JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

Vision and Mission of the University

Vision

The University is primarily promoting quality of education in the areas of Science, Technology, Engineering and Mathematics (STEM) as four academic pillars of education, to excel in teaching, learning, research, consultancy and placements through innovative practices with global perspective.

Mission

1. Design an Industry relevant curriculum from time to time with a Global perspective
2. Promoting quality education by embracing ICT delivery mechanism with continuous pedagogy through e-learning mechanism
3. Spread across for industry collaborations with a focus to pre-training and placements for technology transfer to society
4. Establishing centers of excellence to promote research and innovations in multidisciplinary areas to bring in patent culture and consultancy practices
5. International Collaborations for student outreach
6. Facilitating international students to study in JNTUK to infuse cross culture learning practices.

Vision and Mission of the Institute

Vision and Mission of the Department

Programme Education Objectives (PEOs)of the M.Tech (SE)

PEO 1: To develop students with depth knowledge of computer Science, Computer Applications, Information Technology and Computer Science and Engineering which provide a strong foundation to pursue career in education and software industry for innovation, research and development.

PEO 2: To excel in career involving higher order and challenging tasks and try to become a part of success and growth and work in collaboration with all organization

PEO 3: To develop the leadership qualities, to lead and work in a team in professional environment, demonstrate professional integrity and feel responsibility towards country at an appropriate level in order to address the issues in a responsive, ethical and innovative manner

Mapping of Mission statements to PEOs

Programme Outcomes (POs)

PO1: An ability to independently carry out research /investigation and development work to solve practical problems

PO2: An ability to write and present a substantial technical report/document

PO3: Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: Use modern programming environments, testing tools, DevOps practices, and software project management tools for the effective development and deployment of software applications.

PO5: Incorporate ethical standards, software quality assurance practices, and risk management techniques while ensuring data privacy, security, and regulatory compliance.

PO6: Exhibit leadership and communication skills in multidisciplinary teams, and engage in continuous learning to adapt to emerging software technologies and industry demands.

Mapping of Programme Outcomes to PEOs

Mapping of Programme Outcomes to GAs

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

M.Tech

SOFTWARE ENGINEERING

Programme Course Structure & Syllabus

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

R25 M.TECH PROGRAMME STRUCTURE

I Semester

S. No.	Course Title	L	T	P	C
1	Advanced Algorithms Analysis	3	1	0	4
2	Advanced Object Oriented Software Engineering	3	1	0	4
3	Artificial Intelligence	3	1	0	4
4	**Program Elective – I**	3	0	0	3
5	**Program Elective –II**	3	0	0	3
6	Advanced Algorithms Analysis Lab	0	1	2	2
7	Advanced Object Oriented Software Engineering lab	0	1	2	2
8	**Seminar-I**	0	0	2	1
	TOTAL	**15**	**5**	**6**	**23**

List of Professional Elective Courses in I Semester (Electives I & II)

S.No.	Course Title
1	Secure coding
2	Cryptography and Network Security
3	High performance computing
4	Computer Vision
5	Software Project Management
6	Knowledge representation and reasoning
7	Software Reliability And Quality Management
8	Natural Language Processing

@Minimum2/3themesper elective

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

II Semester

S. No.	Course Title	L	T	P	C
1	Full Stack Technologies	3	1	0	4
2	Machine Learning	3	1	0	4
3	Cloud Computing	3	1	0	4
4	**Program Elective –III**	3	0	0	3
5	**Program Elective –IV**	3	0	0	3
6	Full Stack Technologies Lab	0	1	2	2
7	Machine Learning Lab	0	1	2	2
8	**Seminar–II**	0	0	2	1
	TOTAL	**15**	**5**	**6**	**23**

List of Professional Elective Courses in II Semester (Electives III&IV)

S.No.	Course Title
1	Design Patterns
2	Agile Methodologies
3	Software Design Methodologies
4	Software Architecture & Design Patterns
5	Software Quality Engineering
6	Social Media Analytics
7	Software Requirements and Estimation
8	Block Chain Technologies

@Minimum2/3themesper elective

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

III Semester

Sl. No.	Course Title	L	T	P	C
1	Research Methodology and IPR / *Swayam 12 week MOOC course – RM&IPR*	3	0	0	3
2	Summer Internship/ Industrial Training (8-10 weeks)*	-	-	-	3
3	Comprehensive Viva#	-	-	-	2
4	Dissertation Part – A$	-	-	20	10
	TOTAL	**3**	**-**	**20**	**18**

* Student attended during summer / year break and assessment will be done in 3rd Sem.

Comprehensive viva can be conducted courses completed up to second sem.

$ Dissertation – Part A, internal assessment

IV Semester

Sl. No.	Course Title	L	T	P	C
1	Dissertation Part – B%	-	-	32	16
	TOTAL	**-**	**-**	**32**	**16**

% External Assessment

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

I Semester	ADVANCED ALGORITHMS ANALYSIS	L	T	P	C
		3	1	0	4

Pre-Requisites: UG level course in Algorithm Design and Analysis

Course Objectives:

1. Introduce students to the advanced methods of designing and analyzing algorithms.
2. The student should be able to choose appropriate algorithms and use it for a specific problem.
3. To familiarize students with basic paradigms and data structures used to solve advanced algorithmic problems.
4. Students should be able to understand different classes of problems concerning their computation difficulties.
5. To introduce the students to recent developments in the area of algorithmic design.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Analyze the complexity/performance of different algorithms.	K4
CO2	Determine the appropriate data structure for solving a particular set of problems.	K3
CO3	Categorize the different problems in various classes according to their complexity	K4
CO4	Discuss Dynamic Programming and Fast Fourier Transform algorithm	K4
CO5	Explain Linear Programming	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		3			
CO2	3		3	2		
CO3	3	2	3			
CO4	3	2	3			
CO5	3		2		2	

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Sorting:** Review of various sorting algorithms, topological sorting **Graph:** Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.	10Hrs

UNIT – 2	**Matroids:** Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST. **Graph Matching:** Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.	12Hrs
UNIT – 3	**Flow-Networks:** Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. **Matrix Computations:** Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP decomposition.	12Hrs
UNIT – 4	**Shortest Path in Graphs:** Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. **Modulo Representation of integers/polynomials:** Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem. **Discrete Fourier Transform (DFT):** In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm	12Hrs
UNIT – 5	**Linear Programming:** Geometry of the feasibility region and Simplex algorithm **NP-completeness:** Examples, proof of NP-hardness and NP-completeness. Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.	12Hrs
	Total	58Hrs

Text Books:

1. Cormen, Leiserson, Rivest, Stein, "Introduction to Algorithms".
2. Aho, Hopcroft, Ullman "The Design and Analysis of Computer Algorithms".
3. Kleinberg and Tardos."Algorithm Design".
4. Data Structures: A Pseudocode Approach with C, 2nd Edition, Richard F.Gilberg, Behrouz A. Forouzon, Cengage Learning, 2004
5. Data Structures, Algorithms and Applications in java, 2 nd Edition, SartajSahni, University Press/Orient BlackSwan, 2005

Reference Books:

1. Data Structures And Algorithm Analysis, 2 nd Edition, Mark Allen Weiss, Pearson, 2002
2. Data Structures And Algorithms in C++, 3 rd Edition, Adam Drozdek, Cengage Learning, 2005
3. C and Data Structures: A Snap Shot Oriented Treatise Using Live Engineering Examples, 1 st Edition, N.B.Venkateswarulu, E.V. Prasad, S Chand & Co, 2009
4. Classic Data Structures, 2 nd Edition, Debasis Samantha, PHI Learning, 2009

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING COURSE STRUCTURE AND SYLLABUS

I Semester	ADVANCED OBJECT ORIENTED SOFTWARE ENGINEERING	L	T	P	C
		3	1	0	4

Pre-requisite:

Course Objectives:

To introduce Object-oriented software engineering (OOSE) - which is a popular technical approach to analyzing, designing an application, system, or business by applying the object- oriented paradigm and visual modeling.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

Upon successful completion of the course, a student will be able to:

CO	Course Outcomes	Knowledge Level (K)#
CO1	Understand and apply the fundamental principles of Object-Oriented Programming (OOP) concepts and Unified Modeling Language (UML) basics, in the development of software solutions.	K3
CO2	Analyze and specify software requirements, develop use cases and scenarios, apply object- oriented analysis and design (OOAD) principles	K4
CO3	Experiment with the concept of test-driven development (TDD) and its practical implementation	K3
CO4	Analyze and Evaluate Software Maintenance and Evolution Strategies	K4
CO5	Apply Advanced Object-Oriented Software Engineering Concepts	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2		
CO2	3	3	3	2		
CO3	2	2	2	3	2	2
CO4	2	2	3	2	3	2
CO5	3	2	3	3	2	3

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	Introduction to Object-Oriented Programming: Overview of software engineering, Introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, Introduction to software development process and software development life cycle (SDLC).	10Hrs
UNIT – 2	Requirements Analysis and Design: Requirements analysis and specification, Use cases and scenarios, Object-oriented analysis and design (OOAD), Design patterns, UML modelling techniques (class diagrams, sequence diagrams, state	10Hrs

	machine diagrams, activity diagrams)	
UNIT – 3	Software Construction and Testing: Software construction basics, Object-oriented design principles, Object-oriented programming languages (Java, C++, Python), Software testing basics (unit testing, integration testing, system testing), Test-driven development (TDD)	12Hrs
UNIT – 4	Software Maintenance and Evolution: Software maintenance basics, refactoring techniques Software version control, Code review and inspection, Software evolution and reengineering	10Hrs
UNIT – 5	Advanced Topics in Object-Oriented Software Engineering: Model-driven engineering (MDE), Aspect-oriented programming (AOP), Component-based software engineering (CBSE), Service- oriented architecture (SOA), Agile software development and Scrum methodologies.	12Hrs
	Total	54Hrs

