Professional Electives (7th&8th Sem CSE/CE/IT)

CS 129	ArtificialIntelligence
Credits:3 (3-0-0)	
Pre-requisiteforthis course:	○ StrongknowledgeofMathematics. ○ Goodcommandoverprogramminglanguages. ○ GoodAnalytical Skills. ○ Abilitytounderstandcomplexalgorithms. ○ BasicknowledgeofStatisticsandmodeling
CourseObjective:	Thecourseaimsto: ○ Presentation of artificial intelligence as a coherent body of ideas and methodstoacquaintthestudentwiththebasicprogramsinthefieldand their underlying theory. ○ Studentswillexplorethisthroughproblem-solvingparadigms,logicand theoremproving,languageandimageunderstanding,searchandcontrol methods and learning. .
CourseOutcome:	Oncompletionofthecourse,astudentshouldbeableto: ○ Usevarioussymbolicknowledgerepresentationstospecifydomainsand reasoning tasks of a situated software agent. ○ Use different logical systems for inference over formal domain representations.andtracehowaparticularinferencealgorithmworksona given problem specification. ○ Understandtheconceptualandcomputationaltrade-offsbetweenthe expressiveness of different formal representations. ○ Transferableskills:Uponcompletion,studentswillbeableto: ○ Usekeylogic-basedtechniquesinavarietyofresearch settings. ○ Communicatescientificknowledgeatdifferentlevelsof abstraction.
Syllabus	
Module1 [12] Hours	WhatisArtificialIntelligence?AITechnique,LeveloftheModel,Problem SpacesandSearch:DefiningtheProblemasaStateSpaceSearch,Production

	Systems, Problem Characteristics, Production System Characteristics, Issues in the Design of Search Programs. Heuristic Search Techniques: Generate-and-Test, Hill Climbing, Best-first Search, Problem Reduction, Constraint Satisfaction, Means-ends Analysis, Knowledge Representation: Representations andMappings, Approaches to Knowledge Representation, Using Predicate Logic: Representing Simple Facts in Logic, Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution, Natural Deduction. Using Rules: Procedural Versus Declarative Knowledge, Logic Programming, Forward Versus Backward Reasoning, Matching, Control Knowledge. Symbolic Reasoning Under Uncertainty: Introduction to Nonmonotonic Reasoning, Logics for Nonmonotonic Reasoning, Implementation Issues, Augmenting a Problem-solver, Depth-first Search, Breadth first Search, Weakand Strong Slot-and-Filler Structures: Semantic Nets, Frames, Conceptual Dependency Scripts, CYC.
ModuleII [10] Hours	Game Playing: The Minimax Search Procedure, Adding Alpha-beta Cutoff Iterative Deepening, Planning: The Blocks World, Components of a Planning System, Goal Stack Planning, Nonlinear Planning Using Constraint Posting, Hierarchical Planning Other Planning Techniques, Understanding: What is Understanding, What Makes Understanding Hard?Understanding as Constraint Satisfaction.
Module III [8] Hours	Learning: Rote Learning, Learning by Taking Advice, Learning in Problem-solving, Learning from Examples: Induction, Explanation-based Learning, Discovery, Analogy, Formal Learning Theory, Neural Net Learning and Genetic Learning. Expert Systems: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.
SuggestedBooks: [Minimum 4/5 Books] (IncludeE-books also, if any)	1. StuartRussellandPeterNorvig.ArtificialIntelligence:AModern Approach, Prentice-Hall (fourth Edition, 2020). 2. ElaineRich.KevinKnight.&ShivashankarBNair.ArtificialIntelligence, McGraw Hill, 3rd ed.,2009 3. IntroductiontoArtificialIntelligence&ExpertSystems.DanW Patterson, PHI.,2010 4. SKaushik.ArtificialIntelligence.CengageLearning.1sted.2011
Evaluation:	1. Quizzes:15% 2. MidTerm: 30% 3. EndTermExam: 50% 4. Teacher`sAssessment:5%

CSXXX	Computer Vision (CV)
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Data structures • Linear algebra and vector calculus • Machine learning fundamentals • Programming knowledge (C++ or Python) • Opensource libraries (computer vision libraries, image processing libraries)
Course Objective:	The course objectives: • Introduction of the image fundamentals • Developing mathematical intuition and foundation of computer vision principles • Applications of machine learning in computer vision to solve real-world problems
Course Outcome:	On completion of the course, a student should be able to: • Detailed understanding of image filtering, transformations, geometric and physical models of image formation • In-depth understanding of computer vision principles • Practical exposure and relevance of machine learning techniques in computer vision

