DELHI TECHNOLOGICAL UNIVERSITY

SCHEME OF TEACHING AND EVALUATION

M.TECH SOFTWARE ENGINEERING

The following alphanumeric coding scheme has been adoptedCore Courses XXXYMN
Elective Courses XXXYCMN
XXX abbreviates a particular M. Tech. program, Y – (5 for M. Tech. 1 st year, 6 for M. Tech. 2 nd year),C – credit of the course (4)
MN – Subject code (Odd number for odd semester and even number for even semester courses

Semester I

S. No.	Course Code	Course Name	Type/Area	Cr	L	T	P	CWS	PRS	MTE	ETE	PRE	Total Credits
1	SWE501	Software Requirement Engineering	Department Core -1	4	3	0	2	15	25	20	40	-	
2	SWE503	Object-Oriented Software Engineering	Department Core -2	4	3	0	2	15	25	20	40	-	
3	SWE505	Advanced Data Structures and Algorithms	Department Core -3	4	3	0	2	15	25	20	40	-	
4	SWE507	Machine Learning	Department Core -4	4	3	0	2	15	25	20	40	-	
5	SWE5401/ SWE5403/ …….	Elective - 1	Department Elective -1	4	3 / 3	0 / 1	2 / 0	15 / 25	25 / 0	20 / 25	40 / 50	-	**24**
6	SWE509	Seminar	Self-Study	2	2	0	0	-	-	-	100	-	
7	SWE511	Research Paper Writing	Skill Enhancement Course - 1	2	2	0	0	20	-	30	50	-	
8	SWE513	Audit Course	Audit Course	0									

Semester II

S. No.	Course Code	Course Name	Type/Area	Cr	L	T	P	CWS	PRS	MTE	ETE	PRE	Total Credits
1	SWE502	Software Testing	Department Core -5	4	3	0	2	15	25	20	40	-	
2	SWE504	Empirical Software Engineering	Department Core -6	4	3	1	0	25	-	25	50	-	
3	SWE5402/ SWE5404/ ……	Elective – 2	Department Elective -2	4	3 / 3	0 / 1	2 / 0	15 / 25	25 / 0	20 / 25	40 / 50	-	

4	SWE5416/ SWE5418/ ……	Elective - 3	Department Elective -3	4	3 / 3	0 / 1	2 / 0	15 / 25	25 / 0	20 / 25	40 / 50	-	**24**
5	UEC	Research Methodology	University Core	4	3	0	2	15	25	20	40	-	
6	SWE506	Software Engineering Project and Tools	Skill Enhancement Course - 1	4	0	0	4	-	50	-	-	50	

Semester III

S. No.	Course Code	Course Name	Type/Area	Cr	L	T	P	CWS	PRS	MTE	ETE	PRE	Total Credit
1	SWE601	Software Quality and Metrics	Department Core -7	4	3	1	0	25	-	25	50	-	**16**
2	SWE6401/ SWE6403/ …… ……	Elective - 4	Open Elective - I	4	3	0	2	15	25	20	40	-	
3	SWE603	Minor Project/ Research Thesis/ Patent	Core	8	0	0	8	-	50	-	-	50	

Semester IV

S. No.	Course Code	Course Name	Type/Area	Cr	L	T	P	CWS	PRS	MTE	ETE	PRE	Total Credits
1	SWE602	Major Project/ Research Thesis/ Patent	Core	16	0	0	16	-	-	-	100	-	**16**

LIST OF ELECTIVES :

	S.No.	Course Code	Course Name	Type/ Area	Cr	L	T	P	CWS	PRS	MTE	ETE	PRE
Elective -1	1	SWE5401	Advanced Database Management Systems	Elective	4	3	0	2	15	25	20	40	-
	2	SWE5403	Data Warehousing Data Mining		4	3	0	2	15	25	20	40	-
	3	SWE5405	Probability and Statistics		4	3	0	2	15	25	20	40	-
	4	SWE5407	Advanced Operating System		4	3	1	0	25	-	25	50	-
	5	SWE5409	Software Design Patterns		4	3	0	2	15	25	20	40	-
	6	SWE5411	Agile Methods		4	3	1	0	25	-	25	50	-
	7	SWE5413	Fuzzy Logic and Neural Networks		4	3	1	0	25	-	25	50	-
	S.No.	**Course Code**	**Course Name**	**Type/ Area**	**Cr**	**L**	**T**	**P**	**CWS**	**PRS**	**MTE**	**ETE**	**PRE**
Elective -2	1	SWE5402	Software Project Management	Elective	4	3	1	0	25	-	25	50	-
	2	SWE5404	Distributed Systems		4	3	1	0	25	-	25	50	-
	3	SWE5406	Fundamentals of Information Retrieval		4	3	1	0	25	-	25	50	-
	4	SWE5408	Artificial Intelligence		4	3	1	0	25	-	25	50	-
	5	SWE5410	Advances in Software Engineering		4	3	0	2	15	25	20	40	-
	6	SWE5412	Multimedia Applications		4	3	0	2	15	25	20	40	-
	7	SWE5414	Deep Learning		4	3	0	2	15	25	20	40	-
	S.No.	**Course Code**	**Course Name**	**Type/ Area**	**Cr**	**L**	**T**	**P**	**CWS**	**PRS**	**MTE**	**ETE**	**PRE**
Elective - 3	1	SWE5416	Soft Computing	Elective	4	3	1	0	25	-	25	50	-
	2	SWE5418	Software Reliability		4	3	1	0	25	-	25	50	-
	3	SWE5420	Natural Language Processing		4	3	1	0	25	-	25	50	-
	4	SWE5422	Computer Vision		4	3	0	2	15	25	20	40	-
	5	SWE5424	Optimization Techniques		4	3	1	0	25	-	25	50	-
	6	SWE5426	Information Theory and Coding		4	3	0	2	15	25	20	40	-
	7	SWE5428	Big Data Analytics		4	3	1	0	25	-	25	50	-

	S.No.	Course Code	Course Name	Type/ Area	Cr	L	T	P	CWS	PRS	MTE	ETE	PRE
Open Elective	1	SWE6401	Pattern Recognition	Elective	4	3	1	0	25	-	25	50	-
	2	SWE6403	Mobile Computing		4	3	0	2	15	25	20	40	-
	3	SWE6405	Cloud Computing		4	3	0	2	15	25	20	40	-
	4	SWE6407	Cluster and Grid Computing		4	3	0	2	15	25	20	40	-
	5	SWE6409	Swarm and Evolutionary Computing		4	3	1	0	25	-	25	50	-
	6	SWE6411	Internet of Things		4	3	0	2	15	25	20	40	-
	7	SWE6413	Software Maintainability		4	3	1	0	25	-	25	50	-

DELHI TECHNOLOGICAL UNIVERSITY
Department of Software Engineering
SYLLABUS: M.Tech. (Software Engineering)

SEMESTER I

SWE501 Software Requirement Engineering

The primary objective of the course is to understand the software requirements, how to effectively collect them and to learn the corresponding tools that aid in requirements engineering. The topics include essential of software requirements, different dimensions and good practices for requirements engineering, improving requirements processes, and risk management. It also incorporates review ofvarious activities of requirements engineering and discussion on current trends. Principles and practices of software requirements management, requirements attributes, change management process, requirements traceability matrix, links in requirements chain are also included. RM Tools, implementing requirements management automation and current trends such as aspect-oriented requirement engineering and agent-based requirement engineering are a part of the course structure.

S. No.	Course Outcomes (CO)
CO1	Understand essential software requirements components and dimensions. Apply good practices in requirements engineering, emphasizing processes and risk management.
CO2	Understand current trends in requirements elicitation, analysis models, and methods for verifying requirements.
CO3	Apply principles and practices of Software Requirements Management (RM). Utilize requirements attributes and the Change Management Process.
CO4	Analyse the importance of Requirements Traceability Matrix and links in the requirements chain. Evaluate the benefits of RM tools like Rational Requisite Pro and Caliber RM.
CO5	Evaluate commercial requirements management techniques and tools. Compare and assess the advantages and challenges of implementing requirements management automation. Understand latest trends in requirements engineering, including aspect-oriented and agent-based approaches.

