**	3	0	0	3
7	PC	23A0551L	Artificial Intelligence Lab	0	0	3	1.5
8	PC	23A0552L	Computer Networks Lab	0	0	3	1.5
9	SEC	23A0555L	Full Stack Development - II	0	1	2	2
10	SEC	23A0556L	Tinkering Lab for Computer Engineers	0	0	2	1
11	INTERN	23A0557I	Evaluation of Community Service Internship	-	-	-	2
			Total	**18**	**1**	**10**	**26**

Open Elective – I

S.No.	Category	Course Code	Course Name	Offered by the Dept.
1	OE	23A015DT	Green Buildings	CIVIL
2		23A015ET	Construction Technology and Management	
3		23A025ET	Electrical Safety Practices and Standards	EEE
4		23A035FT	Sustainable Energy Technologies	ME
5		23A045DT	Electronic Circuits	ECE
6		23A045ET	Communication Systems	
7		23A055GT	Quantum Technologies and Applications	CSE & Allied
8		23AHS51T	Mathematics for Machine Learning and AI	Mathematics
9		23AHS52T	Materials Characterization Techniques	Physics
10		23AHS53T	Chemistry of Energy Systems	Chemistry
11		23AHS54T	English for Competitive Examinations	Humanities
12		23AHS56T	Entrepreneurship and New Venture Creation	

Note:

1. A student is permitted to register for Honours or a Minor in IV semester after the results of III Semester are declared and students may be allowed to take maximum two subjects per semester pertaining to their Minor from V Semester onwards.

2. A student shall not be permitted to take courses as Open Electives/Minor/Honours with content substantially equivalent to the courses pursued in the student's primary major.

3. A student is permitted to select a Minor program only if the institution is already offering a Major degree program in that discipline.

III B. Tech II Semester (CSE)

S.No	Category	Course Code	Title	L	T	P	Credits
1	PC	23A0561T	Introduction to Machine Learning	3	0	0	3
2	PC	23A0562T	Cloud Computing	3	0	0	3
3	PC	23A0563T	Cryptography & Network Security	3	0	0	3
4	PE	23A056AT 23A056BT 23A056CT 23A056DT	**Professional Elective-II** 1. Software Testing Methodologies 2. Cyber Security 3. DevOps 4. Embedded Systems Design	3	0	0	3
5	PE	23A056ET 23A056FT 23A056GT 23A056HT	**Professional Elective-III** 1. Software Project Management 2. Mobile Adhoc Networks 3. Natural Language Processing 4. Distributed Operating System	3	0	0	3
6			**Open Elective – II**	3	0	0	3
7	PC	23A0561L	Machine Learning Lab	0	0	3	1.5
8	PC	23A0563L	Cryptography & Network Security Lab	0	0	3	1.5
9	SEC	23AHS65L	**Skill Enhancement course** Soft skills OR IELTS	0	1	2	2
10	AUD	23AHS67T	**Audit Course** Technical Paper Writing & IPR	2	0	0	-
11	PC	23A0564L	**Workshop**	0	0	0	0
			Total	**20**	**1**	**8**	**24**
Mandatory Industry Internship of 08 weeks duration during summer vacation							

Open Elective – II

S.No.	Category	Course Code	Course Name	Offered by the Dept.
1	OE	23A016GT	Disaster Management	CIVIL
2		23A016HT	Sustainability in Engineering Practices	
3		23A026IT	Renewable Energy Sources	EEE
4		23A036KT	Automation and Robotics	ME
5		23A046GT	Digital Electronics	ECE
6		23AHS61T	Optimization Techniques for Engineers	Mathematics
7		23AHS66T	Mathematical Foundation Of Quantum Technologies	
8		23AHS62T	Physics of Electronic Materials And Devices	Physics
9		23AHS63T	Chemistry of Polymers And Applications	Chemistry
10		23AHS64T	Academic Writing and Public Speaking	Humanities

Title of the Course:	Introduction to Artificial Intelligence
Category:	Professional Core
Couse Code:	23A0551T
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: This course is designed to:

1. Introduce Artificial Intelligence.
2. Teach about the machine learning environment
3. Present the searching Technique for Problem Solving
4. Introduce Natural Language Processing and Robotics.

Course Outcomes:

After completion of the course, students will be able to

1. Apply searching techniques for solving a problem.
2. Design Intelligent Agents.
3. Develop Natural Language Interface for Machines.
4. Design mini robots.
5. Summarize past, present and future of Artificial Intelligence. .

Unit I Introduction 09

Introduction: What is AI, Foundations of AI, History of AI, The State of Art
Intelligent Agents: Agents and Environments, Good Behaviour: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

09

Unit II Solving Problems by searching

Problem Solving Agents, Example problems, Searching for Solutions, Uninformed Search Strategies, Informed search strategies, Heuristic Functions, Beyond Classical Search: Local Search Algorithms and Optimization Problems, Local Search in Continues Spaces, Searching with Nondeterministic Actions, Searching with partial observations, online search agents and unknown environments.

Unit III Knowledge Representation and Planning 08

Knowledge-Based Agents and the Wumpus World, Propositional Logic: Syntax, Semantics, and Theorem Proving, Agents based on Propositional Logic and Model Checking, First-Order Logic: Syntax, Semantics, and Inference Mechanisms, Unification, Forward and Backward Chaining, and Resolution, Knowledge Engineering and Ontological Representation, Introduction to Classical Planning and Planning Algorithm

Unit IV Reinforcement Learning & Natural Language Processing 08

Reinforcement Learning: Introduction, Passive Reinforcement Learning, Active Reinforcement Learning, Generalization in Reinforcement Learning, Policy Search, applications of RL Natural Language Processing: Language Models, Text Classification, Information Retrieval, Information Extraction.

Unit V Robotics and Computer Vision 10

Robotics: Introduction, Robot Hardware, Robotic Perception, planning to move, planning uncertain movements, Moving, Robotic software architectures, application domains. **Perception:** Computer Vision, Introduction, Image Formation, Simple Image Features, Classifying Images, Detecting Objects, The 3D World, Using Computer Vision. **Philosophical foundations:** The Limits of AI, Can Machines Really Think? The Ethics of AI.

Prescribed Textbooks:

1. Stuart J.Russell, Peter Norvig, ―Artificial Intelligence A Modern Approach‖, 3rd Edition Pearson Education, 2019.

Reference Books:

1. Nilsson, Nils J., and Nils Johan Nilsson. Artificial intelligence: a new synthesis. Morgai Kaufmann, 1998.
2. Johnson, Benny G., Fred Phillips, and Linda G. Chase. "An intelligent tutoring system fo the accounting cycle: Enhancing textbook homework with artificial intelligence." Journa of Accounting Education 27.1 (2009): 3039.

Online Learning Resources:

http://peterindia.net/AILinks.html
http://nptel.ac.in/courses/106106139/
https://nptel.ac.in/courses/106/105/106105152/

Title of the Course	:	**Introduction to Computer Networks**
Category	:	Professional Core
Couse Code	:	23A0552T
Year	:	III B. Tech
Semester	:	I Semester
Branch	:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: This course is designed to

1. Understand the basic concepts of Computer Networks.
2. Introduce the layered approach for design of computer networks.
3. Expose the network protocols used in Internet environment.
4. Explain the format of headers of IP, TCP and UDP Familiarize with the applications of Internet
5. Familiarize with the applications of Internet
6. Elucidate the design issues for a computer network

Course Outcomes:

After completion of the course, students will be able to

1. Identify the software and hardware components of a computer network.
2. Design software for a computer network.
3. Develop new routing, and congestion control algorithms
4. Assess critically the existing routing protocols.
5. Explain the functionality of each layer of a computer network
6. Choose the appropriate transport protocol based on the application requirements

Unit 1 Computer Networks and the Internet 08

What Is the Internet The Network Edge, The Network Core, Delay, Loss, and Throughput in Packet Switched Networks(Textbook 2), Reference Models, Example Networks, Guided Transmission Media, Wireless Transmission (Textbook 1)

Unit 2 The Data Link Layer, Access Networks, and LANs 10

Data Link Layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols (Textbook 1)

Introduction to the Link Layer, Error-Detection and -Correction Techniques, Multiple Access Links and Protocols, Switched Local Area Networks Link Virtualization: A Network as a Link Layer, Data Center Networking, Retrospective: A Day in the Life of a Web Page Request (Textbook 2)

Unit 3 The Network Layer 08

Unit 4 The Transport Layer 09

Connectionless Transport: UDP (Textbook 2), The Internet Transport Protocols: TCP, Congestion Control (Textbook 1)

Unit 5 The Application Layer 08

Principles of Network Applications, The Web and HTTP, Electronic Mail in the Internet, DNS-The Internet‘s Directory Service, Peer-to-Peer Applications Video Streaming and Content Distribution Networks (Textbook 2)

Textbooks:

1. Andrew S.Tanenbaum, David j.wetherall, Computer Networks, 6th Edition, PEARSON.
2. James F. Kurose, Keith W. Ross, ―Computer Networking: A Top-Down Approach‖, 6th edition, Pearson, 2019.

Reference Books:

1. Forouzan, Datacommunications and Networking, 5th Edition, McGraw Hill Publication.
2. Youlu Zheng, Shakil Akthar, ―Networks for Computer Scientists and Engineers‖, Oxford Publishers, 2016.