Syllabus	
Module I **[10] Hours**	**Image processing fundamentals:** Introduction and Overview. Image Formation: Geometric primitives and transformations, photometric image formation, the principle of digital camera. Image Processing: Image transformations, image filtering, edge detection, corner detection.
Module II **[10] Hours**	**Geometrical techniques in computer vision:** Feature Detection and Matching: Feature detectors, feature descriptors, feature matching and feature tracking. Camera geometry: Camera projections, camera calibration, camera pose estimation. Depth Estimation: Epipolar geometry, sparse and desnse correspondence, multi-view stereo. Two-view Structure from Motion, Object tracking.
Module III **[10] Hours**	**Deep learning for computer vision:** Deep neural networks, convolutional neural networks (CNN), Three-dimensional CNNs, Recurrent neural networks, Transformers, and Generative models and applications.
Suggested Books:	1. Computer Vision: Algorithms and Applications, Springer London, (2011). Richard Szeliski. 2. Multiple View Geometry in Computer Vision, Cambridge University Press, (2004). Richard Hartley, Andrew Zisserman. 3. Computer Vision: A Modern Approach, Pearson, (2012). David Forsyth, Jean Ponce.
Evaluation:	7. Quizzes: 15% 8. Mid Term: 30% 9. End Term Exam: 50% 10. Teacher's Assessment: 5%

7th Semester IT (New Syllabus)

(IWT, IWT Lab, Computer Vision, Technical Writing)

CS124	**IWT**
Credits: 3 (3-0-0)	
Pre-requisite for this course:	Should have prior experience in one programming language.
Course Objective:	The course aims to: • Understanding the various concepts of HTML, CSS, XML, Java Script, jQuery, JSON, React.js, Node.js • Understanding the various steps in designing a static and dynamic website using client and server-side scripting. • Working on projects to gain practical knowledge about web development along with teamwork.
Course Outcome:	On completion of the course, a student should be able to: • Explain the history of the internet and related internet concepts that are vital in understanding web development. Analyze a web page and identify its elements and attributes • Demonstrate the important HTML tags for designing static pages and separate design from content using Cascading Style sheet. • Utilize the concepts of JavaScript to build dynamic website. • Use web application development software tools i.e. Ajax, Java Beans and XML, JQuery, JSON etc. and identify the environments currently available on the market to design web sites. • Learn Server-side programming • Working on projects using web technologies.

Syllabus

Module I **[8] Hours**	**The Web** Internet, Word Wide Web(WWW), History of the Internet, History of the Web, Protocols Governing the Web, Types of Websites, Web Applications, Web Projects, Web architecture, HTML, URL, HTTP, Issues in Web Development, Webserver, web browser, Internet standards, TCP/IP Protocol suite, MIME, Cyber Laws. Understand the need for HTTP, URL and its Anatomy, HTTP message format, Persistent and Non-persistent Connections, Web Caching, Proxy. **HTML** Introduction to HTML, Markup Language, HTML Tags, HTML Document Structure, HTML Comments, head, title, base, link, style, meta, script, Heading tags, paragraph, anchor, image, video, line breaks, text formatting, lists, frames, forms. **CSS** Inline Styles,Internal Style Sheet,External Style Sheet, Selectors, Colors, Backgrounds, Background Image, Background Attachment, Background Shorthand, Borders, Margins, Padding, Fonts, Links, Icons, Lists, Tables, Display, Position, Overflow, Float, inline-block, Horizontal & Vertical Align, Combinators, Pseudo-classes, Pseudo-elements, Opacity / Transparency, Navigation Bar, Dropdowns, Image Gallery, Attribute Selectors, Forms, Counters, Website Layout, Units, Specificity, Text Effects, Animations, Tooltip, Multiple Columns
Module II **[10] Hours**	**Java Script** Scripting: Java script: Introduction, statements, comments, variables, operators, documents, forms, , functions, objects, events, Strings, Numbers, Arrays, Date, Math, Random, Loops, Regxp, errors, this, Let, Const., classes, debugging . **The HTML DOM (Document Object Model)** Introduction ,DOM Methods, DOM Document, DOM Elements , DOM HTML, DOM CSS, DOM Events, DOM Navigation, DOM Nodes , DOM Nodelist **The Browser Object Model (BOM)** The Window Object, Window Size, Window History, Window Navigator, Browser Detection, JavaScript Timing Events, Cookies.