Suggested Books:

1. R. Naik and S. Kishore, “Software Requirements and Estimation”, Tata McGraw Hill, 2007.
2. K. E. Weigers, “Software Requirements” Microsoft Press, 1999.
3. E. Gottesdiener, “Requirements by Collaboration: Workshops for Defining Needs”, Addison Wesley, 2002.

SWE503 Object-Oriented Software Engineering

The goal of this course is to make one design, assess and analyze object-oriented philosophy at each phase of software development life cycle. Basic concepts: Software engineering, object-orientation, object-oriented methodologies and modelling. Software development life cycles, object-oriented software life cycle models, software requirements elicitation and analysis, use case approach. Object-oriented Software Estimation: Need, Lorenz & Kidd estimation, use case points method, class point method, object-oriented function point, risk management. Object-oriented analysis, Object-Orienteddesign, Software testing.

S. No.	Course Outcomes (CO)
CO1	Understand the basics of software engineering, object-oriented paradigms, object-oriented methodologies used, and various basic terminologies.
CO2	Develop real-world software using conventional software development life cycle models, and object-oriented software development life cycle models.
CO3	Apply various techniques to gather requirements from the customers such as interviews, brainstorming session, FAST, and prototyping.
CO4	Design software requirement specification document, software design document, and test case matrix.
CO5	Design UML diagrams such as use case diagrams, class diagrams, sequence diagrams, state chart diagrams, and activity diagrams.
CO6	Analyze existing software by considering the issues of software risk management, and approaches to estimate the risk.

Suggested Books:

1. Y. Singh, R. Malhotra, "Object-Oriented Software Engineering", PHI Learning, 2012.
2. I. Jacobson, "Object Oriented Software Engineering: A Use-case Driven Approach", Pearson Education, 2009.
3. R. Laganiere & T.C. Letbridge, "Object-oriented Software Engineering: Practical Software Development Using UML and Java", Tata McGraw Hill, 2004.

SWE505 Advanced Data Structures and Algorithms

This course covers topics of Review of Elementary data structures, Sparse matrices, Advanced DataStructures: data structures for combinatorial, Operations on Disjoint Divide and Conquer approach, Graph Algorithms: Definitions and Algorithms, Greedy Method and Dynamic Programming, Dynamic Programming, Advanced Algorithms: NP Complete problems, Approximation algorithms for NP complete problem, Algorithms for matching, Flow and circular problems, Bio Inspired Algorithm-Genetic Algorithm, Particle Swam, Artificial Bee Colony, Firefly Algorithm, Bat Algorithm.

S. No.	Course Outcomes (CO)
CO1	Possess an understanding of different tree, heap and graph-based advanced data structures covered in the course.
CO2	Formulate, design and analyze algorithms and complexity analysis of the main operations of advanced data structures covered in the course
CO3	Develop and analyze the efficiency and proof of correctness of the algorithms covered in the course.
CO4	Choose appropriate data structures and algorithms, understand the abstract data types/libraries, and use them to design algorithms for a specific problem
CO5	Comprehend and gain the ability to apply and implement learned algorithm design techniques and data structures to solve problems.

Suggested Books:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to Algorithms", MIT Press, 2009.
2. R.E. Tarjan, "Data Structures and Network algorithms", SIAM Regional Conference series in Applied Mathematics, 1987.
3. Aho, Hopcraft & Ulman, "The Design and Analysis of Computer algorithms", Addison Wesley,1974.
4. S. Dasgupta, C. H. Papadimitriou, and U.V.V Azirani, "Algorithms", Tata McGraw Hill, 2017.
5. Y. Langsam, M. J. Augenstein, A. M. Tenenbaum, "Data Structures using C and C++", Pearson,2006.

SWE507 Machine Learning

The objective is to make the student understand the different supervised, unsupervised and reinforcement learning algorithms and choose the appropriate machine learning tool for different real-world examples. This course contains topics Machine Learning, Types of Machine Learning: Supervised, Unsupervised Learning, Reinforcement Learning, Categories of Supervised Learning; Predictive Modeling, Steps in Model Prediction: Metric Data Analysis, Attribute Reduction, Hypothesis Testing, Performance Evaluation Measures, Model Development, Model Validation, Model Comparison Tests, Decision Trees (ID3, C4.5, CART), Artificial Neural Networks (Single-Layer Networks, Multi-layer Perceptron), Nearest Neighbour, Computing Distance, Support Vector Machine, Dimensionality Reduction, Ensemble Learning. Research applications.

S. No.	Course Outcomes (CO)
CO1	Understand the basic concepts of machine learning, supervised, unsupervised, regression analysis, and machine learning algorithms.
CO2	Apply the learned concepts of machine learning to interpret various problems.
CO3	Analyze different mathematical machine learning models for various systems.
CO4	Evaluate the performance of the machine learning model using various performance measures.
CO5	Develop an efficient machine learning system to solve various real-time problems.

Suggested Books:

1. T. Mitchell, “Machine Learning”, McGraw Hill, 1997.
2. S. Shalev-Shwartz, S. Ben-David, “Understanding Machine Learning: From Theory to Algorithms”, Cambridge University Press, 2014.
3. J. D. Kelleher, B. M. Namee, A. D'Arcy, “Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies”, MIT Press, 2015.

SWE5401 Advanced Database Management

This course covers topics of Relational Databases Integrity Constraints, Extended ER diagram, Relational Algebra & Calculus, Functional, Multivalued and Join Dependency, Normal Forms, Rules about functional dependencies. Advanced Transaction Processing, Query Processing, QueryOptimization: Indexing and Query Optimization, Parallel and Distributed Databases: Distributed Data Storage – Fragmentation & Replication, Location and Fragment Transparency Distributed Query Processing and Optimization, Active Database and Real Time Databases: Triggers in SQL, Event Constraint and Action.

S. No.	Course Outcomes (CO)
CO1	Understand the concepts of DBMS and would have acquired skills to analyse the real-world problem domains in the context of DBMS and demonstrate the same through ER diagram.
CO2	Apply and demonstrate with understanding of relational query languages such as SQL, Relational Algebra and Relational Calculus.
CO3	Relate the concepts of inference rules, data constraints and normalization. Students would also have acquired skills to identify application of the same.
CO4	Familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing. To appraise the basic issues of Transaction processing and Serializability.
CO5	Classify various concurrency control techniques and recovery procedures
C06	Familiar with case studies regarding commercial database, Oracle platforms, Postgres and MYSQL

Suggested Books:

1. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Sixth Edition, Pearson Education, 2017.
2. Garcia-Molina, "Database Systems, The complete book", Pearson, 2014.
3. Silberschatz, H. F. Korth, S. Sudarshan, "Database System Concepts", Sixth Edition, McgrawHill, 2013.
4. C.J. Date, A. Kannan, S. Swaminathan, "An Introduction to Database Systems", Eighth Edition,Pearson Education, 2007

SWE5403 Data Warehousing and Data Mining

This course contains topics of Data Warehousing, Data Warehouse Architecture, Design, Implementation & Maintenance, Data Mining Concepts, Mining Association Rules in Large Databases, Classification and Prediction, Cluster Analysis in Data Mining, Mining Complex Typesof Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Mining Spatial Databases, Mining Multimedia Databases, Mining Time Series and Sequence Data, Mining Text Databases, Applications, Trends in Data mining, spatial mining, and Web Mining.