Online Learning Resources:

1. https://nptel.ac.in/courses/106105183/25
2. http://www.nptelvideos.in/2012/11/computer-networks.html
3. https://nptel.ac.in/courses/106105183/3

Title of the Course:	**Automata Theory and Compiler Design**
Category:	Professional Core
Couse Code:	23A0553T
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. Able to understand the concept of abstract machines, construct FA, Regular Expressions for the regular languages and equivalent FSMs.
2. Able to construct pushdown automata equivalent to Context free Grammars, construct Turing Machines and understand undecidability.
3. Emphasize the concepts learnt in phases of compiler, lexical analyser and Top-down parser.
4. Able to understand the concepts of Bottom-up parser, Intermediate Code Generation.
5. Able to understand the concepts of Code optimizer and Code Generation.

Course Outcomes:

At the end of the course, the student will be able to

1. Demonstrate knowledge on Automata Theory, Regular Expression and Analyze and Design of finite automata, and prove equivalence of various finite automata.
2. Demonstrate knowledge on context free grammar, Analyze and design of PDA and TM.
3. Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.
4. Ability to implement semantic rules into a parser that performs attribution while parsing and apply error detection and correction methods.
5. Apply the code optimization techniques to improve the space and time complexity of programs while programming and Ability to design a compiler.

Unit 1 Introduction to Automata and Regular Expressions 12

Introduction, Alphabets, Strings and Languages, Chomsky Hierarchy, Automata and Grammars, Regular Grammar and Language, Finite Automata, Deterministic finite Automata (DFA), Nondeterministic finite Automata (NFA), Equivalence of NFA and DFA, Minimization of Finite Automata, Regular Expressions, Converting Regular Grammar and Expression into Finite Automata, Pumping lemma for regular sets, Closure properties of regular sets (Without proof).

Unit 2 Context Free Grammars and Pushdown Automata 12

Context Free Language, Context Free Grammar, Derivation and Parse tree, Ambiguity, Simplification of CFG‘s, Chomsky Normal Form, Greibach Normal Form, Push Down Automat (PDA), Design of PDA, Equivalence of PDA and CFL/CFG

Unit 3 Turing Machines and Introduction to Compilers 12

Turing Machine, TM Model, Language acceptance, Design of Turing Machine, Compilers, Phases of Compiler, The role of Lexical Analyzer, Input Buffering.

Unit 4 Parsers and Intermediate Code Generation 12

Parser, Top-Down parsers: Recursive Descent Parsers, Predictive Parsers Bottom–up Parsers: Shift-Reduce Parsing, LR parsers, Intermediate Code Generation: Three address codes.

Unit 5 Code Optimization and Code Generation 12

Code Optimization: Peephole optimization, Basic blocks and flow graphs, DAG, Principles of Source Code Optimization, Code Generation: Issues in Design of Code Generation, Simple Code Generator.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008.
2. Compilers Principles, Techniques and Tools Aho, Ullman, Ravisethi, Pearson Education.

REFERENCES:

1. Introduction to Languages and Theory of Computation, John C Martin, 1/e, 2009, Tata McGraw Hill Education, Hyderabad, India.
2. Introduction to Theory of Computation, Sipser, 2/e, 2005, Thomson, Australia.
3. Compiler Construction: Principles And Practice, Kenneth C. Louden, Thomson/ Delmar Cengage Learning, 2006.
4. Engineering a compiler, Keith Cooper and Linda Torczon, 2 nd Edition, Morgan Kaufmann, 2011.

e-Resources:

1) https://nptel.ac.in/courses/106/104/106104028/
2) https://nptel.ac.in/courses/106/104/106104123/

Department of Computer Science and Engineering

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES

(An Autonomous Institution)

Title of the Course:	**Introduction To Quantum Technologies and Applications**
Category:	Professional Core
Couse Code:	23A0554T
Year:	III B. Tech
Semester:	I Semester
Branch:	CE,EEE,ME,ECE,CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: This course will be able to

1. Introduce fundamental quantum concepts like superposition and entanglement.
2. Understand theoretical structure of qubits and quantum information.
3. Explore conceptual challenges in building quantum computers.
4. Explain principles of quantum communication and computing.
5. Examine real-world applications and the future of quantum technologies

Course Outcomes

1. Explain core quantum principles in a non-mathematical manner.
2. Compare classical and quantum information systems.
3. Identify theoretical issues in building quantum computers.
4. Discuss quantum communication and computing concepts.
5. Recognize applications, industry trends, and career paths in quantum technology

Unit 1 Introduction to Quantum Theory and Technologies 09

The transition from classical to quantum physics, Fundamental principles explained conceptually: Superposition, Entanglement, Uncertainty Principle, Wave-particle duality, Classical vs Quantum mechanics – theoretical comparison, Quantum states and measurement: nature of observation, Overview of quantum systems: electrons, photons, atoms, The concept of quantization: discrete energy levels, Why quantum? Strategic, scientific, and technological significance, A snapshot of quantum technologies: Computing, Communication, and Sensing, National and global quantum missions: India's Quantum Mission, EU, USA, China

Unit 2 Theoretical Structure of Quantum Information Systems 09

What is a qubit? Conceptual understanding using spin and polarization, Comparison: classical bits vs quantum bits, Quantum systems: trapped ions, superconducting circuits, photons (non-engineering view),Quantum coherence and decoherence – intuitive explanation, Theoretical concepts: Hilbert spaces, quantum states, operators – only interpreted in abstract, The role of entanglement and non-locality in systems, Quantum information vs classical information: principles and differences, Philosophical implications: randomness, determinism, and observer role

Unit 3 Building a Quantum Computer – Theoretical Challenges and Requirements 09

What is required to build a quantum computer (conceptual overview)?,Fragility of quantum systems: decoherence, noise, and control, Conditions for a functional quantum system: Isolation, Error management, Scalability, Stability, Theoretical barriers:

Why maintaining entanglement is difficult, Error correction as a theoretical necessity, Quantum hardware platforms (brief conceptual comparison),Superconducting circuits, Trapped ions, Photonics, Vision vs reality: what's working and what remains elusive, The role of quantum software in managing theoretical complexities

Unit 4 Quantum Communication and Computing – Theoretical Perspective 09

Quantum vs Classical Information, Basics of Quantum Communication, Quantum Key Distribution (QKD), Role of Entanglement in Communication, The Idea of the Quantum Internet – Secure Global Networking, Introduction to Quantum Computing, Quantum Parallelism (Many States at Once), Classical vs Quantum Gates, Challenges: Decoherence and Error Correction, Real-World Importance and Future Potential

Unit 5 Applications, Use Cases, and the Quantum Future 09

Real-world application domains: Healthcare (drug discovery),Material science, Logistics and optimization, Quantum sensing and precision timing, Industrial case studies: IBM, Google, Microsoft, PsiQuantum, Ethical, societal, and policy considerations, Challenges to adoption: cost, skills, standardization, Emerging careers in quantum: roles, skillsets, and preparation pathways, Educational and research landscape – India's opportunity in the global quantum race

Textbooks:

1. Michael A. Nielsen, Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 10th Anniversary Edition, 2010.
2. Eleanor Rieffel and Wolfgang Polak, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
3. Chris Bernhardt, *Quantum Computing for Everyone*, MIT Press, 2019.

Reference Books:

1. David McMahon, *Quantum Computing Explained*, Wiley, 2008.
2. Phillip Kaye, Raymond Laflamme, Michele Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2007.
3. Scott Aaronson, *Quantum Computing Since Democritus*, Cambridge University Press, 2013.
4. **Alastair I.M. Rae**, *Quantum Physics: A Beginner's Guide*, Oneworld Publications, Revised Edition, 2005.
5. **Eleanor G. Rieffel, Wolfgang H. Polak**, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
6. **Leonard Susskind, Art Friedman**, *Quantum Mechanics: The Theoretical Minimum*, Basic Books, 2014.

7. **Bruce Rosenblum, Fred Kuttner**, *Quantum Enigma: Physics Encounters Consciousness*, Oxford University Press, 2nd Edition, 2011.
8. **GiulianoBenenti, GiulioCasati, GiulianoStrini**, *Principles of Quantum Computation and Information, Volume I: Basic Concepts*, World Scientific Publishing, 2004.
9. **K.B. Whaley et al.**, *Quantum Technologies and Industrial Applications: European Roadmap and Strategy Document*, Quantum Flagship, European Commission, 2020.
10. **Department of Science & Technology (DST), Government of India**, *National Mission on Quantum Technologies & Applications – Official Reports and Whitepapers*, MeitY/DST Publications, 2020 onward.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**Object Oriented Analysis and Design**
Category:	Professional Elective-I
Couse Code:	23A055AT
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. Describe the activities in the different phases of the object-oriented development lifecycle.
2. Understand the concepts of object-oriented model with the E-R and EER models.
3. Model a real-world application by using UML diagram.
4. Design architectural modelling.
5. Describing an application of UML.

Course Outcomes:

At the end of the course, the student will be able to

1. The importance of modelling in UML.
2. Compare and contrast the object-oriented model with the E-R and EER models.
3. Design use case diagram. Design an application using deployment diagram.
4. Apply UML diagrams to build library application.

Unit 1 09

Introduction to UML: Importance of modelling, principles of modelling, object-oriented modelling, conceptual model of the UML, Architecture, Software Development Life Cycle.

Unit 2 09

Basic Structural Modelling: Classes, Relationships, common Mechanisms, and diagrams. Advanced Structural Modelling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. Class & Object Diagrams: Terms, concepts, modelling techniques for Class & Object Diagrams.

Unit 3

09

Basic Behavioural Modelling-I: Interactions, Interaction diagrams. Basic Behavioural Modelling-II: Use cases, Use case Diagrams, Activity Diagrams.

Unit 4 09

Advanced Behavioral Modelling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. Architectural Modelling: Component, Deployment, Component diagrams and Deployment diagrams.

Unit 5 09

Patterns and Frameworks, Artifact Diagrams. Case Study: The Unified Library application.