Text Books:

1. An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press

Reference Books:

1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville, 1998, Wiley
2. Design Patterns, E.Gamma, R. Helm, R. Johnson, and J. Vlissides
3. The Unified Modeling Language Reference Manual, J. Rumbaugh, I.Jacobson and G. Booch, Addison Wesley

Online Learning Resources: (Pls include hyperlinks related to NPTEL/Vlabs/IITB Spoken Tutorial etc.,)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	ARTIFICIAL INTELLIEGCE	L	T	P	C
		3	1	0	4

Pre-requisite:

Course Objectives:

1. Gain a historical perspective of Artificial Intelligence (AI) and its foundations.
2. Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
3. Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
4. Experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool. Experiment with a machine learning model for simulation and analysis.
5. Explore the current scope, potential, limitations, and implications of intelligent systems.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.	K2
CO2	Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.	K4
CO3	Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing.	K5
CO4	Determine the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.	K5
CO5	Solve problems with uncertain information using Bayesian approaches.	K3

#Based on suggested Revised BTL

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2		
CO2	3	2	3	2		
CO3	3	2	3	3	2	
CO4	3	2	3	3	2	
CO5	3	2	3	2	2	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Introduction to artificial intelligence:** Introduction , history, intelligent systems, foundations of AI, applications, tic-tac-tie game playing, development of AI languages, current trends in AI **Problem solving: state-space search and control strategies:** Introduction, general problem solving, characteristics of problem, exhaustive searches, heuristic search techniques, iterative-deepening a*, constraint satisfaction	10Hrs
UNIT – 2	**Problem reduction and game playing:** Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games **Logic concepts:** Introduction, propositional calculus, proportional logic, natural deduction system, axiomatic system, semantic tableau system in proportional logic, resolution refutation in proportional logic, predicate logic	10Hrs
UNIT – 3	**Knowledge representation:** Introduction, approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, knowledge representation using frames **Advanced knowledge representation techniques:** Introduction, conceptual dependency theory, script structure, cyc theory, case grammars, semantic web.	12Hrs
UNIT – 4	**Uncertainty measure: probability theory:** Introduction, probability theory, Bayesian belief networks, certainty factor theory, dempster-shafer theory, non-monotonic reasoning, TMS.	12Hrs
UNIT – 5	**Fuzzy sets and fuzzy logic:** Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.	12Hrs
	Total	56Hrs

Text Books:

1. Artificial intelligence, A modern Approach, 2nded, Stuart Russel, Peter Norvig, Prentice Hall
2. Artificial Intelligence, Saroj Kaushik, 1st Edition, CENGAGE Learning, 2011.

Reference Books:

1. Artificial intelligence, structures and Strategies for Complex problem solving, 5th Edition, George F Lugar, PEA
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang, Springer, 2017
3. Artificial Intelligence, A new Synthesis, 1st Edition, Nils J Nilsson, Elsevier, 1998
4. Artificial Intelligence- 3rd Edition, Rich, Kevin Knight, Shiv Shankar B Nair, TMH
5. Introduction To Artificial Intelligence And Expert Systems, 1st Edition, Patterson, Pearson India, 2015

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	SECURE CODING	L	T	P	C
		0	1	2	2

Course Objectives:

- Understanding of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities.
- Knowledge of outline of the techniques for developing a secure application.
- Recognize opportunities to apply secure coding principles

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Outline the secure systems and various security attacks	K3
CO2	Demonstrate the development of process of software leads to secure coding practices	K5
CO3	Apply Secure programs and various risk in the software's	K5
CO4	Classify various errors that lead to vulnerabilities	K6
CO5	Design Real time software and vulnerabilities	K6

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1						
CO2						
CO3						
CO4						
CO5						
CO6						

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Introduction-**Need for secure systems, Proactive security development process, Security principles to live by and threat modelling.	10Hrs
UNIT – 2	**Secure Coding in C-** Character strings- String manipulation errors, String Vulnerabilities and exploits Mitigation strategies for strings,	10Hrs

	Pointers, Mitigation strategies in pointer based vulnerabilities Buffer Overflow based vulnerabilities	
UNIT – 3	**Secure Coding in C++ and Java-** Dynamic memory management, Common errors in dynamic memory management, Memory managers, Double –free vulnerabilities, Integersecurity, Mitigation strategies	12Hrs
UNIT – 4	**Database and Web Specific Input Issues**- Quoting the Input, Use of stored procedures, Building SQL statements securely, XSS related attacks and remedies	12Hrs
UNIT – 5	**Software Security Engineering-** Requirements engineering for secure software: Misuse and abuse cases, SQUARE process model Software security practices and knowledge for architecture and design	12Hrs
	Total	56Hrs

Text Book:

1. Writing Secure Code, 2nd Edition, Michael Howard, David LeBlanc, Microsoft Press, 2003

Reference Books:

1. Secure Coding in C and C++, Robert C. Sea cord, 2nd edition, Pearson Education, 2013
2. Software Security Engineering: A guide for Project Managers, 1st ed, Julia H. Allen, Sean J.Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, Addison-Wesley Professional,2008

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	CRYPTOGRAPHY & NETWORK SECURITY (PROGRAM ELECTIVE-I)	L	T	P	C
		3	0	0	3

Course Objectives: This course is aimed at enabling the students to

- Explain the objectives of information security
- Explain the importance and application of each of confidentiality, integrity, authentication and availability
- Understand the basic categories of threats to computers and networks
- Discusses the Mathematics of Cryptography
- Discuss the fundamental ideas of Symmetric and Asymmetric cryptographic Algorithms
- Discusses the Network layer, Transport Layer and Application layer Protocols Enhanced security mechanisms

CO	Course Outcomes	Knowledge Level (K)#
CO1	Student will be able to understand security issues related to computer networks and learn different symmetric key techniques	K2
CO2	Students will be able learn mathematic of cryptography for symmetric and Asymmetric algorithms and apply this knowledge to understand the Cryptographic algorithms	K3
CO3	Students will be able learn different types of symmetric and Asymmetric algorithms	K3
CO4	Students will be able learn different algorithms of Hash functions, message authentication and digital signature and their importance to the security	K4
CO5	Students will be able learn different Enhanced security protocols of Application Layer, Transport Layer and Network layer	K4

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3		2	
CO2	3	2	3	2	3	
CO3	3	2	3	3	3	
CO4	2	2	3		2	
CO5	3	2	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Security Concepts:** Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security Cryptography.Classical Encryption Techniques-symmetric cipher model, Substitution techniques, Transposition techniques, Rotor Machines, Stegnography.	10Hrs
UNIT – 2	**Introduction to Symmetric Cryptography: Algebraic Structures-** Groups, Rings, Fields, GF(2^n) fields, Polynomials.**Mathematics of Asymmetric cryptography:** Primes, Checking For Primness, Eulers phi-functions, Fermat's Little Theorem, Euler's Theorem, Generating Primes, Primality Testing, Factorization, Chinese Remainder Theorem, Quadratic Congruence, Exponentiation And Logarithm..	12Hrs
UNIT – 3	**Symmetric key Ciphers:** Block Cipher principles, DES, AES, Blowfish, IDEA, Block cipher operation, Stream ciphers: RC4, RC5 **Asymmetric key Ciphers:** Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman Key Exchange, Elgamal Cryptographic system, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.	12Hrs
UNIT – 4	**Cryptographic Hash Functions:** Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithms (SHA) **Message Authentication Codes:** Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MAC'S,MAC'S Based On Hash Functions: HMAC, MAC'S Based On Block Ciphers: DAA And CMAC **Digital Signatures:** Digital Signatures, Elgamal Digital Signature Scheme, Elliptic Curve Digital Signature Algorithm, RSA-PSS Digital Signature Algorithm.	12Hrs
UNIT – 5	**Network and Internet Security: Transport-Level Security:** Web Security Considerations, Transport Level Security, HTTPS, SSH. **IP Security:** IP Security Overview, IP Security Policy, Encapsulating Security Payload, Authentication Header Protocol. **Electronic-Mail Security:** Internet-mail Security, Email Format, Email Threats and Comprehensive Email Security, S/MIME, PGP.	12Hrs
	Total	58Hrs

Text Books:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 7th Edition, 2017
2. Cryptography and Network Security: Behrouz A. ForouzanDebdeep, Mc Graw Hill, 3rd Edition, 2015

Reference Books:

1. Cryptography and Network Security: AtulKahate, Mc Graw Hill, 3rd Edition
2. Introduction to Cryptography with Coding Theory: Wade Trappe, Lawrence C. Washington, Pearson.
3. Modern Cryptography: Theory and Practice ByWenbo Mao. Pearson

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	HIGH PERFORMANCE COMPUTING (PROGRAM ELECTIVE-I)	L	T	P	C
		3	0	0	3

Course Objectives: This course is aimed at enabling the students to

- The main objectives of the course is to study parallel computing hardware and programming models, performance analysis and modeling of parallel programs

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Describe different parallel architectures, inter-connect networks, programming Models	**K3**
CO2	Develop an efficient parallel algorithm to solve given problem	**K4**
CO3	Analyze and measure performance of modern parallel computing systems	**K5**
CO4	Build the logic to parallelize the programming task	**K2**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2		
CO2	3	2	3	3		
CO3	3	2	3	2	2	
CO4	3	2	3	3	2	1

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Introduction:** Motivating Parallelism, Scope of Parallel Computing, Parallel Programming Platforms: Implicit Parallelism, Trends in Microprocessor and Architectures, Limitations of Memory, System Performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines, Scalable design principles, Architectures: N-wide superscalar architectures, Multi-core architecture.	10Hrs