S. No.	Course Outcomes (CO)
CO1	Understand Data Warehouse, Need of Data Warehouse, Architecture of Data Warehouse.
CO2	Understand KDD process, applications of Data Mining.
CO3	Understand Classification and various models of Classification like KNN, Rule Based Mining
CO4	Apply Association Rule Mining.
CO5	Analyse and Evaluate various Classification Models on different problems.
CO6	Understand clustering and various types of clustering.
CO7	Apply clustering algorithms like k-means, DBSCAN etc.

Suggested Books:

1. P. Ponniah, "Data Warehousing Fundamentals", John Wiley, 2001.
2. M.H. Dunham, "Data Mining Introductory & Advanced Topics", Pearson Education, 2011.
3. H. Kamber, M. Kaufman, J. Pie, "Data Mining Concepts & Techniques", Third Edition, Morgan Kaufmann, 2012.

SWE5405 Probability and Statistics

This course contains topics of Introduction to Probability theory, Through set and Relative Frequency: Experiments and Sample Spaces, Discrete and Continuous Sample Spaces, Events, Probability and Axioms, Probability as a Relative Frequency, Joint and Conditional Probability, Random Variables, Distribution Function, Density Function, Operations on Random Variables, Sampling Distributions, Estimation, Testing of Hypotheses, Neyman-Pearson Fundamental Lemma, , Chi-square goodness of fit test and its applications, problems

.

S. No.	Course Outcomes (CO)
CO1	Understand and apply fundamental concepts of probability, Bayes' theorem, and various probability distributions.
CO2	Compute marginal and conditional distributions from joint distributions.
CO3	Compute measures of central tendency, dispersion, and apply regression and curve-fitting techniques for data analysis.
CO4	Apply methods for parameter estimation like maximum likelihood estimation.
CO5	Perform hypothesis testing using Chi-square and Neyman-Pearson tests to validate statistical data.

Suggested Books:

1. P. Peebles, "Probability random variables and random signal principles", Fourth Edition, Mc Graw Hill, 2013.
2. A. Poupolis, S. Pillai, "Probability: Random Variables and Stochastic Processes and Probability", Fourth Edition, Mc Graw Hill, 2017.
3. D. A. Lind, W. G. Marchal and S. A. Wathen, "Statistical Technics in Business and Economics", Thirteenth Edition, Tata McGraw Hill, 2007.

SWE5407 Advanced Operating System

The objective of the course is to learn various concepts related to operating systems. Operating system concepts: history, evolution and philosophy of operating systems. Concurrent processes, process coordination and synchronization, CPU scheduling, deadlocks, memory management, virtual memory, secondary storage and file management, device management, security and protection, networking, and distributed and real-time systems.

S. No.	Course Outcomes (CO)
CO1	Understanding and implementing complex process scheduling algorithms, synchronization mechanisms, and inter-process communication techniques.
CO2	Exploring the principles and challenges of distributed operating systems, including resource sharing, fault tolerance, and distributed file systems.
CO3	Examining advanced security models, access control mechanisms, and strategies for protecting system resources against various threats.
CO4	Analyzing the role of operating systems in virtualization technologies and cloud environments, focusing on resource allocation, scalability, and isolation.
CO5	Developing skills to measure, analyze, and optimize the performance of operating systems using various tools and methodologies.

Suggested Books:

1. A Silberschatz, P.B. Galvin, G. Gagne, "Operating Systems Concepts", Eighth Edition, John Wiley Publications, 2008.
2. A.S. Tanenbaum, "Modern Operating Systems", Third Edition, Pearson Education, 2007.
3. W. Stallings, "Operating Systems, Internals & Design Principles", Fifth Edition, Prentice Hall ofIndia, 2008.

SWE5409 Software Design Patterns

This course gives an overview of software architecture and the various design patterns used. Envisioning Architecture: Defining, documenting and reconstructing software architecture. CreatingArchitecture: Quality Attributes, Moving from quality to Architecture, Architectural styles and patterns, Operations, Achieving qualities, shared information systems. Analyzing Software Architecture: Analyzing development qualities at the architectural level, SAAM, ATAM, CBAM, Architecture Reviews. Moving from Architecture to Systems Software: Product Lines, Building systems from off the shelf components, Reuse of Architectural assets within an organization. Patterns Definition: Pattern categories, Pattern Description, Patterns and Software Architecture, Pattern Systems,Classification, Selection, Design Patterns Catalog Creational Pattern, Structural Pattern, BehavioralPatterns, Pattern Community, Designing a document editor. Advanced studies.

S. No.	Course Outcomes (CO)
CO1	Explain fundamental concepts of software architecture, including definition, documentation, and reconstruction.
CO2	Apply quality attributes, architectural styles, and design patterns to create effective software architectures.
CO3	Analyze and evaluate software architectures using methods like SAAM, ATAM, and CBAM.
CO4	Develop systems using reusable architectural assets and off-the-shelf components.
CO5	Classify and implement design patterns to solve software design challenges efficiently.

Suggested Books:

1. E. Gamma, R. Helm, R. Johnson, J. Vlissides, G. Booch, “Design Patterns: Elements of ReuseableObject-Oriented Software”, Addison Wesley, 1997.
2. L. Bass, P. Clements, R. Kazman, “Software Architecture in Practice”, Pearson, 2013.
3. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, “Pattern-Oriented Software Architecture, A System of Patterns”, First Edition, Wiley, 2013.

SWE5411 Agile Methods

This course is an introduction to agile methodology and how it is practiced. Introduction: Iterative development, evolutionary and adaptive development, evolutionary requirement analysis, evolutionary and adaptive planning, incremental delivery, evolutionary delivery. Agile and its significance: Classification of methods, the agile manifesto, agile project management, sustainable discipline, research evidence. Agile Methodology: Overview, lifecycle work products, roles and practices values, sample projects, adoption strategies. Case Study. Agile Practicing and Testing.

S. No.	Course Outcomes (CO)
CO1	Understand the principles of Agile methodology, including iterative and adaptive development, incremental delivery, and evolutionary requirement analysis.
CO2	Analyze the significance of Agile through its classification of methods, the Agile manifesto, and Agile project management practices, supported by research evidence.
CO3	Explore the Agile lifecycle, work products, roles, practices, values, and strategies for successful adoption in various organizational contexts.
CO4	Evaluate real-world Agile case studies to identify challenges, lessons learned, and effective solutions in Agile implementation.
CO5	Apply Agile testing techniques, including test-driven development (TDD), behavior-driven development (BDD), exploratory testing, and automation, to enhance software quality and delivery.

Suggested Books:

1. C. Larman, "Agile and Iterative Development – A Manager's Guide", Pearson Education, 2004.
2. Elisabeth Hendrickson Quality Tree Software Inc, "Agile Testing", 2008.
3. A. Cockburn, "Agile Software Development series", Safari Books, 2001.

SWE5413 Fuzzy Logic and Neural Networks

This course intends to introduce the functioning and application of fuzzy logic and neural networks.Classical & Fuzzy Sets: Properties, Operations and relations, Fuzzy sets, Membership, Uncertainty, fuzzy relations, cardinalities, membership functions. Fuzzy Logic System Components: Fuzzification, Membership value assignment, development of rule base and decision making system, Defuzzificationto crisp sets, Defuzzification methods. Introduction to Neural Networks: Biological and Artificial Neuron Models, Characteristics, McCulloch-Pitts Model. Essentials of Artificial Neural Networks: Artificial Neuron Model, Operations of Artificial Neuron, Types of Neuron Activation Function, ANN Architectures, Classification Taxonomy of ANN – Connectivity, Learning Strategy (Supervised, Unsupervised, Reinforcement), Learning Rules. Single Layer and Feed Forward Neural Networks. Associative Memories: Paradigms of Associative Memory, Pattern Mathematics, Hebbian Learning,General Concepts of Associative Memory, Bidirectional Associative Memory (BAM) Architecture, BAM Training Algorithms, Architecture of Hopfield Network. Applications.