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modelling Language User Guide, Pearson Education 2nd Edition.
2. Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modelling Software Systems Using UML2, WILEY-Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TMH.
5. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**Soft Computing**
Category:	Professional Core
Couse Code:	23A055BT
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. Familiarize with soft computing concepts
2. Introduce and use the idea of fuzzy logic and use of heuristics based on human experience
3. Familiarize the Neuro-Fuzzy modelling using Classification and Clustering techniques
4. Learn the concepts of Genetic algorithm and its applications
5. Acquire the knowledge of Rough Sets.

Course Outcomes:

1. Identify the difference between Conventional Artificial Intelligence to Computational Intelligence.
2. Understand fuzzy logic and reasoning to handle and solve engineering problems Apply the Classification techniques on various applications.
3. Perform various operations of genetic algorithms and Rough Sets.

Unit 1 10

Introduction to Soft Computing: Evolutionary Computing, "Soft" computing versus "Hard" computing, Soft Computing Methods, Recent Trends in Soft Computing, Characteristics of Soft computing, Applications of Soft Computing Techniques.

Unit 2 10

Fuzzy Systems: Fuzzy Sets, Fuzzy Relations, Fuzzy Logic, Fuzzy Rule-Based Systems

Unit 3 10

Fuzzy Decision Making, Particle Swarm Optimization.

Unit 4 10

Genetic Algorithms: Basic Concepts, Basic Operators for Genetic Algorithms, Crossover and Mutation Properties, Genetic Algorithm Cycle, Fitness Function, Applications of Genetic Algorithm.

Unit 5 10

Rough Sets, Rough Sets, Rule Induction, and Discernibility Matrix, Integration of Soft Computing Techniques.

TEXT BOOK:

1. Soft Computing – Advances and Applications - Jan 2015 by B.K. Tripathy and J. Anuradha – Cengage Learning

REFERENCE BOOKS:

1. S. N. Sivanandam & S. N. Deepa, ―Principles of Soft Computing‖, 2nd edition, Wiley India, 2008.
2. David E. Goldberg, ―Genetic Algorithms-In Search, optimization and Machine learning‖, Pearson Education.
3. J. S. R. Jang, C.T. Sun and E.Mizutani, ―Neuro-Fuzzy and Soft Computing‖, Pearson Education, 2004.
4. G.J. Klir & B. Yuan, ―Fuzzy Sets & Fuzzy Logic‖, PHI, 1995.
5. Melanie Mitchell, ―An Introduction to Genetic Algorithm‖, PHI, 1998.
6. Timothy J. Ross, ―Fuzzy Logic with Engineering Applications‖, McGraw- Hill International editions, 1995.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**Data Warehousing & Data Mining**
Category:	Professional Core
Couse Code:	23A055CT
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. Familiarize with mathematical foundations of data mining tools.
2. Introduce classical models and algorithms in data warehouses and data mining.
3. Investigate the kinds of patterns that can be discovered by association rule mining, classification and clustering.
4. Explore data mining techniques in various applications like social, scientific and environmental context.

Course Outcomes:

Upon completion of the course, the students should be able to:

1. Design a Data warehouse system and perform business analysis with OLAP tools
2. Apply suitable pre-processing and visualization techniques for data analysis
3. Apply frequent pattern and association rule mining techniques for data analysis
4. Design appropriate classification and clustering techniques for data analysis
5. Infer knowledge from raw data

Unit 1 09

Basic Concepts – Data Warehousing Components – Building a Data Warehouse – Database Architectures for Parallel Processing – Parallel DBMS Vendors – Multidimensional Data Model – Data Warehouse Schemas for Decision Support, Concept Hierarchies -Characteristics of OLAP Systems – Typical OLAP Operations, OLAP and OLTP.

Unit 2 09

Introduction to Data Mining Systems – Knowledge Discovery Process – Data Mining Techniques – Issues – applications- Data Objects and attribute types, Statistical description of data, Data Preprocessing – Cleaning, Integration, Reduction, Transformation and discretization, Data Visualization, Data similarity and dissimilarity measures.

Unit 3 08

Mining Frequent Patterns, Associations and Correlations – Mining Methods- Pattern Evaluation Method – Pattern Mining in Multilevel, Multi Dimensional Space – Constraint Based Frequent Pattern Mining, Classification using Frequent Patterns.

09

Unit 4

Decision Tree Induction – Bayesian Classification – Rule Based Classification – Classification by Back Propagation – Support Vector Machines — Lazy Learners – Model Evaluation and Selection- Techniques to improve Classification Accuracy. Clustering Techniques – Cluster analysis-Partitioning Methods – Hierarchical Methods – Density Based Methods – Grid Based Methods – Evaluation of clustering – Clustering high dimensional data- Clustering with constraints, Outlier analysis-outlier detection methods.

Unit 5 **WEKA TOOL** 08

Datasets – Introduction, Iris plants database, Breast cancer database, Auto imports database – Introduction to WEKA, The Explorer – Getting started, Exploring the explorer, Learning algorithms, Clustering algorithms, Association–rule learners.

TEXT BOOK:

1. Jiawei Han and Micheline Kamber, ―Data Mining Concepts and Techniques, Third Edition, Elsevier, 2012.

REFERENCES:

1. Alex Berson and Stephen J.Smith, ―Data Warehousing, Data Mining & OLAP‖, Tata McGraw – Hill Edition, 35th Reprint 2016.
2. K.P. Soman, Shyam Diwakar and V. Ajay, ―Insight into Data Mining Theory and Practice, Eastern Economy Edition, Prentice Hall of India, 2006.
3. Ian H.Witten and Eibe Frank, ―Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, Second Edition.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**Microprocessors And Microcontrollers**
Category:	Professional Elective-I
Couse Code:	23A045FT
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Course Objectives:

1. To comprehend the architecture, operation, and configurations of the 8086 microprocessors
2. To get familiar with 8086 programming concepts, instruction set, and assembly language development tools.
3. To study the interfacing of 8086 with memory, peripherals, and controllers for various applications.
4. To learn the architecture, instruction set, and programming of the 8051 microcontrollers.
5. To understand microcontroller interfacing techniques, peripheral programming, and processor comparisons.

Course Outcomes:

At the end of this course, the students will be able to

1. Gain knowledge on the architecture, operation, and configurations of the 8086 microprocessors
2. Get familiar with 8086 programming concepts, instruction set, and assembly language development tools.
3. Know the interfacing of 8086 with memory, peripherals, and controllers for various applications.
4. Learn the architecture, instruction set, and programming of the 8051 microcontrollers.
5. Understand microcontroller interfacing techniques, peripheral programming, and processor comparisons.

Unit 1 10

8086 Architecture: Main features, pin diagram/description, 8086 microprocessor family, internal architecture, bus interfacing unit, execution unit, interrupts and interrupt response, 8086 system timing, minimum mode and maximum mode configuration.

Unit 2 10

8086 Programming: Program development steps, instructions, addressing modes, assembler directives, writing simple programs with an assembler, assembly language program development tools.

Unit 3 10

8086 Interfacing: Semiconductor memories interfacing (RAM, ROM), Intel 8255 programmable peripheral interface, Interfacing switches and LEDS, Interfacing seven segment displays, software and hardware interrupt applications, Intel 8251 USART architecture and interfacing, Intel 8237a DMA controller, stepper motor, A/D and D/A converters, Need for 8259 programmable interrupt controllers.

Unit 4

Microcontroller - Architecture of 8051 – Special Function Registers (SFRs) - I/O Pins Ports and Circuits - Instruction set - Addressing modes - Assembly language programming.

Unit 5

10

Interfacing Microcontroller - Programming 8051 Timers - Serial Port Programming - Interrupts Programming – LCD & Keyboard Interfacing - ADC, DAC & Sensor Interfacing - External Memory Interface- Stepper Motor and Waveform generation - Comparison of Microprocessor, Microcontroller, PIC and ARM processors

Textbooks:

1. Microprocessors and Interfacing – Programming and Hardware by Douglas V Hall, SSSP Rao, Tata McGraw Hill Education Private Limited, 3rdEdition,1994.
2. K M Bhurchandi, A K Ray, Advanced Microprocessors and Peripherals, 3rd edition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2nd edition, Pearson, 2012.

References:

1. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 6th edition, Penram International Publishing, 2013.
2. Kenneth J. Ayala, The 8051 Microcontroller, 3rd edition, Cengage Learning, 2004.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course: Artificial Intelligence Lab
Category: Open Elective Lab-I
Couse Code: 23A0551L
Year: III B. Tech
Semester: I Semester
Branch: CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	3	1.5

Course Objectives:

1. To teach the methods of implementing algorithms using artificial intelligence techniques
2. To illustrate search algorithms
3. To demonstrate the building of intelligent agents

Course Outcomes:

After completion of the course, students will be able to
1. Implement search algorithms.
2. Solve Artificial intelligence problems
3. Design chatbot and virtual assistant

Exercise 1

1. Write a program to implement DFS and BFS

Exercise 2

2. Write a Program to find the solution for traveling salesman Problem

Exercise 3

3. Write a program to implement Simulated Annealing Algorithm

Exercise 4

4.Write a program to find the solution for the wumpus world problem

Exercise 5

5. Write a program to implement 8 puzzle problem

Exercise 6

6. Write a program to implement Towers of Hanoi problem

Exercise 7

6. Write a program to implement A* Algorithm

Exercise 8

8. Write a program to implement Hill Climbing Algorithm

Exercise 9

9. Build a Chatbot using AWS Lex, Pandora bots.

Exercise 10

10. Build a bot that provides all the information related to your college.

Exercise 11

11. Build a virtual assistant for Wikipedia using Wolfram Alpha and Python

Exercise 12

12. The following is a function that counts the number of times a string occurs in another string:
# Count the number of times string s1 is found in string s2 defcountsubstring(s1,s2):
count = 0
for i in range(0,len(s2)- len(s1)+1): if s1 == s2[i:i+len(s1)]:
count += 1
return count
For instance, count substring (‘ab‘,‘cabalaba‘) returns 2.
Write a recursive version of the above function. To get the rest of a string (i.e. everything but the first character).