UNIT – 2	**Parallel Programming:** Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models, The Age of Parallel Processing, the Rise of GPU Computing, A Brief History of GPUs, Early GPU.	12Hrs
UNIT – 3	**Basic Communication:** Operations- One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operations. Programming shared address space platforms: threads- basics, synchronization, Open MP programming	12Hrs
UNIT – 4	**Analytical Models:** Sources of overhead in Parallel Programs, Performance Metrics for Parallel Systems, and The effect of Granularity on Performance, Scalability of Parallel Systems, Minimum execution time and minimum cost, optimal execution time. Dense Matrix Algorithms: MatrixVector Multiplication, Matrix-Matrix Multiplication.	12Hrs
UNIT – 5	**Parallel Algorithms- Sorting and Graph:** Issues in Sorting on Parallel Computers, Bubble Sort and its Variants, Parallelizing Quick sort, All-Pairs Shortest Paths, Algorithm for sparse graph, Parallel Depth-First Search, Parallel Best First Search. CUDA Architecture : CUDA Architecture, Using the CUDA Architecture, Applications of CUDA Introduction to CUDA C-Write and launch CUDA C kernels, Manage GPU memory, Manage communication and synchronization, Parallel programming in CUDA- C.	12Hrs
	Total	58Hrs

Text Books:

1. AnanthGrama, Anshul Gupta, George Karypis, and Vipin Kumar, "Introduction to Parallel Computing", 2nd edition, Addison-Wesley, 2003, ISBN: 0-201-64865-2
2. Jason sanders, Edward Kandrot, “CUDA by Example”, Addison-Wesley, ISBN-13: 978-0-13-138768-3

Reference Books:

1. Kai Hwang, ”Scalable Parallel Computing”, McGraw Hill 1998, ISBN:0070317984
2. Shane Cook, “CUDA Programming: A Developer's Guide to Parallel Computing with GPUs”, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 2013

ISBN: 9780124159884

3. David Culler Jaswinder Pal Singh, ”Parallel Computer Architecture: A Hardware/ Software Approach”, Morgan Kaufmann,1999, ISBN 978-1-55860-343-1
4. Rod Stephens, “Essential Algorithms”, Wiley, ISBN: ISBN: 978-1-118-61210-1

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	COMPUTER VISION (PROGRAM ELECTIVE-II)	L	T	P	C
		3	0	0	3

Course Objectives: This course is aimed at enabling the students to

- To understand the Fundamental Concepts related to sources, shadows and shading
- To understand the Geometry of Multiple Views

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Demonstrate different parallel architectures, inter-connect networks, programming models	K2
CO2	Develop an efficient parallel algorithm to solve given problem	K3
CO3	Analyze and measure performance of modern parallel computing systems	K4
CO4	Build the logic to parallelize the programming task	K6

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2		
CO2	3	2	3	3		
CO3	3	2	3	2	2	
CO4	3	2	3	3	2	1

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**CAMERAS :** Pinhole Cameras Radiometry–Measuring Light: Light in Space, Light Surfaces ,Important Special Cases Sources, Shadows, And Shading: Qualitative Radiometry, Sources and Their Effects, Local Shading Models, Application:PhotometricStereo,Interreflections:GlobalShading ModelsColor: The Physics of Color, Human Color Perception, Representing Color, A Model for Image Color, Surface Color from Image Color.	10Hrs

UNIT – 2	**Linear Filters:** Linear Filters and Convolution, Shift Invariant Linear Systems, Spatial Frequency and Fourier Transforms, Sampling and Aliasing, Filters as Templates, Edge Detection: Noise, Estimating Derivatives, Detecting Edges Texture0:Representing Texture, Analysis (and Synthesis) Using Oriented Pyramids, Application: Synthesis by Sampling Local Models, Shape fromTexture.	12Hrs
UNIT – 3	**The Geometry of Multiple Views:** Two Views Stereopsis: Reconstruction, Human Stereopsis, Binocular Fusion, Using More Cameras Segmentation by Clustering: What Is Segmentation? Human Vision: Grouping and Getstalt, Applications: Shot Boundary Detection and Background Subtraction, Image Segmentation by Clustering Pixels, Segmentation by Graph-Theoretic Clustering,	12Hrs
UNIT – 4	**Segmentation by Fitting a Model:** The Hough Transform, Fitting Lines, Fitting Curves, Fitting as a Probabilistic Inference Problem, Robustness Segmentation and Fitting Using Probabilistic Methods: Missing Data Problems, Fitting, and Segmentation, The EM Algorithm in Practice, Tracking With Linear Dynamic Models: Tracking as an Abstract Inference Problem, Linear Dynamic Models , Kalman Filtering, Data Association, Applications and Examples	12Hrs
UNIT – 5	**Geometric Camera Models:** Elements of Analytical Euclidean Geometry, Camera Parameters and the Perspective Projection, Affine Cameras and Affine Projection Equations Geometric Camera Calibration: Least-Squares Parameter Estimation, A Linear Approach to Camera Calibration, Taking Radial Distortion into Account, Analytical Photogrammetry, **Case study:** Mobile Robot Localization Model- Based Vision: Initial Assumptions, Obtaining Hypotheses by Pose Consistency, Obtaining Hypotheses by pose Clustering, Obtaining Hypotheses Using Invariants, Verification, Case study: Registration In Medical Imaging Systems, Curved Surfaces and Alignment.	12Hrs
	Total	58Hrs

Text Books:

1. David A. Forsyth and Jean Ponce: Computer Vision – A Modern Approach, PHI Learning (Indian Edition), 2009.

Reference Books:

1. E. R. Davies: Computer and Machine Vision – Theory, Algorithms and Practicalities, Elsevier (Academic Press), 4th edition, 2013.
2. R. C. Gonzalez and R. E. Woods “Digital Image Processing” Addison Wesley 2008. 3. Richard Szeliski “Computer Vision: Algorithms and Applications” Springer-Verlag London Limited 2011.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	SOFTWARE PROJECT MANAGEMENT (PROGRAM ELECTIVE-II)	L	T	P	C
		3	0	0	3

Course Objectives: This course is aimed at enabling the students to

At the end of the course, the student shall be able to:

- To describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project
- To compare and differentiate organization structures and project structures
- To implement a project to manage project schedule, expenses and resources with the application of suitable project management tools

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Recall how old software methods worked, why they had problems, and how new ways like teamwork and automation make software better and faster.	K1
CO2	Demonstrate the different phases of software development and identify the key documents and outputs created in each phase, like plans, designs, and reports.	K2
CO3	Organize software architecture from both management and technical views. Use iterative planning to organize tasks, set milestones, and estimate time and cost.	K3
CO4	Determine how software project teams are organized and how automation and metrics help manage and improve the project and its quality.	K5
CO5	Choose the basics of Agile and DevOps, how to use Scrum, and understand how DevOps tools, people, and processes help in faster and better software delivery.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3		2	
CO2	2	3	3		2	
CO3	3	2	3	3	2	2
CO4	3	2	3	3	3	3
CO5	3	2	3	3	3	3

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Conventional Software Management:** The waterfall model, conventional software Management performance. **Evolution of Software Economics:** Software Economics, pragmatics of software cost estimation. **Improving Software Economics:** Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections. **The old way and the new:** The principles of conventional software Engineering, principles of modern software management, transitioning to aniterative process.	10Hrs
UNIT – 2	**Life cycle phases:** Engineering and production stages, inception, Elaboration, construction, transition phases. **Artifacts of the process:** The artifactsets, Management artifacts, Engineering artifacts, program aticartifacts.	12Hrs
UNIT – 3	**Model based software architectures:** A Management perspective and technical perspective. **Work Flows of the process:** Software processes work flows, Iteration work flows. **Check points of the process:** Major milestones, Minor Milestones, Periodic status assessments. **Iterative Process Planning:** Work break down structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.	12Hrs
UNIT – 4	**Project Organizations and Responsibilities:** Line-of Business Organizations, Project Organizations, evolution of Organizations. **Process Automation:** Automation Building blocks, The Project Environment. **Project Control and Process instrumentation:** These vencore Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.	12Hrs
UNIT – 5	Agile Methodology, ADAPTing to Scrum, Patterns for Adopting Scrum, Iterating towards Agility. **Fundamentals of DevOps**: Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps ecosystem. DevOps adoption in projects:	12Hrs

	Technology aspects, Agiling capabilities, Tool stack implementation, People aspect, processes	
	Total	58Hrs

Text Books:

1. Software Project Management, Walker Royce, PEA, 2005.
2. Succeeding with Agile: Software Development Using Scrum, Mike Cohn, Addison Wesley.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim, John Willis, Patrick Debois, Jez Humb, 1st Edition, O'Reilly publications, 2016.