S. No.	Course Outcomes (CO)
CO1	Understand fuzzy logic and neural network applications.
CO2	Apply fuzzy logic components in decision-making.
CO3	Describe neuron models and their operations.
CO4	Analyze activation functions and ANN architectures.
CO5	Apply single-layer and feed-forward neural networks.
CO6	Apply associative memories and Hopfield networks.

Suggested Books:

1. S. Haykins, "Neural Networks- A comprehensive foundation", Pearson Education, 1999.
2. S. Rajasekharan and G. A. Vijayalakshmi Pai, "Neural Networks, Fuzzy logic, Genetic algorithms: synthesis and applications", PHI Publication, 2003.
3. J. Yen, R. Langari, "Fuzzy Logic: Intelligence, Control and Information", First Edition, Pearson,1998.
4. J.A. Freeman, D. Skapura, "Neural Networks", Pearson Education, 2002.

SWE509 SEMINAR

SWE511 Research Paper Writing

The primary objective of the course is to make one aware of the basics and structure of formal research writing. Introduction: Concept of research writing and its importance, types of a study and its process. Systematic Literature Review: Basic concepts, planning, conducting, reporting. Research paper writing: abstract, introduction, related work, experiment design, research methods, research results, discussion & interpretation of results, validity evaluation, conclusions & future work, acknowledgement, references, index. Research ethics and misconduct.

S. No.	Course Outcomes (CO)
CO1	Understand the basic concepts of research, its importance, and its process.
CO2	Understand and apply basic concepts of how to write systematic literature review (planning, conducting and review).
CO3	Understand the various sections which should be included in a research paper.
CO4	Understand the concepts of research ethics, plagiarism, and misconduct.
CO5	Apply concepts to write research paper in the respective majoring (specialized subject) areas.

Suggested Books:

1. R. Malhotra, "Empirical Research in Software Engineering: Concepts, Analysis & Applications", CRC press, 2013.
2. M. Cargill and P. O'Connor, "Writing Scientific Research Articles: Strategy and Steps", 2nd edition, Wiley Blackwell, 2013.
3. J. Mugah, "Essentials of Scientific Writing", Author House, 2016.

SEMESTER II

SWE502 Software Testing

The goal of the course is to make one understand software testing concepts and applications. Introduction: Software Failures, Testing Process, Some Terminologies, Limitations of Testing, The V Shaped Software Life Cycle Model. Functional Testing: Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique. Essentials of Graph Theory: Graph introduction, Matrix Representation of Graphs, Paths and Independent Paths, Generation of a Graph from Program, Identification of Independent Paths. Structural Testing: Control Flow Testing, Data Flow Testing, Slice Based Testing, Mutation testing. Selection, Minimization and Prioritization of Test Cases for Regression Testing: Regression Testing, Regression Test Cases Selection, Reducing the Number of Test Cases, Risk Analysis, Code Coverage Prioritization Technique. Software Testing Activities: Levels of Testing, Debugging, Software Testing Tools. Web Based Testing: Functional Testing, User Interface Testing, Performance Testing, Configurability Testing, Database Testing, Security Testing.

S. No.	Course Outcomes (CO)
CO1	Explain basics of software testing process, limitations, and the V-shaped life cycle model.
CO2	Apply various types of testing to ensure software functionality and reliability.
CO3	Demonstrate various functional testing techniques.
CO4	Implement structural and object-oriented testing methods.
CO5	Investigate automated testing tools and various testing activities.
CO6	Perform testing of web applications and generate automated test cases.

Suggested Books:

1. Y. Singh, “Software Testing”, 1st Ed., Cambridge University Press, 2012.
2. P. C. Jorgenson, Software Testing A Craftsman’s approach, CRC Press, 1997.
3. B. Beizer, “Software Testing Techniques”, Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.

SWE504 Empirical Software Engineering

The goal of the course is to instill the concepts and applications of empirical software engineering. Introduction: What Is Empirical Software Engineering? Overview & Types of Empirical Studies, Empirical Study Process, Ethics, Importance and Basic Elements of Empirical Research, Some Terminologies. Systematic Literature Review, Software Metrics, Experimental Design, Mining Data from Software Repositories, Data Analysis and Statistical Testing, Model Development and Interpretation, Validity Threats, Reporting Results, Mining Unstructured Data, Case Study & Tools.

S. No.	Course Outcomes (CO)
CO1	Explain empirical study types and process for research in software engineering.
CO2	Apply software metrics and experimental design principles to measure software quality and conduct empirical studies.
CO3	Extract and analyze data from software repositories to derive insights using historical analysis techniques.
CO4	Develop predictive models, evaluate their performance and validate models using statistical testing.
CO5	Identify and mitigate validity threats, report findings ethically, and utilize empirical tools and text mining techniques to analyze and interpret software engineering data.

Suggested Books:

1. R. Malhotra, "Empirical Research in Software Engineering: Concepts, Analysis & Applications",CRC press, 2016.
2. B. Boehm, H. D. Rombach, M. V. Zelkowitz, "Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili", Springer, 2010.

SWE5402 Software Project Management

This course contains topics of Project Management concepts, Process Framework, Project Planning Software Life Cycle Models, Artifacts of the Project Management Process, Cost and Scheduling Estimation Models, Project Management Techniques, Project Closure, Software Project Management Renaissance, Advance Topics in Software Project Management.

S. No.	Course Outcomes (CO)
CO1	Understand basic project management concepts and software life cycle models.
CO2	Understand and apply various estimation models for cost, effort, schedule and productivity.
CO3	Analyse various project organizations & responsibilities and Project Tracking and Control Defect Tracking Concepts
CO4	Understand the importance of project closure analysis and apply it.
CO5	Analyse the evolution of software economics from conventional to modern practices.
CO6	Understand the upcoming trends in software project management and explore modern practices.

Suggested Books:

1. W. S. Humphrey, "Managing the Software Process", Pearson Education, 1990.
2. W. Royce, "Software Project Management", Pearson Education, 2002.
3. P. Jalote, "Software Project Management in Practice", Pearson Education, 2002.
4. B. Hughes, "Software Project Management", Tata McGraw Hill, 1995.

SWE5404 Distributed Systems

This course covers topics of Distributed System Models, Transparency, Scalability, Inter- process Communication, Middleware, issues in design of Distributed systems: current & future, Communications, Process and Synchronization, Serializability, Resource Allocation, Distributed Shared Memory, Process Scheduling, Load Balancing & Load Sharing, Mutual Exclusion, Election algorithms. Distributed File Systems Overview of security techniques, Cryptographic algorithms , Digital signatures, Cryptography pragmatics.

S. No.	Course Outcomes (CO)
CO1	Explain fundamental models of distributed systems, including transparency, scalability, and inter-process communication.
CO2	Evaluate middleware solutions and key design challenges in distributed systems, both current and emerging.
CO3	Apply synchronization techniques, serializability principles, and process coordination strategies in distributed environments.
CO4	Demonstrate efficient resource allocation, distributed shared memory management, process scheduling, and load balancing techniques.
CO5	Utilize security mechanisms such as cryptographic algorithms, digital signatures, and cryptography pragmatics to enhance system security.

Suggested Books:

1. S. Taenbaum, M. V. Steen, "Distributed Systems: Principles and Paradigms", Prentice Hall,2015.
2. G. Coulouris, J. Dollimore, T. Kindberg, "Distributed Systems Concepts and Design", Addison Wesley, 1994.
3. A. Kshenkalyani, M.Singhal, "Distributed Computing", Cambridge University Press, 2008.

SWE5406 Fundamentals of Information Retrieval

This course contains topics of Introduction and Search engine architecture, Search engine architecture, Retrieval models, Retrieval evaluation, classical evaluation metrics, e.g., Mean Average Precision, and modern advance, e.g., interleaving. Relevance feedback, Link analysis and Search applications, recommendation, personalization, and online advertising.