Exercise 13

13. Higher order functions. Write a higher-order function count that counts the number of elements in a list that satisfy a given test. For instance: count (lambda x: x>2, [1, 2, 3, 4, 5]) should return 3, as there are three elements in the list larger than 2. Solve this task without using any existing higher order function

Exercise 14

14. Brute force solution to the Knapsack problem. Write a function that allows you to generate random problem instances for the knapsack program. This function should generate a list of items containing N items that each have a unique name, a random size in the range 1….5 and a random value in the range 1…….10.
Next, you should perform performance measurements to see how long the given knapsack solver take to solve different problem sizes. You should perform at least 10 runs with different randomly generated problem instances for the problem sizes 10,12,14,16,18,20 and 22. Use a backpack size of 2:5 x N for each value problem size N. Please note that the method used to generate

random numbers can also affect performance, since different distributions of values can make the initial conditions of the problem slightly more or less demanding.
How much longer time does it take to run this program when we increase the number of items? Does the backpack size affect the answer?
Try running the above tests again with a backpack size of 1 x N and with 4:0 x N.

Exercise 15

15. Assume that you are organising a party for N people and have been given a list L of people who, for social reasons, should not sit at the same table. Furthermore, assume that you have C tables (that are infinitely large).
Write a function layout (N,C,L) that can give a table placement (i.e. a number from 0 : : :C -1) for each guest such that there will be no social mishaps.
For simplicity we assume that you have a unique number 0 N-1 for each guest and that the list of restrictions is of the form [(X, Y)] denoting guests X, Y that are not allowed to sit together. Answer with a dictionary mapping each guest into a table assignment, if there are no possible layouts of the guests you should answer False.

References:

1. David Poole, Alan Mack worth, Randy Goebel,‖ Computational Intelligence: a logical approach‖, Oxford University Press, 2004.
2. G. Luger, ―Artificial Intelligence: Structures and Strategies for complex problem solving‖, Fourth Edition, Pearson Education, 2002.
3. J. Nilsson, ―Artificial Intelligence: A new Synthesis‖, Elsevier Publishers, 1998.
4. Artificial Neural Networks, B. Yagna Narayana, PHI
5. Artificial Intelligence, 2nd Edition, E.Rich and K.Knight, TMH.
6. Artificial Intelligence and Expert Systems, Patterson, PHI.

Online Learning Resources/Virtual Labs:

https://www.tensorflow.org/https://pytorch.org/
https://github.com/pytorch https://keras.io/ https://github.com/kerasteam
http://deeplearning.net/software/theano/
https://github.com/Theano/Theanohttps://caffe2.ai/
https://github.com/caffe2
https://deeplearning4j.org/Scikit-learn:https://scikit-learn.org/stable/
https://github.com/scikit-learn/scikit-learn
https://www.deeplearning.ai/
https://opencv.org/
https://github.com/qqwweee/keras-yolo3
https://www.pyimagesearch.com/2018/11/12/yolo-object-detection- with-opencv/
https://developer.nvidia.com/cuda-math-library
http://vlabs.iitb.ac.in/vlabs-dev/labs/machine_learning/labs/index.php

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**COMPUTER NETWORKS LAB**
Category:	Professional Core Lab
Couse Code:	23A0552L
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	3	1.5

Course Objectives:

1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topology and observe its performance
3. To analyze the traffic flow and the contents of protocol frames.
4. Familiarize with the applications of Internet.

Course Outcomes:

After completion of the course, students will be able to

1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topologyand observe its performance.
3. To analyze the traffic flow and the contents of protocol frames.
4. Critique the existing routing protocols

Exercise 1

Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.

Exercise 2

Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP

Exercise 3

Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.

Exercise 4

Implement Dijkastra‘s algorithm to compute the shortest path through a network

Exercise 5

Take an example subnet of hosts and obtain a broadcast tree for the subnet.

Exercise 6

Implement distance vector routing algorithm for obtaining routing tables at each node.

Exercise 7
Implement data encryption and data decryption

Exercise 8
Write a program for congestion control using Leaky bucket algorithm.

Exercise 9
Write a program for frame sorting technique used in buffers.

Exercise 10
Programs using Wireshark

i. Packet Capture Using Wire shark
ii. Starting Wire shark
iii. Viewing Captured Traffic
iv. Analysis and Statistics & Filters..

Exercise 11
How to run Nmap scan

Exercise 12
Operating System Detection using Nmap

Exercise 13
Do the following using NS2 Simulator

i. NS2 Simulator-Introduction
ii. Simulate to Find the Number of Packets Dropped
iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
iv. Simulate to Find the Number of Packets Dropped due to Congestion
v. Simulate to Compare Data Rate& Throughput.
vi. Simulate to Plot Congestion for Different Source/Destination
vii. Simulate to Determine the Performance with respect to transmission of Packets

Text Books:

1. Andrew S.Tanenbaum, David j.wetherall, Computer Networks, 6th Edition, PEARSON.

2. James F.Kurose, Keith W. Ross, Computer Networking: A Top-Down 6th edition, Pearson, 2019. 2. Computer Networks: A Systems Approach-Bruce Davie, VMware-Larry Peterson, Princeton University-2019.

References:

1. Computer Networks–B. K. MathanNagan, T. Mahalakshmi- Charulatha Publications PrivateLimited-2019.
2. Computer Networks-Dr.Amol V. Dhumane Nitin N. Sakhare-NiraliPrakashan Publishers-2024 3. Data Communications and Networking with TCPIP Protocol Suite-Behrouz A. Forouzan-McGraw Hill-6th Edition

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course: FULL STACK DEVELOPMENT – II
Category: Skill Enhancement Course
Couse Code: 23A0555L
Year: III B. Tech
Semester: I Semester
Branch: CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	1	2	2

Course Objectives: The main objectives of the course are to

1. Make use of Modern- day JavaScript with ES6 standards for designing Dynamic web pages
2. Building robust & responsive User Interfaces using popular JavaScript library _React.js‘. Building robust backend APIs using _Express. js‘
3. Establishing the connection between frontend and backend APIs (Express) with Data Bases(My SQL)
4. Familiarize students with GitHub for remote repository hosting and collaborative development.

Course Outcomes:

CO1: Building fast and interactive UIs
CO2: Applying Declarative approach for developing web apps
CO3: Understanding ES6 features to embrace modern JavaScript
CO4: Building reliable APIs with Express. Js
CO5: Create and manage Git repositories, track changes, and push code to GitHub.

Experiments covering the Topics:

1. Introduction to DOM (Document Object Model), Ecma Script (ES6) standards and features like Arrow functions, Spread operator, Rest operator, Type coercion, Type hoisting, String literals, Array and Object Destructuring.
2. Basics of React. js like React Components, JSX, Conditional rendering
3. Differences between Real DOM and Virtual DOM.
4. Important React.js concepts like React hooks, Props, React forms, Fetch API, Iterative rendering using JavaScript map() function.
5. JavaScript runtime environment node. js and its uses, Express. js and Routing, Micro-Services architecture and MVC architecture, database connectivity using (My SQL)
6. Introduction to My SQL, setting up MySQL and configuring, Databases, My SQL queries, subqueries, creating My SQL driver for database connectivity to Express. js server.
7. Introduction to Git and GitHub and upload project& team collaboration

Exercise 1

1. **Introduction to Modern JavaScript and DOM**
   a. Write a JavaScript program to link JavaScript file with the HTML page
   b. Write a JavaScript program to select the elements in HTML page using selectors
   c. Write a JavaScript program to implement the event listeners
   d. Write a JavaScript program to handle the click events for the HTML button elements
   e. Write a JavaScript program to With three types of functions
      i. Function declaration
      ii. Function definition
      iii. Arrow functions

Exercise 2

2. **Basics of React. Js**
   a. Write a React program to implement a counter button using react class components
   b. Write a React program to implement a counter button using react functional components
   c. Write a React program to handle the button click events in functional component
   d. Write a React program to conditionally render a component in the browser
   e. Write a React program to display text using String literals

Exercise 3

3.Important concepts of React. js
   a. Write a React program to implement a counter button using React use State hook
   b. Write a React program to fetch the data from an API using React use Effect hook
   c. Write a React program with two react components sharing data using Props.
   d. Write a React program to implement the forms in react
   e. Write a React program to implement the iterative rendering using map() function

Exercise 4

Setup

- Install Git on local machine.
- Configure Git (user name, email).
- Create GitHub account and generate a personal access token.

b. **Basic Git Workflow**

- Create a local repository using git init
- Create and add files → git add .
- Commit files → git commit -m "Initial commit"
- Connect to GitHub remote → git remote add origin <repo_url>
- Push to GitHub → git push -u origin main

c. **Branching and Collaboration**

- Create a branch → git checkout -b feature1

- Merge branch to main → git merge feature1
- Resolve merge conflicts (guided)

Exercise 5

Upload React Project to GitHub

Create a new React app using npx create-react-app myapp

o Initialize a git repo and push to GitHub

o Use .gitignore to exclude node_modules

o Create multiple branches: feature/navbar, feature/form

o Practice merge and pull requests (can use GitHub GUI)

Exercise 6

Introduction to Node. js and Express. js

a. Write a program to implement the _hello world' message in the route through the browser using Express

b. Write a program to develop a small website with multiple routes using Express. js

c. Write a program to print the _hello world' in the browser console using Express. Js

d. Write a program to implement the CRUD operations using Express. js

e. Write a program to establish the connection between API and Database using Express – My SQL driver

Exercise 7

Introduction to My SQL

a. Write a program to create a Database and table inside that database using My SQL Command line client

b. Write a My SQL queries to create table, and insert the data, update the data in the table

c. Write a My SQL queries to implement the subqueries in the My SQL command line client

d. Write a My SQL program to create the script files in the My SQL workbench

e. Write a My SQL program to create a database directory in Project and initialize a database. sql file to integrate the database into API

Exercise 8

Introduction to My SQL

- Form groups of 2–3 students
- Create a shared GitHub repo
- Assign tasks and work in branches
- Use Issues, Pull Requests, and Code Reviews
- Document code with README.md

Textbooks:

1. Web Design with HTML, CSS, JavaScript and JQuery Set Book by Jon Duckett Professional JavaScript for Web Developers Book by Nicholas C. Zakas
2. John Dean, Web Programming with HTML5, CSS and JavaScript, Jones & Bartlett Learning, 2019.

3. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.
4. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon
5. AZAT MARDAN, Full Stack Java Script: Learn Back bone. js, Node.jsand Mongo DB.2015

Reference Books:

1. Full-Stack JavaScript Development by Eric Bush.
2. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.
3. Tomasz Dyl , Kamil Przeorski , Maciej Czarnecki, Mastering Full Stack React Web Development 2017

Online Learning Resources:

1. https://ict.iitk.ac.in/product/full-stack-developer-html5-css3-js-bootstrap-php-4/
2. https://www.w3schools.com/html
3. https://www.w3schools.com/css
4. https://www.w3schools.com/js/
5. https://www.w3schools.com/nodejs
6. https://www.w3schools.com/typescript

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**Tinkering Lab for Computer Engineers**
Category:	Skill Enhancement Course
Couse Code:	23A0556L
Year:	III B. Tech
Semester:	I Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	2	1

The aim of tinkering lab for engineering students is to provide a hands-on learning environment where students can explore, experiment, and innovate by building and testing prototypes. These labs are designed to demonstrate practical skills that complement theoretical knowledge.

Course Objectives: The objectives of the course are to

1. Encourage Innovation and Creativity
2. Provide Hands-on Learning and Impart Skill Development
3. Foster Collaboration and Teamwork
4. Enable Interdisciplinary Learning, Prepare for Industry and Entrepreneurship
5. Impart Problem-Solving mind-set

These labs bridge the gap between academia and industry, providing students with the practical experience. Some students may also develop entrepreneurial skills, potentially leading to start ups or innovation-driven careers. Tinkering labs aim to cultivate the next generation of engineers by giving them the tools, space, and mind-set to experiment, innovate, and solve real-world challenges.

List of experiments:

1) Make your own parallel and series circuits using breadboard for any application of your choice.
2) Design and 3D print a Walking Robot
3) Design and 3D Print a Rocket.
4) Temperature & Humidity Monitoring System (DHT11 + LCD)
5) Water Level Detection and Alert System
6) Automatic Plant Watering System
7) Bluetooth-Based Door Lock System
8) Smart Dustbin Using Ultrasonic Sensor
9) Fire Detection and Alarm System
10) RFID-Based Attendance System
11) Voice-Controlled Devices via Google Assistant
12) Heart Rate Monitoring Using Pulse Sensor
13) Soil Moisture-Based Irrigation

14) Smart Helmet for Accident Detection
15) Milk Adulteration Detection System
16) Water Purification via Activated Carbon
17) Solar Dehydrator for Food Drying
18) Temperature-Controlled Chemical Reactor
19) Ethanol Mini-Plant Using Biomass
20) Smart Fluid Flow Control (Solenoid + pH Sensor)
21) Portable Water Quality Tester
22) AI Crop Disease Detection
23) AI-based Smart Irrigation
24) ECG Signal Acquisition and Plotting25) AI-Powered Traffic Flow Prediction
26) Smart Grid Simulation with Load Monitoring
27) Smart Campus Indoor Navigator
28) Weather Station Prototype
29) Firefighting Robot with Sensor Guidance
30) Facial Recognition Dustbin
31) Barcode-Based Lab Inventory System
32) Growth Chamber for Plants
33) Biomedical Waste Alert System
34) Soil Classification with AI
35) Smart Railway Gate
36) Smart Bin Locator via GPS and Load Sensors
37) Algae-Based Water Purifier
38) Contactless Attendance via Face Recognition

Note: The students can also design and implement their own ideas, apart from the list of experiments mentioned above.
Note: A minimum of 8 to 10 experiments must be completed by the students.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**INTRODUCTION TO MACHINE LEARNING**
Category:	Professional Core
Couse Code:	23A0561T
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: The course is introduced for students to

1. To introduce the fundamental concepts and types of machine learning
2. To develop a deep understanding of supervised and unsupervised learning algorithms.
3. To understand mathematical foundations of learning models and algorithms.
4. To evaluate model performance using appropriate statistical and analytical tools.
5. To apply machine learning techniques to solve real-world problems using tools such as Scikit learn.

Course Outcomes:

After completion of the course, students will be able to:

1. Understand and distinguish among different types of learning methods.
2. Apply supervised and unsupervised learning algorithms to datasets
3. Analyze model performance using cross-validation and error metrics
4. Build, test, and improve machine learning models for classification and prediction.
5. Use Python-based libraries (e.g., Scikit-learn) to implement ML algorithms.

Unit 1 Introduction to Machine Learning and Linear Models

Definition and Scope of Machine Learning, Applications and Types of Learning: Supervised, Unsupervised, Reinforcement, Linear Regression: Least Squares, Cost Function, Gradient Descent, Polynomial Regression and Overfitting, Evaluation Metrics: RMSE, MAE, R^2 Score, Bias-Variance Trade off.

Unit 2 Classification Algorithms

Classification Overview and Decision Boundaries, Logistic Regression: Sigmoid Function and Cost, K Nearest Neighbors (KNN), Naïve Bayes Classifier, Decision Trees and Random Forests, Model Evaluation: Confusion Matrix, Precision, Recall, F1-Score.

Unit 3 Support Vector Machines and Ensemble Methods

Support Vector Machines: Concepts, Kernels, Hyperplane and Margin Concepts, Kernel Tricks: RBF and Polynomial, Ensemble Learning: Bagging, Boosting, and Voting, Gradient Boosting, AdaBoost, and XGBoost, Model Tuning and Hyperparameter Optimization.

Unit 4 Unsupervised Learning Techniques

Clustering Overview: Applications, K-Means Clustering Algorithm, Hierarchical Clustering, DBSCAN and Density-Based Methods, Principal Component Analysis (PCA) for Dimensionality Reduction, Silhouette Score, Davies-Bouldin Index for Cluster Validation.

Unit 5 Advanced Topics and Applications

Reinforcement Learning Basics and Markov Decision Processes, Introduction to Neural Networks and Deep Learning, Cross-Validation Techniques: k-Fold, Leave-One-Out, Feature Engineering and Feature Selection, Deployment of ML Models (Flask, Streamlit, etc.), Case Studies: Medical Diagnosis, Spam Detection, Credit Scoring.

Textbooks:

1.Tom Mitchell, Machine Learning, McGraw-Hill Education.

2.Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.

3.Ethem Alpaydin, Introduction to Machine Learning, MIT Press.

Reference Books:

1.Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer..

2. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press.

3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer.

Online Learning Resources:

1. Coursera – Machine Learning by Andrew Ng (Stanford University)
2. Scikit-learn Documentation
3. Kaggle Learn – Machine Learning
4. Google's Machine Learning Crash Course
5. YouTube – StatQuest with Josh Starmer

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**Cloud Computing**
Category:	Professional Core
Couse Code:	23A0562T
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. To explain the evolving computer model called cloud computing.
2. To introduce the various levels of services that can be achieved by cloud.
3. To describe the security aspects in cloud

Course Outcomes:

After completion of the course, students will be able to

1. Ability to create cloud computing environment
2. Ability to design applications for Cloud environment
3. Design & develop back up strategies for cloud data based on features.
4. Use and Examine different cloud computing services.
5. Apply different cloud programming model as per need.

Unit 1 Basics of Cloud computing 08

Introduction to cloud computing: Introduction, Characteristics of cloud computing, Cloud Models, Cloud Services Examples, Cloud Based services and applications

Cloud concepts and Technologies: Virtualization, Load balancing, Scalability and Elasticity, Deployment, Replication, Monitoring, Software defined, Network function virtualization, Map Reduce, Identity and Access Management, services level Agreements, Billing.

Cloud Services and Platforms: Computer Services, Storage Services, Database Services, Application services, Content delivery services Analytics Services, Deployment and Management Services, Identity and Access Management services, Open Source Private Cloud software.

Unit 2 Hadoop and Python 09

Hadoop Map Reduce: Apache Hadoop, Hadoop Map Reduce Job Execution, Hadoop Schedulers, Hadoop Cluster setup.

Cloud Application Design: Reference Architecture for Cloud Applications, Cloud Application Design Methodologies, Data Storage Approaches.

Python Basics: Introduction, Installing Python, Python data Types & Data Structures, Control flow, Function, Modules, Packages, File handling, Date/Time Operations, Classes.

Unit 3 **Python for Cloud computing** 08

Python for Cloud: Python for Amazon web services, Python for Google Cloud Platform, Python for windows Azure, Python for Map Reduce, Python packages of Interest, Python web Application Frame work, Designing a REST full web API.

Cloud Application Development in Python: Design Approaches, Image Processing APP, Document Storage App, Map Reduce App, Social Media Analytics App.

Unit 4 **Big data, multimedia and Tuning** 08

Big Data Analytics: Introduction, Clustering Big Data, Classification of Big data Recommendation of Systems.