Reference Books:

1. Software Project Management, Bob Hughes, 3/e, Mike Cotterell, TMH
2. Software Project Management, Joel Henry, PEA
3. Software Project Management in practice, Pankaj Jalote, PEA, 2005,
4. Effective Software Project Management, Robert K.Wysocki, Wiley, 2006.
5. Project Management in IT, Kathy Schwalbe, Cengage

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	KNOWLEDGE REPRESENTATION AND REASONING (PROGRAM ELECTIVE-II)	L	T	P	C
		3	0	0	3

Course Objectives:

- To investigate the key concepts of Knowledge Representation (KR) techniques and different notations.
- To integrate the KR view as a knowledge engineering approach to model organizational knowledge.
- To introduce the study of ontologies as a KR paradigm and applications of ontologies.
- To understand various KR techniques and process, knowledge acquisition and sharing of ontology.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Analyze and design knowledge-based systems intended for computer implementation.	K4
CO2	Demonstrate theoretical knowledge about principles for logic-based representation and reasoning.	K2
CO3	Make use of knowledge-engineering process	K3
CO4	Experiment with production systems, frames, inheritance systems and approaches to handle uncertain or incomplete knowledge.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2		
CO2	3	2	3			
CO3	3	2	3	2		
CO4	2	2	3	2	2	1

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**The Key Concepts:** Knowledge, Representation, Reasoning, Why knowledge representation and reasoning, Role of logic: Historical background, Representing knowledge in logic, Varieties of logic,	10Hrs

	Name, Type, Measures, Unity Amidst diversity	
UNIT – 2	**Ontology:** Ontological categories, Philosophical background, Top-level categories, Describing physical entities, Defining abstractions, Sets, Collections, Types and Categories, Space and Time	12Hrs
UNIT – 3	**Knowledge Representations:** Knowledge Engineering, Representing structure in frames, Rules and data, Object-oriented systems, Natural language Semantics, Levels of representation	12Hrs
UNIT – 4	**Processes:** Times, Events and Situations, Classification of processes, Procedures, Processes and Histories, Concurrent processes, Computation, Constraint satisfaction, Change Contexts: Syntax of contexts, Semantics of contexts, First-order reasoning in contexts, Modal reasoning in contexts, Encapsulating objects in contexts.	12Hrs
UNIT – 5	**Knowledge Soup:** Vagueness, Uncertainty, Randomness and Ignorance, Limitations of logic, Fuzzy logic, Nonmonotonic Logic, Theories, Models and the world, Semiotics Knowledge Acquisition and Sharing: Sharing Ontologies, Conceptual schema, Accommodating multiple paradigms, Relating different knowledge representations, Language patterns, Tools for knowledge acquisition	12Hrs
	Total	58Hrs

Text Books:

1. Knowledge Representation logical, Philosophical, and Computational Foundations by John F. Sowa, Thomson Learning.
2. Knowledge Representation and Reasoning by Ronald J. Brachman, Hector J. Levesque, Elsevier.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	SOFTWARE RELIABILITY AND QUALITY MANAGEMENT	L	T	P	C
		3	0	0	3

Pre-requisite: Software Engineering, Probability and Statistics

Course Objectives:

1.
2.
3.
4.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Demonstrate Software Reliability during different phases of Software Development Life Cycle	K2
CO2	Analyze Software Reliability parameters using Markovian Modeling.	K4
CO3	Estimate Software Reliability parameters using Maximum Likelihood and Least Square Method.	K5
CO4	Evaluate performance of Binomial-Type, Poison-Type and Markovian Models.	K5
CO5	Predict Software Reliability using Intelligent Techniques.	K6
CO6	Design Quality Attributes for Software Quality Assurance (SQA).	K6

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	3	3	
CO2	3		3			
CO3	3		3			
CO4	3		3			
CO5	3		3	2	2	
CO6	3	3	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours

UNIT – 1	**Introduction to Software Reliability:** The need for Software Reliability, Some Basic Concepts, Software Reliability and Hardware Reliability, Availability, Modelling and General Model Characteristics.	10Hrs.
UNIT – 2	**Software Reliability Modeling:** Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric, Error Seeding Models, Failure Rate Models, Curve Fitting Models, Reliability Growth Models, Markov Structure Models, Time Series Models, Non-homogeneous Poison Process Models.	12Hrs.
UNIT – 3	**Markovian Models:** General Concepts, General Poison-Type Models, Binomial -Type Models, Poison- Type Models, Comparison of Binomial-Type and Poison-Type Models, Fault Reduction Factor for Poison- Type Models.	10Hrs.
UNIT – 4	**Descriptions of Specific Models:** Finite Failure Category Models, Infinite Failure Category Models. Parameter Estimation: Maximum Likelihood Estimation, Least Squares Estimation, Bayesian Inference. Comparison of Software Reliability Models: Comparison Criteria, Comparison of Predictive Validity of Model Groups, Evaluation of other Criteria. **SoftwareReliabilityPrediction:**ProblemsassociatedwithdifferentSoftware ReliabilityModels,Software Reliability prediction parameters, Intelligent Techniques for Software Reliability Prediction.	14Hrs.
UNIT – 5	**Software Quality Management:** Software Quality Attributes, Quality Measurement & Metrics, Verification & Validation Techniques, Verification & Validation in the Life Cycle, Software Quality Assurance functions, Tool support for SQA.	10Hrs.
	Total	56Hrs.

Text Books:

1. M.Xie, Software Reliability Modelling, World Scientific;1991.
2. JohnD. Musa, Anthony Iannino, Kazuhira Okumoto, Software Reliability Measurement, Prediction, Application. McGraw-Hill Book Company; 1987.
3. Hoang Pham, System Software Reliability, Springer;2005
4. David C. Kung, Object-Oriented Software Engineering: An Agile Unified Methodology, McGraw Hill Education (India) Edition 2015.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	NATURAL LANGUAGE PROCESSING	L	T	P	C
		3	0	0	3

Course Objectives: This course introduces the fundamental concepts and techniques of natural language processing (NLP).

- Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information.
- The course examines NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches.
- Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Demonstrate a given text with basic Language features	K2
CO2	Design an innovative application using NLP components	K6
CO3	Explain a rule based system to tackle morphology/syntax of a language	K1
CO4	Design a tag set to be used for statistical processing for real-time applications	K6
C05	Compare and contrast the use of different statistical approaches for different types of NLP applications	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	3	1	2	1
CO2	3	2	3	3	3	2
CO3	3	2	3	2	2	1
CO4	3	2	3	3	3	2
CO5	3	3	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**INTRODUCTION:** Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular	10Hrs

	Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance	
UNIT – 2	**WORD LEVEL ANALYSIS:** Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part- of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models	12Hrs
UNIT – 3	**SYNTACTIC ANALYSIS**: Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures	12Hrs
UNIT – 4	**SEMANTICS AND PRAGMATICS:** Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.	12Hrs
UNIT – 5	**DISCOURSE ANALYSIS AND LEXICAL RESOURCES:** Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC)	12Hrs
	Total	58Hrs

Text Books:

1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2nd Edition, Daniel Jurafsky, James H. Martin -Pearson Publication, 2014.
2. Natural Language Processing with Python, First Edition, Steven Bird, Ewan Klein and Edward Loper, OReilly Media,2009.

Reference Books:

1.Language Processing with Java and Ling Pipe Cookbook, 1st Edition, Breck Baldwin, Atlantic Publisher, 2015.
2.Natural Language Processing with Java, 2nd Edition, Richard M Reese, OReilly

Media, 2015.

3.Handbook of Natural Language Processing, Second, Nitin Indurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.Edition

4. Natural Language Processing and Information Retrieval, 3rdEdition, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press, 2008.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	ADVANCED ALGORITHMS LAB	L	T	P	C
		0	1	2	2

Course Objective:

The student can able to attain knowledge in advanced algorithms.

Course Outcomes:

The student can able to analyze the performance of algorithms

List of Experiments:

UNIT	CONTENTS	Contact Hours
1	Implement assignment problem using Brute Force method	3Hrs
2	Perform multiplication of long integers using divide and conquer method.	3Hrs
3	Implement a solution for the knapsack problem using the Greedy method.	3Hrs
4	Implement Gaussian elimination method.	3Hrs
5	Implement LU decomposition	3Hrs
6	Implement Warshall algorithm	3Hrs
7	Implement the Rabin Karp algorithm.	3Hrs
8	Implement the KMP algorithm.	3Hrs
9	Implement Harspool algorithm	3Hrs
10	Implement max-flow problem.	3Hrs
	Total	30 Hrs

TEXT BOOK:

1. Design and Analysis of Algorithms, S.Sridhar, OXFORD University Press

REFERENCES:

1. Introduction to Algorithms, second edition, T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, PHI Pvt. Ltd./ Pearson Education.

2. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahni and Rajasekharam, Universities Press.

3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	OBJECT ORIENTED SOFTWARE ENGINEERING LAB	L	T	P	C
		0	1	2	3

Suggested Software Tools: Star UML/UML Graph/ Topcased / Umberollo / Argo UML/ Eclipse IDE, Visual Paradigm for UML/Rational Software Architect/Any other Open Source Tool

List of Experiments:

Select domain of interest (e.g. College Management System) and identify multi-tier software application to work on (e.g. Online Fee Collection). Analyze, design and develop this application using OOSE approach:

Pre-requisite:

Course Objectives:

1.
2.
3.
4.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1		
CO2		
CO3		
CO4		
CO5		
CO6		

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1						
CO2						
CO3						
CO4						
CO5						
CO6						

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
1	Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart).	
2	Understanding of System modeling: Data model i.e. ER – Diagram and draw the ER Diagram with generalization, specialization and aggregation of specified problem statement	
3	Understanding of System modeling: Functional modeling: DFD level 0 i.e. Context Diagram and draw it	
4	Understanding of System modeling: Functional modeling: DFD level 1 and DFD level 2 and draw it.	
5	Identify use cases and develop the use case model.	
6	Identify the business activities and develop an UML Activity diagram.	
7	Identity the conceptual classes and develop a domain model with UML Class diagram.	
8	Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.	
9	Draw the state chart diagram.	
10	Identify the user interface, domain objects, and technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.	
11	Implement the technical services layer.	
12	Implement the domain objects layer.	
13	Implement the user interface layer.	
14	Draw component and deployment diagrams.	
	Total	

*Note:

Text Books:

1. Tremblay J.P. and Manohar R., Discrete Mathematical Structures, MGH, 1987

Reference Books:

1. Ellis Horowitz, SartajSahni and SanguthevarRajasekaran, Fundamentals of Computer
2. Algorithms, Second Edition, Universities Press, 2011

Online Learning Resources: (Pls include hyperlinks related to NPTEL/Vlabs/IITB Spoken Tutorial etc.,)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	**FULL STACK TECHNOLOGIES**	**L**	**T**	**P**	**C**
		3	**1**	**0**	**4**

Pre-requisite:

Course Objectives:

1. Translate user requirements into the overall architecture and implementation of new systems and Manage Project and coordinate with the Client.
2. Write backend code in Python/Java, PHP languages and Writing optimized front end code HTML and JavaScript.
3. Understand, create and debug database related queries and Create test code to validate the applications against client requirement.
4. Monitor the performance of web applications & infrastructure and troubleshooting web application with a fast and accurate a resolution

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Identify the Basic Concepts of Web & Markup Languages	K3
CO2	Develop web Applications using Scripting Languages & Frameworks	K3
CO3	Creating & Running Back-end scripts & Connecting to Databases	K6
CO4	Demonstrate Database Queries & DBMS and Working with JQuery Framework	K2
CO5	Adapt to Deployment Techniques & Working with cloud	K6

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**
CO1	2	2	3	2	1	1
CO2	2	2	3	3	2	2
CO3	2	2	3	3	2	2
CO4	2	2	3	3	2	2
CO5	2	2	3	3	3	3

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	**CONTENTS**	**Contact Hours**
UNIT – 1	**Introduction to web-** Internet and world wide web, Domain name service, Protocols: HTTP, FTP, SMTP, **Html5** concepts, **CSS3**, Anatomy of a web page, **XML-** Document type Definition, XML	10Hrs

	schemas, Document object model, XSLT, DOM and SAX Approaches.	
UNIT – 2	**Javascript- The Basics of Javascript:** Objects, Primitives Operations and Expressions, Control Statements, Arrays, Functions, Constructors, Pattern Matching using Regular Expressions, **Angular Java Script- AngularJS Expressions:** ARRAY, Objects, $eval, Strings, AngularJS Form Validation & Form Submission.	10Hrs
UNIT – 3	**PHP Programming: Back – end-** Scripts PHP, Node js, **Working with PHP-** Using variables, Using constants, Data types, Operators. Conditional & Control statements, Arrays, functions. Working with forms and Databases such as MySQL, **Node.js-** Introduction, Advantages, Node.js Process Model, Node JS Modules	12Hrs
UNIT – 4	**JQuery:** Introduction to JQuery, Syntax, Selectors & Events, **MySQL:** Practice MySQL Queries, Aggregate Functions, Regular Expressions, Joins & Unions, Sub-Queries, Database Connectivity with MySql**.**	12Hrs
UNIT – 5	**Mongo DB-** Introduction, Architecture, Features, Examples, Database Creation & Collection in Mongo DB, **Deploying Applications-** Web hosting & Domains, Deployment Using Cloud Platforms, **Web Services-** SOAP, WSDL and RESTful Architecture	12Hrs
	Total	56Hrs

Text Books:

1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013
2. Web Technologies,1st Edition 7th impression, Uttam K Roy, Oxford, 2012.
3. Pro Mean Stack Development, 1st Edition, ELadElrom, Apress O'Reilly, 2016
4. JavaScript & jQuery the missing manual,2nd Edition, David sawyer mcfarland, O'Reilly, 2011
5. Web Hosting for Dummies, 1st Edition, Peter Pollock, John Wiley & Sons,2013
6. RESTful web services, 1st Edition, Leonard Richardson, Ruby, O'Reilly, 2007

Reference Books:

1. Ruby on Rails Up and Running, Lightning fast Web development, 1st Edition, Bruce Tate, Curt Hibbs, Oreilly, 2006
2. Programming Perl, 4th edition, Tom Christiansen, Jonathan Orwant, O'Reilly ,2012
3. Web Technologies, HTML< JavaScript, PHP, Java, JSP, XML and AJAX, Black book, 1st Edition, Dream Tech, 2009
4. An Introduction to Web Design, Programming, 1st Edition, Paul S Wang, Sanda S Katila, Cengage Learning, 2003

Web Reference Links:1. http://www.upriss.org.uk/perl/PerlCourse.html

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	MACHINE LEARNING	L	T	P	C
		3	1	0	4

Pre-requisite:

Course Objectives:

1. Define machine learning and its different types (supervised and unsupervised) and understand their applications.
2. Apply supervised learning algorithms including decision trees and k-nearest neighbours (k-NN).
3. Implement unsupervised learning techniques, such as K-means clustering.
4.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

CO	Course Outcomes	Knowledge Level (K)#
CO1	Enumerate the Fundamentals of Machine Learning	K1
CO2	Build Nearest neighbour based models	K2
CO3	Apply Models based on decision trees and Bayes rule	K4
CO4	Choose appropriate clustering technique.	K2
CO5	Determine algorithms to generate code for a target machine	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	1	1
CO2	3	2	3	3	2	2
CO3	3	2	3	3	2	2
CO4	3	2	3	3	2	2
CO5	3	2	3	3	2	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Introduction to Machine Learning:** Evolution of Machine Learning, Paradigms for ML, Learning by Rote, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Selection, Model Learning, Model Evaluation, Model Prediction, Search and	10Hrs

	Learning, Data Sets	
UNIT – 2	**Nearest Neighbor-Based Models:** Introduction to Proximity Measures, Distance Measures, Non-Metric Similarity Functions, Proximity Between Binary Patterns, Different Classification Algorithms Based on the Distance Measures ,K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression, Performance of Classifiers, Performance of Regression Algorithms	12Hrs
UNIT – 3	**Models Based on Decision Trees**: Decision Trees for Classification, Impurity Measures, Properties, Regression Based on Decision Trees, Bias–Variance Trade-off, Random Forests for Classification and Regression **The Bayes Classifier:** Introduction to the Bayes Classifier, Bayes' Rule and Inference, The Bayes Classifier and its Optimality, Multi-Class Classification | Class Conditional Independence and Naive Bayes Classifier (NBC)	12Hrs
UNIT – 4	**Linear Discriminants for Machine Learning**: Introduction to Linear Discriminants, Linear Discriminants for Classification, Perceptron Classifier, Perceptron Learning Algorithm, Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick, Logistic Regression, Linear Regression, Multi-Layer Perceptron's (MLPs), Backpropagation for Training an MLP.	12Hrs
UNIT – 5	**Clustering** : Introduction to Clustering, Partitioning of Data, Matrix Factorization | Clustering of Patterns, Divisive Clustering, Agglomerative Clustering, Partitional Clustering, K-Means Clustering, Soft Partitioning, Soft Clustering, Fuzzy C-Means Clustering, Rough Clustering, Rough K-Means Clustering Algorithm, Expectation Maximization-Based Clustering, Spectral Clustering	12Hrs
	Total	58Hrs

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024

Reference Books:

1. Machine Learning", Tom M. Mitchell, McGraw-Hill Publication, 2017
2. "Machine Learning in Action",Peter Harrington, DreamTech
3. "Introduction to Data Mining", Pang-Ning Tan, Michel Stenbach, Vipin Kumar, 7th Edition, 2019

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	CLOUD COMPUTING	L	T	P	C
		3	1	0	4

Pre-requisite:

Course Objectives:

- To explain the evolving utility computing model called cloud computing.
- To introduce the various levels of services offered by cloud.
- To discuss the fundamentals of cloud enabling technologies such as distributed computing, service-oriented architecture and virtualization.
- To emphasize the security and other challenges in cloud computing.
- To introduce the advanced concepts such as containers, server less computing and cloud-centric Internet of Things.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Explain different types of Services	K3
CO2	Discuss the hardware architectures for parallel computing	K5
CO3	Demonstrate the building blocks of containers	K4
CO4	Explain the different challenges in cloud computing	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	2	1
CO2	3	2	3	3	2	2
CO3	3	2	3	3	2	2
CO4	3	2	3	2	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	**Contact Hours**
UNIT – 1	**Introduction to Cloud Computing Fundamentals:** Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services(IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google AppEngine).	10Hrs

UNIT – 2	**Cloud Enabling Technologies:** Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures for parallel computing (SISD, SIMD, MISD, MIMD), elements of distributed computing, Inter-process communication, technologies for distributed computing, remote procedure calls (RPC), service-oriented architecture (SOA), Web services, virtualization.	12Hrs
UNIT – 3	**Virtualization and Containers:** Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud Computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM(e.g. Amazon EC2) and container (e.g. Amazon Elastic Container Service)offerings	12Hrs
UNIT – 4	**Cloud computing challenges :** Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.	12Hrs
UNIT – 5	**Advanced concepts in cloud computing :** Server less computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g. AWS Lambda) and open-source (e.g. OpenFaaS) serverless platforms, Internet of Things (IoT), applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.	12Hrs
	Total	58Hrs

Text Books:

1. Andrew Stellman, Jill Alison Hart, Learning Agile, O'Reilly, 2015.

Reference Books:

1. Andrew stellman, Jennifer Green, Head first Agile, O'Reilly, 2017.
2. Rubin K , Essential Scrum : A practical guide to the most popular Agile process,
Addison-Wesley, 2013

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

I Semester	INFORMATION STORAGE MANAGEMENT	L	T	P	C

Pre-requisite:

Course Objectives:

1. To understand various segments of storage technology and architectures
2. To explore the inherent power of information
3. To describe the different backup, recovery and replication strategies

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understand the evolution of storage technology and Intelligent Storage Systems	
CO2	Explore the key concepts of various Storage Networking Technologies- DAS, SANs, NAS and CAS	
CO3	Understand the basics of Storage Virtualization	
CO4	Understand the concepts of Storage security and Storage Infrastructure Management	
CO5	Analyze the purpose of backup, recovery and replication Strategies	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	2	1
CO2	3	2	3	3	2	2
CO3	3	2	3	2	2	2
CO4	3	2	3	3	3	2
CO5	3	2	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Introduction to Information Storage and Management:** Information Storage, Evolution of Storage Technology and Architecture, Data Center Infrastructure, Key Challenges in Managing Information, Information Lifecycle. Storage System Environment - Data Protection: RAID - Intelligent Storage System.	
UNIT – 2	**Direct-Attached Storage and Introduction to SCSI:** Types of DAS, DAS Benefits and Limitations, Disk Drive Interfaces, Introduction to Parallel SCSI, SCSI Command Model,	