S. No.	Course Outcomes (CO)
CO1	Understand the fundamentals of Boolean retrieval, term vocabularies, postings lists, and techniques for dictionary-based and tolerant retrieval, as well as index construction and compression.
CO2	Apply scoring methods, term weighting, vector space models, probabilistic models, and language models to compute and rank search results effectively in an information retrieval system.
CO3	Evaluate information retrieval systems using classical metrics like Mean Average Precision and modern techniques such as interleaving, along with relevance feedback and query expansion.
CO4	Develop skills in text classification using Naive Bayes, vector space methods, support vector machines, clustering methods, and matrix decompositions for advanced document analysis.
CO5	Explore web search fundamentals, including web crawling, link analysis, XML retrieval, recommendation systems, personalization techniques, and strategies for online advertising.

Suggested Books:

1. C. D. Manning, P. Raghavan and H. Schutze. “Introduction to Information Retrieval”, Cambridge University Press, 2008.

SWE5408 Artificial Intelligence

This course covers topics of AI Problems, Task Domains of AI, AI Techniques, Basic Problem solvingMethod: state space search, problem characteristics, Heuristic search Techniques, Knowledge Representation Knowledge Representation: using Predicate Logic: Unification, resolution. Natural deduction, using Rules, Structured Knowledge Representation, Programming Languages: Prolog or Lisp , Symbolic Reasoning under uncertainty, Statistical Reasoning, Concept of learning, learningin problem solving, learning by inductions, genetic algorithm, , Neural Network, Genetic theorem, Expert Systems Research issues in different domains.

S. No.	Course Outcomes (CO)
CO1	Understand AI problems, task domains, and problem-solving methods.
CO2	Apply predicate logic for knowledge representation.
CO3	Demonstrate symbolic reasoning and structured knowledge in languages like Prolog.
CO4	Explain statistical reasoning, learning, and genetic algorithms.
CO5	Analyze neural networks and expert system research.

Suggested Books:

1. J E. Rich. K. Knight, "Artificial Intelligence", Tata McGraw Hill, Second Edition, 1992.
2. N.J. Nilsson, "Principles of AI", Narosa Publ. House, 1990.
3. D.W. Patterson, "Introduction to AI and Expert Systems", PHI, 1992.
4. M. Negnevitsky, "Artificial Intelligence: A Guide to Intelligent Systems", Second Edition, Addison-Wesley, 2005.

SWE5412 Multimedia Applications

This course contains topics of Introduction to Multimedia Systems Architecture and Components, Multimedia Distributed Processing Model, Synchronization, Orchestration and Quality of Service Architecture. Usage of Text in Multimedia, Families and Faces of Fonts, Outline Fonts, Bitmap Fonts International Character Sets and Hypertext, Digital Fonts Techniques. Audio and Speech, Images and , Multimedia and Hypermedia, Hypermedia Presentation.

S. No.	Course Outcomes (CO)
CO1	Explain fundamental concepts of multimedia systems, including architecture, components, and distributed processing models.
CO2	Demonstrate knowledge of multimedia elements such as text, sound, digital audio, video capture, and their integration in multimedia applications.
CO3	Implement various data compression algorithms to optimize multimedia storage and transmission.
CO4	Compare different speech, image, and video compression techniques, including JPEG, MPEG, and speech synthesis, for efficient multimedia representation and transmission.
CO5	Demonstrate various tools and technologies for solving real-world problems.

Suggested Books:

1. T. Vaughan, “Multimedia: Making it work”, Tata McGraw-Hill, Ninth Edition, 2017.
2. R. Aggarwal, B. B Tiwari, “Multimedia Systems”, Excel Publication, 2007.
3. Z. Li & M.S. Drew, “Fundamentals of Multimedia”, Pearson Education, 2009.
4. D. Hillman, “Multimedia Technology and Application”, Galgotia Publication, 2000.

SWE5414 Deep Learning

Introduction: Introduction to K-Nearest Neighbors, Deep Feed forward Networks, Regularization of deep learning, Optimization for training deep models, convolutional networks, recurrent networks, Applications. **Convolutional Neural Networks:** Invariance, stability Properties of CNN representations: invertibility, stability, invariance. Variants of the Basic Convolution Function history of CNN and deep learning. **Recurrent and Recursive Nets:** Recurrent and Recursive Nets, Unfolding Computational Graphs, Recurrent Neural Network, Bidirectional RNNs, Deep Recurrent Networks, Recursive Neural Networks, The Challenge of Long-Term Dependencies, Echo State Networks, Leaky Units and Other Strategies for Multiple, Time Scales, The Long Short-Term Memory and Other Gated RNNs, optimization for Long-Term Dependencies, Explicit Memory. **Linear Factor Model and Autoencoder:** Linear Factor Models, Probabilistic PCA and Factor Analysis Independent Component Analysis (ICA), Slow Feature Analysis, Sparse Coding, Autoencoders, Undercomplete Autoencoders, Regularized Autoencoders, Stochastic Encoders and Decoders, Applications of Autoencoders. **Deep Supervised Learning:** Introduction to Deep Supervised Learning, Convolution & Pooling, Dropout, Transfer Learning Transfer Learning Scenarios, Applications of Transfer Learning, transfer Learning Methods, Fine Tuning and Data Augmentation, Related Research Areas.

S. No.	Course Outcomes (CO)
CO1	Understand Deep Learning and various applications of Deep Learning
CO2	Understand, apply and evaluate performance of CNN for image classification.
CO3	Understand and apply various Object Detection algorithms like Sliding Window Protocol, RCNN, Faster RCNN, YOLO etc.
CO4	Understand and Apply BERT algorithm
CO5	Understand and apply Text processing deep learning models like RNN, LSTM, GRU

Suggested Books:

1. Goodfellow, Y. Bengio and A.Courville, "Deep Learning", MIT Press, 2015.
2. A. Gibson and J Patterson, "Deep Learning", O'Reilly Media, Inc., 2017.
3. Russell, Norvig, "Artificial Intelligence: A Modern Approach", 3rd edition, Prentice Hall, 2010.
4. R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification", Wiley, 1973.
5. A. M. Bishop, "Neural Networks for Pattern Recognition". Oxford University Press, 1995.

SWE5416 Soft Computing

This course contains topics of Soft Computing & Artificial Intelligence, Applications of Soft Computing, AI Search Algorithm, Predicate Calculus, Rules of Interference, Semantic Networks, Frames, Objects, Hybrid Models. Artificial Neural Networks, Back propagation Networks, Applications of NN.Fuzzy Logic and Fuzzy Sets , Fuzzy Arithmetic, Neuro - Fuzzy Modeling , Genetic Algorithms and Swarm Optimizations: Fitness Computations, Evolutionary Programming, Genetic Programming Parse Trees, Variants of GA, Applications.

S. No.	Course Outcomes (CO)
CO1	Understand soft computing concepts and AI evolution towards computational intelligence.
CO2	Apply soft computing techniques and AI models in problem-solving.
CO3	Analyze ANN architecture, backpropagation, and their applications.
CO4	Explain fuzzy logic, fuzzy arithmetic, and neuro-fuzzy modeling techniques.
CO5	Apply genetic algorithms and swarm optimization in problem-solving.

Suggested Books:

1. S. Patnaik, B. Zhong, "Soft Computing Techniques in Engineering Applications", Springer, 2014.
2. H. J. Krogh, R.G. Palmer, "Introduction to the Theory of Neural Computation", Addison-Wesley,1991.
3. M. Mitchell, "An Introduction to Genetic Algorithm", PHI, 1998.
4. S. Kaushik, "Artificial Intelligence", Cengage Learning, 2007.
5. J. A. Anderson, "An Introduction to Neural Networks", MIT Press, 1997.
6. G.J. Klir & B. Yuan, "Fuzzy Sets & Fuzzy Logic", PHI, 1996.