	Working on Cookies using Java script. **Advanced JavaScript Concepts** Closures, Inheritances, Event Loops,Callback Functions, Async/Await And Promises, Functional Programming, High-Order Functions, Generators, Hoisting, IIFEs, Memoization, Currying
Module III [12] Hours	**XML** Introduction to XML, XML vs HTML, Structures of a XML Document, Document Type Declaration (DTD), XML Validation, Well Formed XML Documents, Valid XML Document, XML DOM, XSL, XSL Transformation, XML Namespaces, XML Schema. AJAX AJAX Introduction, AJAX - The XMLHttpRequest Object, AJAX - Server Response, Ajax XML, Ajax Database jQuery jQuery DOM Selectors, HTML Content, jQuery CSS Styles, jQuery HTML DOM JSON Introduction, syntax, Data Types, Parsing, Stringify, Object, Arrays React.js Introduction, ES6, Render HTML, JSX, Components , props.state, 2Lifecycle, Events, forms, CSS Node.js Introduction to Node JS, Setup Development Environment, Node JS Modules, Node Package Manager, Creating Web Server, File System, Debugging Node JS Application, Events, Express JS, Serving Static Resources, Database Connectivity
Suggested Books: [Minimum 4/5	1. Xavier, C, "Web Technology and Design", New

Books] **(Include E-books also, if any)**	Age International. 2. Ivan Bayross," HTML, DHTML, Java Script, Perl & CGI", BPB Publication. Balagurusamy E, "Programming in JAVA", TMH. 3. David Geary , Cay S. Horstmann , "Core JavaServer™ Faces", Third Edition, Printice Hall 4. Ramesh Bangia, "Internet and Web Design", New Age International. 5. Patel and Barik,"Introduction to Web Technology & Internet", Acme Learning. 6. Don Goseselin, Ruth Guthrie, Luis A. Lopez et al. ,"The Web Warrior Guide to Web Design Technologies", Cenage Learning
Evaluation:	1. Assignments: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

	IWT Lab
Credits: 1(0-1-0)	
Pre-requisite for this course:	Should have prior experience in one programming language.
Course Objective:	The course aims to: • To develop web applications using HTML, CSS and JavaScript • To develop full stack applications using node.js
Course Outcome:	On completion of the course, a student should be able to: • Develop responsive web applications using HTML, CSS, JavaScript • Can apply advanced JavaScript concepts • Can develop Full Stack web applications.

Syllabus

	Lab 1: 1. **Create a Hello World Program using HTML** 2. **Write a program to demonstrate the use of all tags inside <Head> tag** 3. **Write a program in HTML to demonstrate the use of text formatting tags** **Lab 2:** 1. **Write a program in HTML to demonstrate the create different types of lists** 2. **Write a program in HTML to create tables** 3. **Write a program in HTML to use <DIV> tag** **Lab 3: Write a program in HTML and CSS to demonstrate the use of different types of CSS** **Lab 4: Develop a course page for a faculty** **Lab 5: Write a program in HTML and CSS to create a personal Home Page** **Lab 6: Write a program in HTML and CSS to create an interactive Tutorial** **Lab 7: Develop a Quiz Program using HTML, CSS and Java Script** **Lab 8: Develop a website for a university** **Lab 9: Develop an Event Registration Page** **Lab 10: Develop an Online Examination System** **Lab 11: Develop an online music site** **Lab 12: Develop an online bookstore application** **Lab 13: Develop an online multiplayer Game** **Lab 14: Develop a social media platform**

Suggested Books: [Minimum 4/5 Books] (Include E-books also, if any)	1. Xavier, C, "Web Technology and Design", New Age International. 2. Ivan Bayross," HTML, DHTML, Java Script, Perl & CGI", BPB Publication. Balagurusamy E, "Programming in JAVA", TMH. 3. David Geary , Cay S. Horstmann , "Core JavaServer™ Faces", Third Edition, Printice Hall 4. Ramesh Bangia, "Internet and Web Design", New Age International. 5. Patel and Barik,"Introduction to Web Technology & Internet", Acme Learning. 6. Don Goseselin, Ruth Guthrie, Luis A. Lopez et al. ,"The Web Warrior Guide to Web Design Technologies", Cenage Learning
Evaluation:	5. Lab Work: 70 6. Project and Viva: 30