	Storage Area Networks: Fibre Channel: Overview, The SAN and Its Evolution, Components of SAN, FC Connectivity, Fibre ChannelPorts, Fibre Channel Architecture, Zoning, Fibre Channel Login Types, FC Topologies, Concepts in Practice: EMC Connectrix **Network-Attached Storage:** General-Purpose Servers vs. NAS Devices, Benefits of NAS, NAS File I/O, Components of NAS, NAS Implementations, NAS File-Sharing Protocols, NAS I/O Operations, Factors Affecting NAS Performance and Availability, Concepts in Practice: EMC Celerra	
UNIT – 3	**Content-Addressed Storage:** Fixed Content and Archives, Types of Archives, Features and Benefits of CAS, CAS Architecture, Object Storage and Retrieval in CAS, CAS Examples, Concepts in Practice: EMC Centera **Storage Virtualization:** Forms of Virtualization, SNIA Storage Virtualization Taxonomy, Storage Virtualization Configurations, Storage Virtualization Challenges, Types of Storage Virtualization, Concepts in Practice	
UNIT – 4	**Backup and Recovery** Backup Purpose, Backup Considerations, Backup Granularity, Recovery Considerations, Backup Methods, Backup Process, Backup and Restore Operations, Backup Topologies, Backup in NAS Environments, Backup Technologies, Concepts in Practice: EMC Net Worker **Local Replication:** Local Replication, Source and Target, Uses of Local Replicas, Data Consistency, Local Replication Technologies, Restore and Restart Considerations, Creating Multiple Replicas, Management Interface, Concepts in Practice: EMC Time Finder and EMC Snap View **Remote Replication:** Modes of Remote Replication, Remote Replication Technologies, Network Infrastructure, Concepts in Practice: EMC SRDF, EMC SAN Copy, and EMC Mirror View	
UNIT – 5	**Securing the Storage Infrastructure** Storage Security Framework, Risk Triad, Storage Security Domains, Security Implementations in Storage Networking **Managing the Storage Infrastructure** Monitoring the Storage Infrastructure, Storage Management Activities, Storage Infrastructure Management Challenges, Developing an Ideal Solution, Concepts in Practice: EMC Control Center	

	Total	

Text Books:

1. Marc Farley Osborne, "Building Storage Networks'', Tata McGraw-Hill, 2001.
2. Robert Spalding and Robert Spalding, "Storage Networks: The Complete Reference", Tata McGraw Hill, 2003.
3. Meeta Gupta, "Storage Area Network Fundamentals", Pearson Education Ltd., 2002.

Reference Books:

1. Gerald JKowalski and Mark T May bury,"Information Storage Retrieval Systems theory & Implementation", BS Publications, 2000.
2. The jendra BS, "Disaster Recovery& Business continuity", Shroff Publishers& Distributors, 2006.

Online Learning Resources: (Pls include hyperlinks related to NPTEL/Vlabs/IITB Spoken Tutorial etc.,)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	AGILE METHODOLOGIES	L	T	P	C
		3	0	0	3

Pre-requisite:

Course Objectives:

The main objectives of this course are to introduce the important concepts of agile software development Process, emphasize the role of stand-up meetings in software collaboration, impart the knowledge on values and principles in understanding agility

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	How to learn Agile and get into brain	K1
CO2	Explain about the Agile Principles and Agile Project	K3
CO3	Describe the concepts of XP, Simplicity, and Incremental Design	K2
CO4	Develop a new Lean, Eliminating Waste, and Seeing	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	3	2	1	1
CO2	3	2	3	3	2	2
CO3	3	2	3	3	2	2
CO4	3	2	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Learning Agile**: Getting Agile into your brain, Understanding Agile values, No Silver Bullet, Agile to the Rescue, adding Agile makes a difference. A fractured perspective, How a fractured perspective causes project problems. The Agile Manifesto, Purpose behind Each Practice. Individuals and Interactions Over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract Negotiation, Responding to Change over Following a Plan, Principles over Practices. Understanding the Elephant, Methodologies Help You Get It All in Place at Once, Where to Start with a New Methodology	10Hrs
UNIT –	**The Agile Principles:** The 12 Principles of Agile Software, The Customer	12Hrs

2	Is Always Right, "Do As I Say, Not As I Said". Delivering the Project, Better Project Delivery for the Ebook Reader Project. Communicating and Working Together, Better Communication for the Ebook Reader Project. Project Execution—Moving the Project Along, A Better Working Environment for the Ebook Reader Project Team. Constantly Improving the Project and the Team. The Agile Project: Bringing All the Principles Together	
UNIT – 3	**SCRUM and Self-Organizing Teams:** The Rules of Scrum, Act I: I Can Haz Scrum?, Everyone on a Scrum Team owns the Project, The Scrum Master Guides the Team's Decisions, The Product Owner Helps the Team Understand the Value of the Software, Everyone Owns the Project, Scrum Has Its Own Set of Values ,Status Updates Are for Social Networks!, The Whole Team Uses the Daily Scrum, Feedback and the Visibility-Inspection-Adaptation Cycle, The Last Responsible Moment, How to Hold an Effective Daily Scrum. Sprinting into a Wall, Sprints, Planning, and Retrospectives, Iterative or Incremental?, The Product Owner Makes or Breaks the Sprint, Visibility and Value, How to Plan and Run an Effective Scrum Sprint **Scrum Planning And Collective Commitment:** Not Quite Expecting the Unexpected, User Stories, Velocity, and Generally Accepted Scrum Practices, Make Your Software Useful, User Stories Help Build Features Your Users Will Use, Conditions of Satisfaction, Story Points and Velocity, Burndown Charts, Planning and Running a Sprint Using Stories, Points, Tasks, and a Task Board. Victory Lap, Scrum Values Revisited, Practices Do Work Without the Values (Just Don't Call It Scrum), Is Your Company's Culture Compatible with Scrum Values.	12Hrs
UNIT – 4	**XP And Embracing Change:** Going into Overtime, The Primary Practices of XP, Programming Practices, Integration Practices, Planning Practices, Team Practices, Why Teams Resist Changes, and How the Practices Help. The Game Plan Changed, but We're Still Losing, The XP Values Help the Team Change Their Mindset, XP Helps Developers Learn to Work with Users, Practices Only "Stick" When the Team Truly Believes in Them, An Effective Mindset Starts with the XP Values, The XP Values, Paved with Good Intentions. The Momentum Shifts, Understanding the XP Principles Helps You Embrace Change, The Principles of XP, XP Principles Help You Understand Planning, XP Principles Help You Understand Practices—and Vice Versa, Feedback Loops. **XP, Simplicity, and Incremental Design:** Code and Design, Code Smells and Antipatterns (or, How to Tell If You're Being Too Clever), XP Teams	12Hrs

	Look for Code Smells and Fix Them, Hooks, Edge Cases, and Code That Does Too Much. Make Code and Design Decisions at the Last Responsible Moment, Fix Technical Debt by Refactoring Mercilessly, Use Continuous Integration to Find Design Problems, Avoid Monolithic Design, Incremental Design and the Holistic XP Practices. Teams Work Best When They Feel Like They Have Time to Think, Team Members Trust Each Other and Make Decisions Together. The XP Design, Planning, Team, and Holistic Practices Form an Ecosystem Incremental Design Versus Designing for Reuse, When Units Interact in a Simple Way, the System Can Grow Incrementally, Great Design Emerges from Simple Interactions, Final Score.	
UNIT – 5	**Lean, Eliminating Waste, and Seeing the whole:** Lean Thinking, Commitment, Options Thinking, and Set-Based Development, Creating Heroes and Magical Thinking. Eliminate Waste, Use a Value Stream Map to Help See Waste Clearly, Gain a Deeper Understanding of the Product, See the Whole, Find the Root Cause of Problems That You Discover. Deliver As Fast As Possible, Use an Area Chart to Visualize Work in Progress, Control Bottlenecks by Limiting Work in Progress. **Kanban, Flow, and Constantly Improving:** The Principles of Kanban, Find a Starting Point and Evolve Experimentally from There. Stories Go into the System; Code Comes Out, Improving Your Process with Kanban, Visualize the Workflow, Limit Work in Progress. Measure and Manage Flow, Managing Flow with WIP Limits Naturally Creates Slack. Make Process Policies Explicit So Everyone Is on the Same Page. Emergent Behavior with Kanban. **The Agile Coach:** Coaches Understand Why People Don't Always Want to Change. The Principles of Coaching.	12Hrs
	Total	58Hrs

Text Books:

1. Andrew Stellman, Jill Alison Hart, Learning Agile, O'Reilly, 2015.

Reference Books:

1. Andrew stellman, Jennifer Green, Head first Agile, O'Reilly, 2017.
2. Rubin K , Essential Scrum : A practical guide to the most popular Agile process, Addison-Wesley, 2013

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	SOFTWARE DESIGN METHODOLOGIES	L	T	P	C
		3	0	0	3

Pre-requisite: Software Engineering.