SWE5418 Software Reliability

This course gives a detailed introduction to software reliability concepts, how it can be measured and improved. Introduction to System Reliability: Reliability mathematics, probability distributions, system reliability, maintainability and availability, designing for higher reliability, redundancy. System Reliability Concepts: Software and hardware reliability, basic concepts, reliability model classification, software reliability growth models, markovian models. Non-homogeneous poisson process models: NHPP models, Musa models, Okumoto model, Yamada delayed S-shaped model, Imperfect debugging models, Kapur- Garg model, Subburaj-Gopal model for the learning phenomenon, Subburaj-Gopap- Kapur versatile debugging model. Comparison of software reliability models. Advanced topics in software reliability.

S. No.	Course Outcomes (CO)
CO1	To learn reliability mathematics to examine software reliability of a system
CO2	To understand system reliability concepts and terminologies.
CO3	To analyze and compare and evaluate Software Reliability Growth Models.
CO4	To design, select and apply appropriate ML algorithm to solve computer engineering problems.
CO5	To implement procedures for testcase generation, preparing test, executing test and developing operational profile for a system.

Suggested Books:

1. J. D. Musa, A. Iannino, K. Okumoto, "Software Reliability – Measurement, Prediction, Application,Series in Software Engineering and Technology", McGraw Hill., 1987.
2. M. Lyu, "Handbook of Software Reliability Engineering", IEEE Computer Society Press, 1996.
3. J. D. Musa, "Software Reliability Engineering", Tata McGraw Hill, 1999.
4. P. D. T. O'Connor, "Practical Reliability Engineering", Fourth Edition, John Wesley & sons, 2003.

SWE5420 Natural Language Processing

This course contains topics Phases in natural language processing, applications. Words and Word Forms, Morphology Paradigms; Finite State Machine Based Morphology; Automatic Morphology Learning; Shallow Parsing; Named Entities; Maximum Entropy Models; Random Fields. Morphology, acquisition models, Theories of Parsing, Parsing Algorithms; Robust and Scalable Parsing onNoisy Text as in Web documents; Lexical Knowledge Networks, Wordnet Theory; Indian Language Wordnets and Multilingual Dictionaries; Semantic Roles; Word Sense Disambiguation; WSD and Multilinguality, Web 2.0 Applications.

S. No.	Course Outcomes (CO)
CO1	Extract information from text automatically using concepts and methods from natural language processing (NLP) including stemming, n-grams, POS tagging, and parsing.
CO2	Develop speech-based applications that use speech analysis (phonetics, speech recognition, and synthesis)
CO3	Analyze the syntax, semantics, and pragmatics of a statement written in a natural language.
CO4	Apply machine learning algorithms to natural language processing. CO5: Evaluate the performance of NLP tools and systems.
CO5	Evaluate WSD techniques and NLP applications in web environments like blogs and social media.

Suggested Books:

1. D. Jurafsky, J. H. Martin, "Speech and Language Processing", Second Edition, Pearson Education,2009.
2. A. James, "Natural Language Understanding", Second Edition, Pearson Education, 1994.
3. Bharati, R. Sangal V. Chaitanya, "Natural Language Processing: A Paninian Perspective", PHI,2000.
4. T. Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", OUP, 2008.

SWE5422 Computer Vision

This course introduces the theoretical and practical aspects of computer vision, covering both classical and state of the art deep-learning based approaches. This course covers everything from the basics of the image formation process in digital cameras and biological systems, through a mathematical and practical treatment of basic image processing, space/frequency representations, classical computer vision techniques for making 3-D measurements from images, and modern deep-learning based techniques for image classification and recognition.

S. No.	Course Outcomes (CO)
CO1	Understand the fundamentals of visual perception, image formation, and image representation, including the roles of pixels and color in digital and biological systems.
CO2	Apply image processing techniques such as filtering, edge detection, frequency domain analysis, and feature/keypoint extraction to analyze and process visual data.
CO3	Explore classical computer vision concepts, including camera calibration, depth estimation, motion analysis, stereopsis, structure from motion (SfM), and SLAM.
CO4	Develop proficiency in deep learning techniques for vision tasks such as image classification, object detection, segmentation, and transfer learning while addressing model training challenges.
CO5	Investigate advanced topics like generative models, augmented reality, computational photography, and ethical considerations, with an emphasis on emerging trends in computer vision.

Suggested Books:

1. R. Szeliski, “Computer Vision: Algorithms and Applications”, Second Edition, Springer Nature Switzerland, 2022.
2. A. Torralba, P. Isola, W. T. Freeman, “Foundations of Computer Vision (Adaptive Computation and Machine Learning series)”, The MIT Press, 2024.
3. R. Shanmugamani, “Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras”, Packt Publishing, 2018.

SWE5424 Optimization Techniques

This course contains topics of Introduction to Linear Programming, Graphical method, The Simplex Method, Post optimality Analysis, Duality Theory and Sensitivity Analysis, Dynamic Programming, Integer Programming, Nonlinear Programming, Queuing Theory.

S. No.	Course Outcomes (CO)
CO1	Apply linear programming concepts and solve LPP using graphical methods and the Simplex algorithm.
CO2	Analyze post-optimality, duality theory, and sensitivity analysis in linear programming problems.
CO3	Apply dynamic programming principles and solve integer programming problems using branch and bound methods.
CO4	Explain nonlinear programming methods and apply queuing models in resource management and service systems.
CO5	Apply genetic algorithms for unconstrained and constrained optimization problems.

Suggested Books:

1. H.A. Taha, "Operations Research", Ninth Edition, Pearson Education, 2011.
2. J. K. Sharma, "Operations Research", Third Edition, Mcmillan, 2007.
3. F. Hiller & J. Lieberman, "Operations Research", Eighth Edition, Tata McGraw Hill, 2005.
4. K. Swarup, P. K. Gupta, M. Mohan, "Operations Research", Sultan Chand & Sons, 2017.

SWE5426 Information Theory and Coding

This course is an introduction to various information theory concepts and applications. Introduction to Probability, Elementary Theorems, Random Variable, Uncertainty and Information, Shannon Entropy,Joint and conditional Entropies Mutual Information, Uniquely decipherable and Instantaneous codes, Noiseless coding problem. Source coding Theorem, Block coding, construction of Optimal codes, Huffman's & Shannon – Fano methods. Discrete memory less channel, channel capacity BSC and other channels. Information measure for continuous ensembles capacity of AWGN channel. Error control coding. The channel coding Theorem, Application to BSC , Source Coding with fidelity criteria. Types of codes, error and error control strategies, Linear block codes, syndrome and error detection, Minimum distance, Error detecting and correcting capabilities of a block code, Syndrome decoding, Hamming codes. Cyclic codes, Generator and parity, Latest Research.

S. No.	**Course Outcomes (CO)**
CO1	Introduce the principles and applications of information theory.
CO2	Comprehend various communication channel and error control code properties.
CO3	Apply linear block codes for error detection and correction.
CO4	Apply cyclic codes and parity generator for performance analysis & cyclic codes for error detection and correction.
CO5	Apply information theory and coding concepts to solved real-world problems.

Suggested Books:

1. R. B. Ash, "Information Theory", Dover Science Publications, 1965.
2. T. M. Cover, J.A. Thomas, "Elements of Information Theory", John Wiley & Sons, 2006.
3. S. Lin, D. J. Costello Jr, "Error Control coding: Fundamental & Application", Prentice Hall, 1983.
4. C. E. Shannon, W. Weaver, "A Mathematical Theory of Communication", University of IllinoisPress, 1998.