Multimedia Cloud: Introduction, Case Study: Live video Streaming App, Streaming Protocols, case Study**:** Video Trans coding App.

Cloud Application Bench marking and Tuning: Introduction, Work load Characteristics, Application Performance Metrics, Design Considerations for a Bench marking Methodology, Bench marking Tools, Deployment Prototyping, Load Testing & Bottleneck Detection case Study, Hadoop bench marking case Study.

Unit 5 **Applications and Issues in Cloud** 09

Cloud Security: Introduction, CSA Cloud Security Architecture, Authentication, Authorization, Identity Access Management, Data Security, Key Management, Auditing.

Cloud for Industry, Health care & Education: Cloud Computing for Health care, Cloud computing for Energy Systems, Cloud Computing for Transportation Systems, Cloud Computing for Manufacturing Industry, Cloud computing for Education.

Migrating in to a Cloud: Introduction, Broad Approaches to migrating into the cloud, the seven– step model of migration in to a cloud.

Organizational readiness and change Management in the Cloud Age: Introduction, Basic concepts of Organizational Readiness, Drivers for changes: A frame work to comprehend the competitive environment, common change management models, change management maturity models, Organizational readiness self– assessment.

Legal Issues in Cloud Computing: Introduction, Data Privacy and security Issues, cloud contracting models, Jurisdictional issues raised by virtualization and at a location, commercial and business considerations, Special Topics.

Textbooks:

1. Cloud computing A hands - on Approach ǁBy Arshdeep Bahga, Vijay Madisetti, Universities Press, 2016
2. Cloud Computing Principles and Paradigms: By RajKumar Buyya, James Broberg, Andrzej Goscinski, Wiley, 2016

Reference Books:

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, TMH
2. Cloud computing AHands-On Approach by Arshdeep Bahga and Vijay Madisetti.

3. Cloud Computing: A Practical Approach, Anthony T.Velte, To by J.Velte, Robert Elsenpeter, Tata Mc Graw Hill, rp 2011.
4. Enterprise Cloud Computing, Gautam Shroff, Cambridge University Press, 2010.
5.Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, George Reese,O_Reilly, SPD, rp 2011.
6. Essentials of Cloud Computing by K.Chandrasekaran. CRC Press.

Online Learning

Cloud computing –Course nptel.ac.in)

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**CRYPTOGRAPHY & NETWORK SECURITY**
Category:	Professional Core
Couse Code:	23A0563T
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: This course aims at training students to master the:

1. The concepts of classical encryption techniques and concepts of finite fields and number theory
2. Working principles and utilities of various crypto graphic algorithms including secret key cryptography, hashes, and message digests, and public key algorithms
3. Design issues and working principles of various authentication protocols, PKI standards
4. Various secure communication standards including Kerberos, I Psec, TLS and email
5. Concepts of cryptographic utilities and authentication mechanisms to design secure applications

Course Outcomes:

After completion of the course, students will be able to

1. Identify information security goals, classical encryption techniques and acquire fundamental knowledge on the concepts off in it fields and number theory
2. Compare and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication
3. Apply the knowledge of crypto graphic check sums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes.
4. Demonstrate the ability to apply user authentication principles including Kerberos for secure authentication
5. Gain proficiency in securing web communications using TLS and HTTPS, manage secure remote access with SSH, and design firewall policies

Unit 1 **Computer and Network Security Concepts** 09

Computer and Network Security Concepts: Computer Security Concepts, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, **Classical Encryption Techniques:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography,

Block Ciphers: Traditional Block Cipher Structure, The Data Encryption Standard, Advanced Encryption Standard: AES Structure, AES Transformation Functions

Unit 2 Number Theory 09

The Euclidean Algorithm, Modular Arithmetic, Fermat's and Euler's Theorems, The Chinese Remainder Theorem, Discrete Logarithms, Finite Fields: Finite Fields of the Form GF(p), Finite Fields of the Form GF(2n) .Public Key Cryptography: Principles, Public Key Cryptography Algorithms, RSA Algorithm, Diffie Hellman Key Exchange, Elliptic Curve Cryptography.

Unit 3 Cryptographic Hash Functions: 09

Cryptographic Hash Functions: Application of Cryptographic Hash Functions, Requirements & Security, Secure Hash Algorithm, Message Authentication Functions, Requirements & Security, HMAC &CMAC. Digital Signatures: NIST Digital Signature Algorithm, Distribution of Public Keys, X.509 Certificates, Public- Key Infrastructure

Unit 4 User Authentication 09

User Authentication : Remote User Authentication Principles, Kerberos. Electronic Mail Security: Pretty Good Privacy (PGP) And S/MIME.
IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange.

Unit 5 Transport Level Security 08

Transport Level Security: Web Security Requirements, Transport Layer Security (TLS), HTTPS, Secure Shell (SSH).
Firewalls: Fire wall Character is tics and Access Policy, Types of Fire walls, Fire wall Location and Configurations.

Textbooks:

1) Cryptography and Network Security – William Stallings, Pearson Education, 8th Edition.
2) Cryptography, Network Security and Cyber Laws–Bernard Menezes,Cengage Learning, 2010 edition

Reference Books:

1) Cryptography and Network Security-BehrouzA Forouzan, Debdeep Mukhopadhyaya, Mc-Graw Hill, 3rd Edition, 2015.
2) Network Security Illustrated, Jason Albanese and Wes Sonnenreich, MGH Publishers, 2003.

Online Learning Resources:

1) https://nptel.ac.in/courses/106/105/106105031/lecture
2) https://nptel.ac.in/courses/106/105/106105162/lecturebyDr.SouravMukhopadhyayIIT Kharagpur[Video Lecture]
3) https://www.mitel.com/articles/web-communication-cryptography-and-network-security web articles by Mitel Power Connections

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**SOFTWARE TESTING METHODOLOGIES**
Category:	**(Professional Elective-II)**
Couse Code:	23A056AT
Year:	III B. Tech
Semester:	II Semester
Branch:	**CSE**

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. To study the fundamental concepts of software testing which includes objectives, process, criteria, strategies, and methods
2. To discuss various software testing types and levels of testing like black and white box testing along with levels unit test, integration, regression, and system testing.
3. It also helps to learn the types of bugs, testing levels with which the student can very well identify a bug and correct as when it happens.
4. It provides knowledge on transaction flow testing and data flow testing techniques so that the flow of the program is tested as well.
5. To learn the domain testing, path testing and logic-based testing to explore the testing process easier.

Course Outcomes:

1. Know the basic concepts of software testing and its essentials.
2. Able to identify the various bugs and correcting them after knowing the consequences of the bug.
3. Use of program‘s control flow as a structural model is the corner stone of testing.
4. Performing functional testing using control flow and transaction flow graphs.

Unit 1 09

Introduction: -Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs, Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

Unit 2 08

Transaction Flow Testing:-transaction flows, transaction flow testing techniques. Dataflow testing:- Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

Unit 3 08

Domain Testing:-domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

Unit 4 09

Paths, Path products and Regular expressions:- path products &path expression, reduction procedure, applications, regular expressions & flow anomaly detection. Logic Based Testing:- over view, decision tables, path expressions, kv charts, specifications.

Unit 5 Transport Level Security 09

State, State Graphs and Transition testing:- state graphs, good & bad state graphs, state testing, Testability tips. Graph Matrices and Application:-Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools

Textbooks:

1. Software Testing techniques – Boris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K.V.K.K. Prasad, Dreamtech.

Reference Books:

1. The craft of software testing – Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**CYBER SECURITY**
Category:	**Professional Elective-II**
Couse Code:	23A056BT
Year:	III B. Tech
Semester:	II Semester
Branch:	**CSE**

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

The course is designed to provide awareness on different cybercrimes, cyber offenses, tools and methods used in cybercrime.

Course Outcomes:

After completion of the course, students will be able to

1. Classify the cybercrimes and understand the Indian ITA 2000
2. Analyse the vulnerabilities in any computing system and find the solutions
3. Predict the security threats of the future
4. Investigate the protection mechanisms
5. Design security solutions for organizations

Unit 1 Introduction to Cybercrime 8

Introduction, Cybercrime, and Information Security, Who are Cybercriminals, Classifications of Cybercrimes, And Cybercrime: The legal Perspectives and Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes.

Unit 2 Cyber Offenses: How Criminals Plan Them 9

Introduction, How Criminals plan the Attacks, Social Engineering, Cyber stalking, Cyber cafe and Cybercrimes, Botnets: The Fuel for Cybercrime, Attack Vector, Cloud Computing

Unit 3 Cybercrime: Mobile and Wireless Devices 9

Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones.

Mobile Devices:

Security Implications for Organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops.

Unit 4 Tools and Methods Used in Cybercrime 8

Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Key loggers and Spywares, Virus and Worms, Trojan Horse and Backdoors, Steganography, DoS and DDoS attacks, SQL Injection, Buffer Overflow.

Unit 5 Cyber Security: Organizational Implications 8

Introduction, Cost of Cybercrimes and IPR issues, Web threats for Organizations, Security and Privacy Implications, Social media marketing: Security Risks and Perils for Organizations, Social Computing and the associated challenges for Organizations.

Prescribed Textbooks:

Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Nina Godbole and Sunil Belapure, Wiley INDIA.