Course Objectives:

- To develop the knowledge, understanding, skills and values to solve problems through the creation of software solutions
- To design and experiment with software prototypes
- To elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.
- To build solutions using different technologies, architectures and life-cycle approaches.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understand the need of software design and challenges.	
CO2	Demonstrate architectures and methods for software design.	
CO3	Analyze software design process with objects and components.	
CO4	Identify suitable metrics for project and process management.	
CO5	Analyze project scheduling and risk management strategies	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	3	2	2	1
CO2	2	2	3	3	2	1
CO3	3	2	3	3	2	2
CO4	2	2	2	3	3	2
CO5	3	2	2	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	**Contact Hours**
UNIT – 1	**Software Requirements:** Functional and non-functional requirements, User requirements, System requirements, Interface specification, the	

	software requirements document. Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.	
UNIT – 2	**Software Design** The nature of the design process, transferring design knowledge, constraints upon the design process and product, recording design decisions, designing with others, context for design, economic factors, assessing design qualities, quality attributes of the design product, assessing the design process. Representing abstract ideas, design viewpoints, the architecture concept, design methods, design patterns, design representations, rationale for design methods. Design Processes and Strategies: The role of strategy in design methods, describing the design process – The D – Matrix, design by top-down decomposition, design by composition, organizational influences upon design.	
UNIT – 3	**Designing with objects and components:** Designing with objects- Design practices for object-oriented paradigm, Object-oriented paradigm, Object-oriented frameworks, Hierarchical object oriented design process and heuristics, the fusion method, the unified process. Component - based design- The component concept, designing with components, designing components, COTS. User Interface design-The Golden rules, Interface analysis and design models, user and task analysis, analysis of display content and work environment, applying interface design issues, design evaluation.	
UNIT – 4	**Project Management Concepts:** Project Management- The management spectrum, people, product, process and project, W5HH principle, Critical practices. Metrics for Process and Projects- Process metrics, project metrics, size- oriented metrics, function-oriented metrics, Object-oriented and use-case metrics, metrics for software quality, integrating metrics within software process.	
UNIT – 5	**Project Scheduling and Management** Project Scheduling- Basic concepts, project scheduling, defining at ask set and task network, timeline charts, tracking the schedule, tracking the progress for an OO project, Earned value analysis. Risk Management- Reactive vs. Proactive risk strategies, software risks, risk identification, risk projection, risk refinement, risk mitigation and monitoring, the RMMM plan.	
	Total	

*Note:

Text Books:

1. Software design, David Budgen, second edition, Pearson education,2003.
2. Software Engineering: A practitioner's Approach, Roger S Pressman, sixth edition. McGraw- Hill International Edition, 2005.

Reference Books:

1. Applying domain-driven design and patterns, jimmy Nilsson, Pearson education,2006
2. Software Engineering Foundations, Ian Somerville, seventh edition, Pearson education, 2004.
3. Software Project Management, Bob Hughes & Mike Cotterell, Fourth edition, Tata Mc Graw Engineering: A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
4. The Art of Project Management, Scott Berkun, O'Reilly,2005.
5. Software Engineering, Project Management, Richard H. Thayer & Edward Yourdon, second edition, Wiley India, 2004.
6. Software Engineering foundations, Yingxu Wang Auerbach publications,2008.
7. Applied Software Project Management, Andrew Stellman& Jennifer Greene,O'Reilly,200

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	SOFTWARE ARCHITECTURE & DESIGN PATTERNS	L	T	P	C
		3	0	0	3

Pre-requisite:

Course Objectives:

1.
2.
3.
4.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understand the basic concepts to identify state behavior of real world objects	
CO2	Apply Object Oriented Analysis and Design concepts to solve complex problems	
CO3	Construct various UML models using the appropriate notation for specific problem context	
CO4	Design models to Show the importance of systems analysis and design in solving complex problems using case studies	
CO5	Study of Pattern Oriented approach for real world problems	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	3	2	1	1
CO2	3	2	3	3	2	2
CO3	2	2	3	3	2	1
CO4	3	3	3	3	2	2
CO5	2	2	3	2	2	1

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	**Contact Hours**
UNIT – 1	**Introduction**: What is a design pattern? Describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern What is object oriented development? key concepts of object	

	oriented design other related concepts, benefits and drawbacks of the paradigm	
UNIT – 2	**Analysis a System:** Overview of the analysis phase, stage 1 gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain Design and Implementation, discussions and further reading	
UNIT – 3	**Design Pattern Catalog:** Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.	
UNIT – 4	**Interactive systems and the MVC architecture:** Introduction The MVC architectural pattern, analyzing a simple drawing program designing the system, designing of the subsystems, getting into implementation, implementing undo operation drawing incomplete items, adding a new feature pattern based solutions	
UNIT – 5	**Designing with Distributed Objects:** Client server system, java remote method invocation, implementing an object oriented system on the web, Web services (SOAP, Restful), Enterprise Service Bus	
	Total	

*Note:

Text Books:

1. Object oriented analysis, design and implementation, brahma dathan, sarnathrammath , universities press,2013
2. Design patterns, Erich Gamma, Richard helan, Ralph johman , john vlissides, PEARSON Publication,2013

Reference Books:

1. Frank Bachmann, RegineMeunier , Hans Rohnert “Pattern Oriented Software Architecture” Volume 1, 1996.
2. William J Brown et al., "Anti Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	**SOFTWARE QUALITY ENGINEERING**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Pre-requisite:

Course Objectives:

Knowledge on significance of Quality, quality assurance, quality engineering.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understand software quality and its perspectives	
CO2	Analyze defect prevention and defect reduction in software quality assurance	
CO3	Illustrate software quality engineering activities and its process	
CO4	Understand and apply various test planning, execution, automation, and management techniques for software testing. Analyze coverage and usage-based testing methods using checklists, partitions, and operational profiles with real-world case studies.	
CO5	Analyze coverage and usage-based testing methods using checklists, partitions, and operational profiles with real-world case studies.	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**
CO1	2	1	2	2	3	1
CO2	3	2	3	2	3	2
CO3	2	2	3	3	2	1
CO4	2	2	3	3	3	2
CO5	3	2	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	**Contact Hours**
UNIT – 1	**Software Quality:** perspectives and expectations, Quality frameworks and ISO-9126, correctness and defects: Definitions, properties and Measurements, Ahistorical perspective of quality, software quality.	
UNIT – 2	**Quality Assurance:** Classification: QA as dealing with defects, Defect prevention- Education and training, Formal method, Other defect	

	prevention techniques, Defect Reduction- Inspection: Direct fault detection and removal, Testing: Failure observation and fault removal, other techniques and risk identification, Defect Containment- software fault tolerance, safety assurance and failure containment	
UNIT – 3	**Quality Engineering:** Activities and process, Quality planning: Goal setting and Strategy formation, Quality assessment and Improvement, Quality engineering in software process.	
UNIT – 4	**Test Activities, Management and Automation:** Test planning and preparation, Test execution, Result checking and measurement, Analysis and follow- up, Activities People and Management, Test Automation.	
UNIT – 5	**Coverage and usage testing based on checklist and partitions:** Checklist based testing and its limitations, Testing for partition Coverage, Usage based Statistical testing with Musa's operational profiles, Constructing operational profiles **Case Study:** OP for the cartridge Support Software	
	Total	

Text Books:

1. JeffTia`n, Software Quality Engineering, Testing, Quality Assurance, and Quantifiable improvement
2. Richard N.Taylor, Software Architecture: Foundations,Theory,andPractice

Reference Books:

Online Learning Resources: (Pls include hyperlinks related to NPTEL/Vlabs/IITB Spoken Tutorial etc.,)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	SOCIAL MEDIA ANALYTICS	L	T	P	C
		3	0	0	3

Pre-requisite:

Course Objectives:

1. Knowledge on social media and its analytics
2.
3.
4.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understanding characteristics and types of social media.	
CO2	Knowledge on layers of social media analytics	
CO3	Apply text analysis tools on social media data	
CO4	Understand the significance of action analytics	
CO5	Detect viral topics on social media(YouTube)	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	3	1	1	1
CO2	2	1	3	2	2	1
CO3	3	2	3	3	2	2
CO4	2	2	3	2	2	2
CO5	3	2	3	2	2	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Introduction To Social Media:** World Wide Web, Web 1.0, Web 2.0, Web 3.0, Social Media, Core Characteristics Of Social Media ,Types Of Social Media, Social Networking Sites, Using Face book For Business Purposes, Content Communities	
UNIT – 2	**Social Media Analytics Overview:** Purpose of Social Media Analytics, Social Media Vs. Traditional Business Analytics, Seven Layers Of Social Media Analytics, Types Of Social Media Analytics, Social Media Analytics Cycle, Challenges To Social Media Analytics, Social Media Analytics Tools. **Case Study**: The Underground Campaign That Scored Big	

UNIT – 3	**Social Media Text Analytics:** Types Of Social Media Text, Purpose Of Text Analytics, Steps In Text Analytics, Social Media Text Analysis Tools. **Case Study**: Tapping into Online Customer Opinions	
UNIT – 4	**Social Media Actions Analytics:** Introduction To Actions Analytics, Common Social Media Actions, Actions Analytics Tools. **Case Study**: Cover-More Group	
UNIT – 5	**Social Media Hyperlink Analytics:** Types Of Hyperlinks, Hyperlink Analytics, Types Of Hyperlink Analytics, Hyperlink Analytics Tools. **Case Study**: Hyperlinks and Viral YouTube Videos	
	Total	

*Note:

Text Books:

1. Seven Layers of Social Media Analytics Mining Business Insights From Social Media Text, Actions, Networks, Hyperlinks, Apps, Search Engine, And Location Data By GoharF.Khan Isbn: 1507823207, Isbn-13: 9781507823200

Reference Books:

1. Social Media Analytics: Techniques and Insights for Extracting Business Value out of Social Media By Matthew Ganis, Avinash Kohirkar, Pearson Education.
2. Social Media Analytics: Effective Tools for Building, Interpreting, and Using Metrics, Marshall Sponder, MGH.
3. Big Data and Analytics, Seema Acharya, Subhasin in Chellappan, WileyPublications.
4. Big Data, Black Book^{tm},Dreamtech Press,2015Edition

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	SOFTWARE REQUIREMENTS AND ESTIMATION	L	T	P	C
		3	0	0	3

Pre-requisite:

Course Objectives:

1. Students will author a software requirements document.
2. Students will demonstrate an understanding of the proper contents of a software requirements document and proficiency in software development cost estimation