SWE5428 Big Data Analytics

The objective is to get students exposed with the basic rudiments of business intelligence system, understand the modeling aspects behind business intelligence and understand the business intelligence life cycle and the techniques used in it. This course contains topics of Evolution of Big data, Big data characteristics, Big Data Use Cases, Advanced Analytical Theory and Methods, K-means, Decision Tree, Advanced Analytical Theory and Methods-Association Rules, Naïve Bayes, Bayes' Theorem, Naïve Bayes Classifier. Stream Memory, NoSQL Data Management for Big Data And Visualization, Review of Basic Data Analytic Methods using R.

S. No.	Course Outcomes (CO)
CO1	Understand the definition, characteristics, and challenges of Big Data, and explore Big Data technologies and applications.
CO2	Apply clustering techniques like K-means and evaluate methods for determining the number of clusters.
CO3	Analyze decision tree algorithms and Naïve Bayes classifiers for data classification.
CO4	Explain NoSQL databases and their role in Big Data management, including key-value, document, and graph stores.
CO5	Apply data analysis techniques to Big Data using R, with focus on platforms like Twitter and e-commerce.

Suggested Books:

1. P.C. Zikopoulos, C. Eaton, D. Deroos, T. Deutsch, J. Lapis, "Understanding Big Data", McGraw Hill, 2012.
2. J. Leskovec, A. Rajaraman, J. D. Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.
3. B. Lublinsky, K.T. Smith, A. Yakubovich, "Professional Hadoop Solutions", Wiley, 2015.
4. T. White, "HADOOP: The definitive Guide", O Reilly, 2012.

UEC Research Methodology

SWE506 Software Engineering Projects and Tools

Projects based on software engineering concept with practical implementation: Design, specification, coding, and testing of a significant team programming project. The demonstration of automated software engineering tools and the tools will be used for software projects.

SEMESTER III

SWE601 Software Quality & Metrics

The primary objective of the course is to make one understand software quality concepts and associated metrics to deliver good quality maintainable software. Introduction to software quality: What is software quality? software quality attributes, elements of a quality system, software quality models. Software metrics, their categories and application areas, measurement scales, analyzing metric data, metrics for measuring size, structure and software quality. Software maintenance: categories, challenges, maintenance of object-oriented software, software rejuvenation, estimation of maintenance effort, configuration management, regression testing. Case study pertaining to software quality improvement.

S. No.	Course Outcomes (CO)
CO1	Understand basic concepts of software quality, the components of SQA, and SQA plan.
CO2	Analyze various software quality models and assess their impact on software quality.
CO3	Demonstrate internal and external product attributes using software metrics, including size, complexity, and modularity.
CO4	Assess software quality at different levels (product, process, and maintenance) using appropriate quality metrics and methodologies, including object-oriented metrics.
CO5	Demonstrate quality estimation tools and evaluate computer aided quality engineering technique for quality assurance.

Suggested Books:

1. Y. Singh, R. Malhotra, “Object-Oriented Software Engineering”, PHI Learning, 2012.
2. S.H. Kan, “Metrics and Models in Software Engineering”, Second Edition, Pearson Education,2003.
3. A. Basu, “Software Quality Assurance Testing and Metrics”, PHI Learning, 2015.

SWE6401 Pattern Recognition

This course contains topics of Pattern Recognition, Feature Detection, Classification, Review of Probability Theory, Conditional Probability and Bayes Rule, Decision Theory, Sufficient Statistics, Template-based Recognition, Feature Extraction, Eigenvector and Multilinear Analysis, Training Methods, Maximum Likelihood and Bayesian Parameter Estimation, Linear Discriminant/Perceptron Learning, Optimization by Gradient Descent. Support Vector Machines, K-Nearest-Neighbor Classification, Non-parametric Classification, Unsupervised Learning Algorithms, Linear Dynamical Systems, Kalman Filtering, Bayesian Networks.

S. No.	Course Outcomes (CO)
CO1	Understand fundamental principles, and methodologies of pattern recognition, including real-world applications and models.
CO2	Implement bayesian classifier, discriminant functions, to address issues like missing and noisy features using Bayesian networks.
CO3	Utilize Maximum Likelihood and Bayesian parameter estimation methods, including PCA, Fisher Discriminant Analysis, and Expectation-Maximization for dimensionality reduction.
CO4	Develop models using hidden markov models, dynamic bayesian networks, perceptron, and other non-parametric density estimation techniques.
CO5	Apply clustering techniques like K-means, Mixture Modeling, Hidden Markov Models, and Kalman Filtering for pattern recognition tasks.

Suggested Books:

1. R. O. Duda, P. Hart, D. Stork, "Pattern Classification", Second Edition, Wiley, 2000.
2. C.M. Bishop,"Pattern Recognition and Machine Learning", Springer, 2007.
3. C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
4. S. Theodoridis, K. Koutroumbas, "Pattern Recognition", Fourth Edition, Academic Press, 2008.

SWE6403 Mobile Computing

This course covers topics of Network Technologies and Cellular Communications, Discussion on Bluetooth & GSM. Introduction to Mobile Computing, Medium Access Control, Mobile Architecture, Mobile Network Layer, Mobile Transport Layer, Mobile Ad hoc Networks (MANETs), Wireless Sensor Networks, Protocols and Tools: Wireless Application Protocol WAP. (Introduction, protocol architecture, and treatment of protocols of all layers), J2ME and latest technologies.

S. No.	Course Outcomes (CO)
CO1	Describe basics of mobile computing architecture, application development, and security considerations.
CO2	Analyze Cellular Technologies & Wireless Communication.
CO3	Implement wireless application protocols (WAP, MMS), WLAN, and intelligent networking concepts.
CO4	Design and develop mobile applications using platforms such as Palm OS, Symbian OS, Windows CE, and J2ME.
CO5	Assess security risks, encryption techniques, and security frameworks for mobile computing.

Suggested Books:

1. R. Kamal, "Mobile Computing", Second Edition, Oxford Higher Education, 2002.
2. W. Stallings, "Wireless Communication and Networks", Pearson Education, 2003.
3. V. Garg, J. Wilkes, "Wireless and Personal Communications Systems", Prentice-Hall, 1996.
4. M. LotherMerk, S. Nicklaus, T. Stober, "Principle of Mobile Computing", Second Edition, Springer,2003.

SWE6405 Cloud Computing

The goal of this course is to introduce the concepts and applications of cloud computing. Overview of Computing Paradigm and introduction to cloud computing: Recent trends in computing, Evolution of Cloud Computing, Cloud service providers, Properties, Characteristics & Disadvantages, Cloud computing vs. Cluster computing vs. Grid computing, Role of Open Standards. Cloud computing architecture, Role of networks and web services in cloud computing, Service models, Deployment Models. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). Service management in Cloud Computing, Cloud Security.

S. No.	Course Outcomes (CO)
CO1	Explain the fundamental concepts, evolution, and recent trends in cloud computing, differentiating it from other computing paradigms like cluster and grid computing.
CO2	Evaluate cloud computing architecture, including network roles, web services, service models (IaaS, PaaS, SaaS), and deployment models.
CO3	Identify major cloud service providers, understand cloud service properties, and assess service management techniques.
CO4	Apply knowledge of cloud computing technologies in designing and deploying cloud-based applications.
CO5	Assess security challenges in cloud computing and implement cryptographic and security mechanisms to protect cloud-based services.

Suggested Books:

1. B. Sosinsky, "Cloud Computing Bible", Wiley, 2010.
2. R. Buyya, J. Broberg, A. M. Goscinski, "Cloud Computing: Principles and Paradigms", Wiley, 2011.
3. N. Antonopoulos, L. Gillam, "Cloud Computing: Principles, Systems and Applications", Springer,2012.
4. R. L. Krutz, R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley, 2010.