Reference Books:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan- Hwa(john) Wu,J. David Irwin.CRC Press T&F Group

Online Learning Resources:

http://nptel.ac.in/courses/106105031/40
http://nptel.ac.in/courses/106105031/39
http://nptel.ac.in/courses/106105031/38-O29szjTrs_O

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**DevOps**
Category:	**Professional Elective-II**
Couse Code:	23A056CT
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Pre-requisite:
Fundamentals of software development and maintenance

Course Objectives: This course will be able to

1. Understand collaboration and productivity by automating infrastructure and workflows
2. Familiarize with continuous measuring applications performance

Course Outcomes:

After completion of the course, students will be able to

1. Enumerate the principles of continuous development and deployment, automation of configuration management, inter-team collaboration, and IT serviceability
2. Describe Dev Ops & Dev Sec Ops methodologies and their key concepts
3. Illustrate the types of version control systems, continuous integration tools, continuous monitoring tools, and cloud models
4. Setup Complete private Infrastructure using Version Control systems and CI/CD tools

Unit 1 8

Dev Ops: An Overview, Dev Ops: Origins, Dev Ops: Roots, Dev Ops: Practices Dev Ops: Culture. **Adopting Dev Ops**: Developing the Playbook. Developing a Business Case for a Dev Ops: Developing the Business Case

Unit 2 9

Completing the Business Model Canvas, Customer Segments, Value Segments, Value Propositions, Channels, Customer Relationships, Revenue Streams, Key Resources, Key

Activities, Key Partnerships, Cost Structures. Dev Ops Plays for Optimizing the delivery Pipeline: Dev Ops as an optimization Exercise, Core Themes, The Dev Ops Plays, Specializing Core Plays

Unit 3 8

Dev Ops Plays for Driving Innovation: Optimize to Innovate, The Uber Syndrome, Innovation and the Role of Technology, Core Themes, play: Build a Dev Ops Platform, play: Deliver Micro services Architectures, play: DevOps an API Economy, play: Organizing for Innovation.

Unit 4 10

Scaling Dev Ops for the Enterprise: Core Themes, play: Dev Ops Centre of Competency, play: Developing Culture of Innovation at Scale, play: Developing a Culture of continuous Improvement, play: Team Models for Dev Ops, play: Standardization of Tools and Process, play: Security Considerations for Dev Ops, Play: Dev Ops and Outsourcing.

Unit 5 10

Leading Dev Ops Adoption in the Enterprise: Play: Dev Ops as a transformation Exercise, play: Developing a Culture of Collaboration and Trust, play: Dev Ops Thinking for the Line of Business, play: starting with Pilot Projects, Play: Rearing Unicorns on an Aircrafts Carrier. Appendix Case Study: Example Dev Ops Adoption Roadmap Organization Background, Roadmap Structure, Adoption Roadmap.

Prescribed Textbooks:

1. Sanjeev Sharma, The Dev Ops Adoption Playbook, Published by John Wiley & Sons, Inc.2017

Reference Books:

1. Sanjeev Sharma & Bernie Coyne, Dev Ops for Dummies, Published by John Wiley & Sons, Inc.
2. Michael Huttermann, Dev Ops for Developers, Apress publishers,2012.

Online Learning Resources:

Learning DevOps through terra form infrastructure Automation Course | Udemy

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**EMBEDDED SYSTEMS DESIGN**
Category:	Professional Elective-II
Couse Code:	23A056DT
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: This course will be able to

1. To understand the history, classification, and design process of embedded systems.
2. To explore the core components of embedded systems, including processors, memory, and I/O components.
3. To introduce onboard and external communication interfaces used in embedded systems.
4. To explain different firmware design approaches and programming techniques for embedded systems.
5. To provide an understanding of real-time operating systems and task management in embedded systems.

Course Outcomes:

At the end of the course, the student will be able to

1. Classify embedded systems based on their purpose, generation, and
2. Identify and select appropriate hardware components for an embedded system design.
3. Differentiate and implement various communication protocols like I2C, SPI, and CAN.
4. Develop firmware using assembly and high-level programming languages.

5. Analyze and apply RTOS-based task scheduling and synchronization techniques.

Unit 1 Introduction to Embedded Systems 09

History of embedded systems, Classification of embedded systems based on generation and complexity, Purpose of embedded systems, The embedded system design process-requirements, specification, architecture design, designing hardware and software, components, system integration, Applications of embedded systems, and characteristics of embedded systems.

Unit 2 Typical Embedded System 12

Core of the embedded system-general purpose and domain specific processors, ASICs, PLDs, COTs; Memory-ROM, RAM, memory according to the type of interface, memory shadowing, memory selection for embedded systems, Sensors, actuators, I/O components: seven segment LED, relay, piezo buzzer, push button switch, other sub-systems: reset circuit, brownout protection circuit, oscillator circuit real time clock, watch dog timer.

Unit 3 Communication Interface 09

Onboard communication interfaces-I2C, SPI, CAN, parallel interface; External communication interfaces-RS232 and RS485, USB, infrared, Bluetooth, Wi-Fi, ZigBe, GPRS, GSM.

Unit 4 Embedded Firmware Design and Development 09

Embedded firmware design approaches-super loop based approach, operating system based approach; embedded firmware development languages-assembly language based development, high level language based development

Unit 5 RTOS based Embedded System Design 09

Operating system basics, types of operating systems, tasks, process and threads, multiprocessing and multitasking, task scheduling: non-pre-emptive and pre-emptive scheduling; task communication shared memory, message passing, Remote Procedure Call and Sockets, Task Synchronization: Task Communication/ Synchronization Issues, Task Synchronization Techniques

Text books:

1. Introduction to Embedded Systems - Shibu KV, Mc Graw Hill Education.
2. Computers as Components –Wayne Wolf, Morgan Kaufmann (second edition).

References:

1. Embedded System Design -Frank Vahid, Tony Grivargis, john Wiley.
2. Embedded Systems- An integrated approach - Lyla b das, Pearson education 2012.
3. Embedded Systems – Raj Kamal, TMH

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**SOFTWARE PROJECT MANAGEMENT**
Category:	**Professional Elective-III**
Couse Code:	23A056ET
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

This course is designed to enable the students to understand the fundamental principles of Software Project management & will also have a good knowledge of the responsibilities of a project manager and how to handle them.

Course Outcomes:

After completion of the course, students will be able to

1. Describe the fundamentals of Project Management
2. Recognize and use Project Scheduling Techniques
3. Familiarize with Project Control Mechanisms
4. Understand Team Management
5. Recognize the importance of Project Documentation and Evaluation

Unit 1 9

Conventional Software Management: The water fall model, conventional software Management performance Evolution of Software Economics: Software Economics. Pragmatic Software Cost Estimation Improving Software Economics: Reducing Software Product Size, Improving Software Processes, Improving Team Effectiveness, Improving Automation, Achieving Required Quality, Peer Inspections.

Unit 2 **9**

The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

Lifecycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.

Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts

Unit 3 **9**

Workflows of the process: Software process workflows, Inter Trans workflows. Check points of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments. Iterative Process Planning: work break down structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning

Unit 4 9

Process Automation: Automation Building Blocks, The Project Environment.

Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators Tailoring the Process: Process discriminants. Managing people and organizing teams.

Unit 5 9

Project Organizations and Responsibilities: Line - of-Business Organizations, Project Organizations, evolution of Organizations.

Future Software Project Management: modern Project Profiles, Next generation Software economics, modern process transitions.

Case Study: The Command Center Processing and Display System-Replacement(CCPDS-R)

Text books:

1. Software Project Management, Walker Royce,Pearson Education,2012
2. BobHughes,MikeCotterellandRajibMall–SoftwareProjectManagement‖,6thEdition, Mc Graw Hill Edition, 2017

Reference Books:

1. PankajJalote,–SoftwareProjectManagementinpractice‖,5thEdition,PearsonEducation, 2017.
2. Murali K.Chemuturi,Thomas M.Cagley Jr.‖Mastering Software Project Management: Best Practices, Tools and Techniques‖, J.Ross Publishing, 2010
3. SanjayMohapatra,–SoftwareProjectManagement‖,CengageLearning,2011

Online Learning Resources:

http://nptel.ac.in/courses/106101061/29

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**MOBILE ADHOC NETWORKS**
Category:	**Professional Elective-III**
Couse Code:	23A056FT
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. Knowledge of mobile ad hoc networks, design and implementation issues, and available solutions.
2. Knowledge of routing mechanisms and the three classes of approaches: proactive, on-demand, and hybrid.
3. Knowledge of clustering mechanisms and the different schemes that have been employed, e.g., hierarchical, flat, and leaderless
4. Knowledge of the 802.11 Wireless Lan (WiFi) and Bluetooth standards.

Course Outcomes:

1. Describe the unique issues in ad-hoc/sensor networks.
2. Describe current technology trends for the implementation and deployment of wireless ad- hoc/sensor networks.
3. Discuss the challenges in designing MAC, routing and transport protocols for wireless ad- hoc/sensor networks.
4. Discuss the challenges in designing routing and transport protocols for wireless Adhoc/sensor networks.
5. Comprehend the various sensor network Platforms, tools and applications

Unit 1

Introduction to Ad Hoc Networks:

Characteristics of MANETs, Applications of MANETs and challenges of MANETs -Routing in MANETs: Criteria for classification, Taxonomy of MANET routing algorithms, Topology based routing algorithms, Position based routing algorithms, Other routing algorithms.

Unit 2

Data Transmission

Broadcast storm problem, Broadcasting, Multicasting and Geocasting -TCP over Ad Hoc: TCP protocol overview, TCP and MANETs, Solutions for TCP over Ad hoc

Unit 3

Basics of Wireless, Sensors and Applications:

Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer.

Unit 4

Data Retrieval in Sensor Networks:

Routing layer, Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs, Sensor Networks and mobile robots-Security: Security in Adhoc networks, Key management, Secure routing, Cooperation in MANETs, Intrusion Detection systems.

Unit 5

Sensor Network Platforms and Tools: Sensor Network Hardware, Berkeley motes, Sensor Network Programming Challenges, Node-Level Software Platforms -Operating System: Tiny OS -Imperative Language: nesC, Data flow style language: Tiny GALS, Node Level Simulators, ns-2 and its sensor network extension.