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Identify the importance of software requirements and their validation.	
CO2	Analyse the software requirement management and its principles.	
CO3	Apply and estimate the factors and approaches for software cost.	
CO4	Utilize tools for management requirements and estimation requirements	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	3	2	2	1
CO2	3	2	3	2	2	2
CO3	3	2	3	2	3	2
CO4	3	2	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Software Requirements and Engineering:** Software Requirements - Essential Software requirement, Good practices for requirements engineering, improving requirements processes, Software requirements and risk management. Software Requirements Engineering- Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality	
UNIT – 2	**Software Requirements Management and Modeling:** Software Requirements Management - Requirements Management Principles and	

	practices, Requirements attributes , Change Management Process, Requirements Traceability Matrix, and Links in requirements chain. Software Requirements Modeling Use Case Modeling, Analysis Models, Dataflow diagram, state transition diagram, class diagrams, Object analysis, Problem Frames	
UNIT – 3	**Software Estimation:** Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation. Size Estimation-Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, and Conversion between size measures.	
UNIT – 4	**Effort, Schedule and Cost Estimation:** Productivity, Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Putnam Estimation Model, Algorithmic models, Cost Estimation.	
UNIT – 5	**Tools for Requirements Management and Estimation Requirements Management Tools:** Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation, Software Estimation Tools: Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools.	
	Total	

*Note:

Text Books:

1. Software Requirements and Estimation by Rajesh Naik and Swapna Kishore, Tata Mc Graw Hill.

Reference Books:

1. Software Requirements by Karl E. Weigers, Microsoft Press.
2. Managing Software Requirements, Dean Leffingwell& Don Widrig, Pearson Education, 2003.
3. . Mastering the requirements process, second edition, Suzanne Robertson & James Robertson, Pearson Education, 2006.
4. Estimating Software Costs, Second edition, Capers Jones, TMH, 2007.
5. Practical Software Estimation, M.A. Parthasarathy, Pearson Education, 2007.
6. Measuring the software process, William A. Florac& Anita D. Carleton, Pearson Education, 1999

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	**BLOCK-CHAIN TECHNOLOGIES (PROFESSIONAL ELECTIVE-VIII)**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Pre-requisite:

Course Objectives:

1. To learn the fundamentals of Block Chain and various types of block chain and consensus mechanism.

2.To understand public block chain system, Private block chain system and consortium block chain.

3.Able to know the security issues of block chain technology.

4.

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understand crypto currencies like Bitcoin, altcoins, and how they are used.	K1
CO2	Explain the concept, characteristics, types, and real-world applications of smart contracts and oracles.	K2
CO3	Analyze smart contracts, permissioned algorithms, and applications of blockchain in private and consortium environments.	K4
CO4	Understand the security, privacy, and regulatory aspects of blockchain systems including Bitcoin and Hyperledger Fabric.	K2
CO5	Build real-world blockchain applications across sectors like retail, banking, healthcare, and energy through case studies.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**
CO1	2		2		2	
CO2	2	2	3		2	
CO3	3	2	3	2	3	
CO4		2	2		3	
CO5	3	3	3	3	3	2

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	**Contact Hours**
UNIT – 1	**Fundamentals of Blockchain:** Introduction, Origin of Blockchain, Blockchain Solution, Components of Blockchain, Block in a Blockchain,	

	The Technology and the Future. Blockchain Types and Consensus Mechanism: Introduction, Decentralization and Distribution, Types of Blockchain, Consensus Protocol. Cryptocurrency: Bitcoin, Altcoin and Token: Introduction, Bitcoin and the Cryptocurrency, Cryptocurrency Basics, Typesof Cryptocurrencies, Cryptocurrency Usage.	
UNIT – 2	**Public Blockchain System:** Introduction, Public Blockchain, Popular Public Blockchains, The Bitcoin Blockchain, Ethereum Blockchain. Smart Contracts: Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart Contracts, Types of Oracles, Smart Contracts in Ethereum, Smart Contracts in Industry.	
UNIT – 3	**Private Blockchain System**: Introduction, Key Characteristics of Private Blockchain, Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, E- commerce Site Example, Various Commands (Instructions) in E-commerce Blockchain, Smart Contract in Private Environment, State Machine, Different Algorithms of Permissioned Blockchain, Byzantine Fault, Multichain. Consortium Blockchain: Introduction, Key Characteristics of Consortium Blockchain, Need of Consortium Blockchain, Hyperledger Platform, Overview of Ripple, Overview of Corda.	
UNIT – 4	**Security in Blockchain:** Introduction, Security Aspects in Bitcoin, Security and Privacy Challenges of Blockchain in General, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract (DApp), Security Aspects in Hyperledger Fabric.	
UNIT – 5	**Blockchain Case Studies:** Case Study 1 – Retail, Case Study 2 – Banking and Financial Services, Case Study 3 – Healthcare, Case Study 4 – Energy and Utilities. Blockchain Platform using Python: Introduction, Learn How to Use Python Online Editor, Basic Programming Using Python, Python Packages for Blockchain. Blockchain platform using Hyperledger Fabric: Introduction, Components of Hyperledger Fabric Network, Chain codes from Developer.ibm.com, Blockchain Application Using Fabric Java SDK.	
	Total	

Text Books:

1. “Block chain Technology”, Chandramouli Subramanian, Asha A.George, Abhilasj K A and MeenaKarthikeyan , Universities Press.

Reference Books:

1. Block chain Blue print for Economy, Melanie Swan, SPD Oreilly.
2. Block chain for Business, Jai Singh Arun, Jerry Cuomo, Nitin Gauar, Pearson Addition Wesley

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	FULL STACK TECHNOLOGIES LAB	L	T	P	C
		0	1	2	2

Pre-requisites:

COURSE OBJECTIVES: From the course the student will

- Learn the core concepts of both the frontend and backend programming course.
- Get familiar with the latest web development technologies.
- Learn all about SQL and Mongo databases.

Learn complete web development process.

Software's Required:

UNIT	CONTENTS	Contact Hours
Experiment– 1	1. Design a web page consisting of a) Home page b) Login page c) Catalogue page.	
Experiment– 2	Design a webpage to Embed elements like google maps and youtube into the webpage and make them responsive	
Experiment– 3	Design a dynamic web page with validation using JavaScript.	
Experiment– 4	Design a HTML having a text box and four buttons viz Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate javascript function should be called to display a. Factorial of that number b. Fibonacci series up to that number c. Prime numbers up to that number d. Is it palindrome or not	
Experiment– 5	Write JavaScript programs on Event Handling a. Validation of registration form b. Open a Window from the current window c. Change color of background at each click of button or refresh of a page d. Display calendar for the month and year selected from combo box e. On Mouse over event	
Experiment– 6	Write an XML file which will display the Book information which includes	

	the following: 1) Title of the book 2) Author Name 3) ISBN number 4) Publisher name 5) Edition 6) Price Write a Document Type Definition (DTD) to validate the above XML file.	
Experiment– 7	Write a program to create Calculator Node.js Module with functions adds, subtract & multiply and use the Calculator module in another Node.js file.	
Experiment– 8	Write a Node.js for File System to perform the following operations i) Create a File ii) Read a File iii) Write to a File iv) Delete a File	
Experiment– 9	Write a program to implement jQuery Selectors and Operations	
Experiment– 10	Write a program to implement jQuery Event Handling	
Experiment– 11	Write an example perl program to connect to a MySQl database table and execute simple commands	
Experiment– 12	Write a PHP program for registering users of a website and login	
Experiment– 13	User Authentication:Assume four users user1,user2,user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a PHP for doing the following. a. Write a program to Create a Cookie and add these four user id's and passwords to this Cookie. b. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies. If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display " You are not an authenticated user ''. Use init-parameters to do this.	
Experiment– 14	Install a database (Mysql or Oracle): Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).	

	a) Write a PHP program to connect to that database and extract data from the tables and display them. b) Experiment with various SQL queries. c) Insert the details of the users who register with the web site, Whenever a new user clicks the submit button in the registration page.	
Experiment–15	Write a PHP program which does the following job: Insert the details of the 3 or 4 users who register with the web site by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (Similar to week8 instead of cookies).	

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

R25 M.TECH SOFTWARE ENGINEERING

II Semester	MACHINE LEARNING LAB	L	T	P	C
		0	1	2	2

Pre-requisites: Data Base Management Systems, Python Programming

COURSE OBJECTIVES: The main objective of the courseisto

- To learn about computing central tendency measures and Datapre-processing techniques
- To learn about classification and regression algorithms
- To apply different clustering algorithms for a problem.

Software's Required: Python/R/Weka

UNIT	CONTENTS	Contact Hours
Experiment–1	Compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation.	10Hrs
Experiment–2	Apply the following Pre-processing techniques for a given dataset. a. Attribute selection b. Handling Missing Values c. Discretization d. Elimination of Outliers	12Hrs
Experiment–3	Apply KNN algorithm for classification and regression	12Hrs
Experiment–4	Demonstrate decision tree algorithm for a classification problem and Perform parameter tuning for better results	
Experiment–5	Demonstrate decision tree algorithm for a regression problem	12Hrs
Experiment–6	Apply Random Forest algorithm for classification and regression	12Hrs
Experiment–7	Demonstrate NaïveBayes Classification algorithm	
Experiment–8	Apply Support Vector algorithm for classification	

Experiment–9	Demonstrate simple linear regression algorithm for a regression problem	
Experiment–10	Apply Logistic regression algorithm for a classification problem	
Experiment–11	Demonstrate Multi-layer Perceptron algorithm for a classification problem	
Experiment–12	Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.	
Experiment–13	Demonstrate the use of Fuzzy C-Means Clustering	
Experiment–14	Demonstrate the use of Expectation Maximization based clustering algorithm	
	Total	58Hrs