SWE6407 Cluster and Grid Computing

This course contains topics of Cluster and Grid computing, Meta-computing, Web services and Grid Computing, e-Governance and the Grid, OGSA and WSRF, Technologies and Architectures for Grid Computing- Issues, Functional requirements, Web Services and the Service Oriented Architecture, Globus Toolkit, GT4 Architecture, GT4 Containers, The Grid and Databases, Cluster Computing, Cluster Middleware, Networking, Protocols and I/O for clusters, Setting Up and Administering a Cluster, Cluster Technology for High Availability, Process Scheduling, Load Sharing and Load Balancing.

S. No.	Course Outcomes (CO)
CO1	Understand basics concepts of cluster and grid computing, web services, and web oriented architecture.
CO2	Explain OGSA, WSRF, and Globus Toolkit, and analyze their role in Grid computing, security, and database integration.
CO3	Compare parallel computing approaches and evaluate cluster architecture, middleware, and resource management.
CO4	Demonstrate cluster setup and administration, and apply fault tolerance for high availability.
CO5	Design job scheduling and load balancing strategies.

Suggested Books:

1. W. Gropp, E. Lusk, T. Sterling, "Beowulf Cluster Computing with Linux", Second edition, MITPress, 2003.
2. B. Jacob, M. Brown, K. Fukul, N. Trivedi, "Introduction to grid computing", IBM, 2005.
3. G. F. Pfister, "In Search of Clusters: The ongoing battle in lowly parallel computing", Second Edition, PHI, 1998.
4. C.S.R. Prabhu, "Grid and Cluster Computing", PHI, 2008.
5. R. Buyya, "High Performance Cluster Computing: Architectures and Systems", Volume 1, Pearson Education, 2008.

SWE6409 Swarm and Evolutionary Computing

The course is designed to introduce the basic concepts of evolutionary and swarm computing alongwith their applications. Introduction to Evolutionary Computing: Components, global optimization, evolution strategies, fitness functions, learning classifier systems, parameter control, multi-modal problems. Swarm Intelligence: Its application to optimization problems, particle swarm optimization. Genetic Algorithm: Basics, reproduction, cross-over and mutation, Genetic algorithm convergence, Genetic programming. Hybrid Methods and Multi-objective Evolutionary Algorithms: Variants of Particle Swarm optimization and Genetic Algorithm, their hybridization, hybrid Multi-objective Optimization algorithms. Other recent algorithms: Cuckoo search algorithm, Artificial Bee Colony Optimization, Ant Colony Optimization, Fire-fly algorithm, Bacterial Foraging, Application to the travelling salesman problem. Application to real world optimization problems.

S. No.	Course Outcomes (CO)
CO1	Understand the components and techniques of evolutionary computing, including global optimization and fitness functions.
CO2	Apply swarm intelligence methods like particle swarm optimization and genetic algorithms to optimization problems.
CO3	Analyze hybridization techniques and multi-objective optimization algorithms using evolutionary computing methods.
CO4	Evaluate algorithms like cuckoo search, artificial bee colony, and ant colony optimization for optimization tasks.
CO5	Apply evolutionary computing algorithms to real-world optimization problems like the traveling salesman problem.

Suggested Books:

1. P. Engelbrech, “Computational Intelligence”, Second Edition, John Wiley & Sons, 2008.
2. M. Mitchell, “An Introduction to Genetic Algorithm”, MIT Press, 1996.
3. D. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley, 1989.
4. A.E Eiben, J.E. Smith, “Introduction to Evolutionary Computing”, Second Edition, Springer, 2007.
5. K. DeJong, “Evolutionary Computation: A Unified Approach”, MIT Press, 2006.0

SWE6411 Internet of Things

Introduction to Internet of Things: Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs IoT enabaled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle. IoT and M2M: Software defined networks, network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCOZF, YANG- NETCONF, YANG, SNMP NETOPEER. Introduction to Python: Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling Python packages – JSON, XML, HTTP Lib, URL Lib, SMTPLib. IoT Physical Devices and Endpoints: Introduction to Raspberry PI-Interfaces (serial, SPI, I2C) Programming – Python program with Raspberry PI with focus of interfacing external gadgets, controlling output, and reading input from pins. IoT Physical Servers and Cloud Offerings: Introduction to Cloud Storage models and communication APIs Webserver – Web server for IoT, Cloud for IoT, Python web application framework designing a RESTful web API.

S. No.	Course Outcomes (CO)
CO1	Interpret the impact and challenges posed by IoT networks leading to new architectural models
CO2	Illustrate the smart objects and the technologies to connect them to network
CO3	Compare different Application protocols for IoT.
CO4	Infer the role of Data Analytics and Security in IoT.
CO5	Identify sensor technologies for sensing real world entities and understand the role of IoT in various domains of Industry.

Suggested Books:

1. Bahga and V. Madisetti, “Internet of Things – A Hands-on Approach”, Universities Press, 2015.
2. M. Richardson and S. Wallace, Getting Started with Raspberry Pi, O’Reilly (SPD), 2014.
3. J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos and D. Boyle, “From Machine-to Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st edition, Academic Press, 2014.
4. B.S. Reiter and F.Michahelles, “Architecting the Internet of Things”, Springer, 2011.
5. W. Stallings, “Foundations of modern networking: SDN, NFV, QOE, IOT, and cloud” publisher: Addison-Wesley, 2015.

SWE6413 Software Maintainability

Fundamentals of Software Maintenance: Meaning of software maintenance, software change, ongoing support, economic implications of modifying software, the nomenclature and image problem, software maintenance framework, potential solutions to maintenance problem. Maintenance process models: Definition of critical appraisal of traditional process models, maintenance process models. Program understanding: Aims of program comprehension, maintainers and their information needs comprehension process models, mental models, program comprehension strategies, factors that affect understanding, implications of comprehension theories and studies. **Reverse Engineering**: Definition, purposes and objectives, levels of reverse Engineering, supports techniques, benefits. Reuse and reusability: Definitions, objective and benefit of reuse, approach to reuse, domain. Analysis, COMPONENTS engineering, reuse process model, factors that impact upon reuse. Maintenance measures, Definitions, objectives of software measurement, example measures, guidelines for selecting maintenance measures. **Configuration Management:** Definition for configuration management, change control, documentation. Management and organizational issues, Management responsibilities, enhancing maintenance productivity, maintenance teams, personnel Education and Training, organization modes. **Building and Sustaining Maintainability:** Quality Assurance, fourth generation languages, object oriented paradigms. Maintenance tools: Criteria for selecting tools, taxonomy of tools, program understanding and reverse engineering testing, configuration management, and other tasks. Past, present and future of software maintenance. **Software Administration**: Analyzing system logs, operating system updates, patches, and configuration changes, Performing backups. Installing and configuring new hardware and software. Adding, removing, or updating user account information, resetting passwords, System performance tuning. Performing routine audits of systems and software.

S. No.	Course Outcomes (CO)
CO1	Develop a comprehensive understanding of software maintenance concepts, including maintenance frameworks, economic implications, and challenges like nomenclature and image problems.
CO2	Analyze and evaluate traditional and modern maintenance models, decision-making processes, and program comprehension strategies to support maintainers effectively.
CO3	Gain proficiency in reverse engineering, reengineering, and reuse approaches, emphasizing domain analysis, component engineering, and their applications in software maintenance.
CO4	Apply software measurement techniques and configuration management practices to ensure maintainability, quality, and effective change control in software systems.
CO5	Explore, select, and utilize advanced maintenance tools and modern trends, including object-oriented paradigms and future research directions, to enhance software maintainability and address real-world problems.

Suggested Books:

1. A. A. Takang and P. A. Grubb, “Software Maintenance: concepts and Practice”, International Thomson Computer press, London.

2. R. S. Pressman, “Software Engineering”, 6th edition, Tata McGraw-Hill, 2004.

SWE603 Minor Project/ Research Thesis/ Patent

SEMESTER IV

SWE602 Major Project/ Research Thesis/ Patent