TEXT BOOKS:

1. Ad Hoc and Sensor Networks –Theory and Applications, Carlos Corderio Dharma P. Aggarwal,World Scientific Publications, March 2006,ISBN –981-256-681-3
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science, ISBN –978-1-55860-914-3 (Morgan Kauffman)

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**NATURAL LANGUAGE PROCESSING**
Category:	**Professional Elective-III**
Couse Code:	23A056GT
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. Explain and apply fundamental algorithms and techniques in the area of natural language processing(NLP)
2. Discuss approaches to syntax and semantics in NLP.
3. Examine current methods for statistical approaches to machine translation.
4. Teach machine learning techniques used in NLP.

Course Outcomes:

After completion of the course, students will be able to

1.Understand the various NLP Applications and Organization of Natural language, able to learn and implement realistic applications using Python.

2.Apply the various Parsing techniques, Bayes Rule, Shannon game, Entropy and Cross Entropy.

3.Understand the fundamentals of CFG and parsers and mechanisms in ATN‘s.

4.Apply Semantic Interpretation and Language Modelling.

5.Apply the concept of Machine Translation and multilingual Information Retrieval systems and Automatic Summarization.

Unit 1 Introduction to Natural Language

The Study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different Levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems, Linguistic Background: An outline of English Syntax.

Unit 2 Grammars and Parsing

Grammars and Parsing – Top – Down and Bottom-Up Parsers, Transition Network Grammars, Feature Systems and Augmented Grammars, Morphological Analysis and the Lexicon, Parsing

with Features, Augmented Transition Networks, Bayes Rule, Shannon game, Entropy and Cross Entropy.

Unit 3 Grammars for Natural Language

Grammars for Natural Language, Movement Phenomenon in Language, Gap Threading, Human Preferences in Parsing, Shift Reduce Parsers, Deterministic Parsers.

Unit 4

Semantic Interpretation: Semantic & Logical form, Word senses & ambiguity, The basic logical form language, Encoding ambiguity in the logical Form, Verbs & States in logical form, The maticroles, Speech acts & embedded sentences, Defining semantics structure model theory

Language Modelling: Introduction, n- Gram Models, Language model Evaluation, Parameter Estimation, Language Model Adaption, Types of Language Models, Language-Specific Modelling Problems, Multilingual and Cross lingual Language Modelling

Unit 5

Machine Translation

Survey: Introduction, Problems of Machine Translation, Is Machine Translation Possible, Brief History, Possible Approaches, Current Status . Anusarakaor Language Accessor: Background, Cutting the GordianKnot, The Problem, Structure of Anusaraka System, User Interface, Linguistic Area, Giving up Agreement in Anusarsaka Output, Language Bridges.

Multilingual Information Retrieval

Introduction, Document Pre-processing, Monolingual Information Retrieval, CLIR, MLIR, Evaluation in Information Retrieval, Tools, Software and Resources.

Multilingual Automatic Summarization

Introduction, Approaches to Summarization, Evaluation, How to Build a Summarizer, Competitions and Datasets.

Textbooks:

1. James Allen, Natural Language Understanding, 2nd Edition, 2003, Pearson Education.
2. Multilingual Natural Language Processing Applications: From Theory To Practice- Daniel M.Bikel and ImedZitouni, Pearson Publications.
3. Natural Language Processing, A paninian perspective, Akshar Bharathi, Vineetchaitanya,Prentice –Hall of India.

Reference Books:

1. Charniack, Eugene, Statistical Language Learning, MITPress,1993.
2. Jurafsky, Dan and Martin, James, Speech and Language Processing, 2nd Edition, Prentice Hall,2008.
3. Manning, Christopher and Hen rich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press,1999.

Online Learning Resources:

https://nptel.ac.in/courses/106/105/106105158

http://www.nptelvideos.in/2012/11/natural-language- processing.html

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**DISTRIBUTED OPERATING SYSTEM**
Category:	**Professional Elective-III**
Couse Code:	23A056HT
Year:	III B. Tech
Semester:	II Semester
Branch:	CSE

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

1. To study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems)
2. Hardware and software features that support these systems.

Course Outcomes:

1. Understand the design approaches of advanced operating systems
2. Analyze the design issues of distributed operating systems
3. Evaluate design issues of multi processor operating systems.
4. Identify the requirements Distributed File System and Distributed Shared Memory.
5. Formulate the solutions to schedule the real time applications.

Unit 1

Architectures of Distributed Systems: System Architecture Types, Distributed Operating Systems, Issues in Distributed Operating Systems, Communication Primitives. Theoretical Foundations: Inherent Limitations of a Distributed System, Lam port‘s Logical Clocks, Vector Clocks, Causal Ordering of Messages, Termination Detection.

Unit 2

Distributed Mutual Exclusion: The Classification of Mutual Exclusion Algorithms, Non-Token –Based Algorithms: Lamport‘s Algorithm, The Ricart-Agrawalsa Algorithm, Maekawa‘s Algorithm,Token- Based Algorithms: Suzuki-Kasami‘s Broadcast Algorithm, Singhal‘s Heurisric Algorithm, Raymond‘s Heuristic Algorithm.

Unit 3

Distributed Deadlock Detection: Preliminaries, Deadlock Handling Strategies in Distributed Systems, Issues in Deadlock Detection and Resolution, Control Organizations for Distributed Deadlock Detection, Centralized- Deadlock – Detection Algorithms, Distributed Deadlock Detection Algorithms, Hierarchical Deadlock Detection Algorithms

Unit 4

Multiprocessor System Architectures: Introduction, Motivation for multiprocessor Systems, Basic Multiprocessor System Architectures Multi Processor Operating Systems: Introduction, Structures of Multiprocessor Operating Systems, Operating Design Issues, Threads, Process

Synchronization, Processor Scheduling. Distributed File Systems: Architecture, Mechanisms for Building Distributed File Systems, Design Issues

Unit 5

Distributed Scheduling: Issues in Load Distributing, Components of a Load Distributed Algorithm, Stability, Load Distributing Algorithms, Requirements for Load Distributing, Task Migration, Issues in task Migration Distributed Shared Memory: Architecture and Motivation, Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues

TEXT BOOK:

1. Advanced Concepts in Operating Systems, Mukesh Singhal, Niranjan G. Shivaratri, Tata Mc Graw- Hill Edition 2001

REFERENCE BOOK:

1. Distributed Systems: Andrew S. Tanenbaum, Maarten Van Steen, Pearson Prentice Hall, Edition – 2, 2007.

Title of the Course: Machine Learning Lab
Category: Professional Core Lab
Couse Code: 23A0561L
Year: III B. Tech
Semester: II Semester
Branch: **CSE**

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	3	1.5

Course Objectives:

1. Make use of Data sets in implementing the machine learning algorithms
2. Implement the machine learning concepts and algorithms in any suitable language of choice.

Course Outcomes:

After completion of the course, students will be able to

1. Understand the Mathematical and statistical prospectives of machine learning algorithms through python programming
2. Appreciate the importance of visualization in the data analytics solution.
3. Derive insights using Machine learning algorithms

List of Experiments:

Note:

a. The programs can be implemented in either JAVA or Python.

b. For Problems 1 to 6 and 10, programs are to be developed without using the built-in classes or APIs of Java/Python.

c. Data sets can be taken from standard repositories (https://archive.ics.uci.edu/ml/datasets.html) or constructed by the students.

Exercise 1

Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

Exercise 2

For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

Exercise 3

Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Exercise 4

Build an Artificial Neural Network by implementing the Back-propagation algorithm and test the same using appropriate data sets.

Exercise 5

Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

Exercise 6

Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

Exercise 7

Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.

Exercise 8

Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

Exercise 9

Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

Exercise 10

Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs..

Projects

1. Predicting the Sale price of a house using Linear regression
2. Spam classification using Naïve Bayes algorithm
3. Predict car sale prices using Artificial Neural Networks
4. Predict Stock market trends using LSTM
5. Detecting faces from images

References:

1.Python Machine Learning Workbook for beginners, AI Publishing, 2020.

Online Learning Resources/ Virtual Labs:

1) Machine Learning A-Z (Python & R in Data Science Course) | Udemy
2) Machine Learning | Coursera

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES
(An Autonomous Institution)

Title of the Course:	**CRYPTOGRAPHY & NETWORK SECURITY LAB**
Category:	Professional Core Lab
Couse Code:	23A0563L
Year:	III B. Tech
Semester:	II Semester
Branch:	**CSE, CSE(AI), CSE(AI&ML), CSE(DS),AI&DS**

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	3	1.5

List of Experiments:

Exercise 1
Write a C program that contains a string (char pointer) with a value Hello world‘. The program should XOR each character in this string with 0 and displays the result.

Exercise 2
Write a C program that contains a string (char pointer) with a value Hello world‘. The program should AND or and XOR each character in this string with 127 and display the result.

Exercise 3
Write a Java program to perform encryption and decryption using the following algorithms
a. Ceaser cipher b. Substitution cipher c. Hill Cipher

Exercise 4
Write a C/JAVA program to implement the DES algorithm logic.

Exercise 5
Write a C/JAVA program to implement the Blowfish algorithm logic.

Exercise 6
Write a C/JAVA program to implement the Rijndael algorithm logic.

Exercise 7
Write the RC4 logic in Java Using Java cryptography; encrypt the text ―Hello world‖ using Blowfish. Create your own key using Java key tool.

Exercise 8
Write a Java program to implement RSA algorithm.

Exercise 9
Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.

Exercise 10
Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

Exercise 11
Calculate the message digest of a text using the MD5 algorithm in JAVA.