	Machine Learning
Credits:3(3-0-0)	
Pre-requisitefor this course:	**Shouldhaveprior experienceinoneprogramming language.**
Course Objective:	Thecourseaims to: • Understanding the various concepts of Machine Learning. • To formulate Machine Learning problems corresponding to different applications. • To understand a range of machine learning algorithms along with their strengths and weaknesses. • To be able to apply machine learning algorithms to solve problems of moderate complexity. • To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models. • Working on projects to gain practical knowledge about data science along with teamwork.
Course Outcome:	Oncompletion ofthe course, astudent should beable to: • Appreciate the importance of visualization in the data analytics solution • Apply structured thinking to unstructured problems Understand a very broad collection of machine learning algorithms and problems Learn algorithmic topics of machine learning and mathematically deep enough to introduce the required theory • Develop an appreciation for what is involved in learning from data.

Syllabus	
ModuleI [10] Hours	**Introduction to Machine Learning** Introduction, Machine Learning – Types, Feature Engineering (Feature Selection, Feature Extraction), Mathematics for ML, Hypothesis Testing (Significance Tests), How's it works on the datasets? Bias-Variance Tradeoff, Regularization
ModuleII [10] Hours	**Supervised Learning** Introduction to classification and regression,Linear Regression, PACLearning, , Perceptron (Convergence),Multi-Layer Perceptron, Backpropagation,Gradient Descent (Linear Regression), Linear Regression with multiple features,Logistic Regression, k-Nearest Neighbor, Decision Tree (Gini Impurity), Decision Tree (ID3 Algorithm), Decision Tree(Regression), Support Vector Machine, Support Vector Machine – Kernels, Support Vector Machine – Regression,Naive Bayes Classifier
ModuleIII [10]Hours	**Ensemble Learning and Unsupervised Learning** Bagging,Ensemble Method - Boosting (Regression), Ensemble Method - Boosting (Classification), Clustering KMeans, Unsupervised - Clustering DBSCAN, Unsupervised - Dimensionality Reduction PCA,Unsupervised - Dimensionality Reduction UMAP,Unsupervised Algorithms - Dimensionality Reduction Autoencoder, Reinforcement Learning, Introduction to deep neural networks.
SuggestedBooks: [Minimum 4/5 Books] (IncludeE-books also, if any)	1. 1.Mitchell, T. M., Mitchell, T. M. (1997). Machine Learning. Germany: McGraw-Hill. 2. Michalski, R. S., Carbonell, J. G., Mitchell, T. M. (2014). Machine Learning: An Artificial Intelligence Approach (Volume I). United States: Elsevier Science. 3. Dr. M Gopal, *Applied Machine Learning*, 1st Edition, McGraw-Hill,2018 4. Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. 5. Burkov, Andriy. The hundred-page machine learning book. Vol. 1. Quebec City, QC, Canada: Andriy Burkov, 2019. 6. Zhang, Yagang, ed. New advances in machine learning. BoD–Books on Demand, 2010. 7. Unsupervised Learning: Foundations of Neural Computation, eds. Geoffrey Hinton and Terrence J. Sejnowski, The MIT Press, Cam- bridge, Massachusetts, 1999, 398 pp., ISBN 0-262-58168-X.

	8. Hiran, Kamal Kant, et al. Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition). BPB Publications, 2021. 9. Weber, Cornelius, Mark Elshaw, and N. Michael Mayer, eds. Reinforcement Learning. BoD–Books on Demand, 2008.
Evaluation:	1. Quizzes/Projects:15% 2. MidTerm:30% 3. EndTermExam:50% 4. Teacher'sAssessment:5%

	Machine Learning LAB
Credits:1(1-0-0)	
Pre-requisitefor this course:	**Shouldhaveprior experienceinoneprogramming language.**
Course Objective:	Thecourseaims to: Understanding the various concepts of Machine Learning. To formulate Machine Learning problems corresponding to different applications. To understand a range of machine learning algorithms along with their strengths and weaknesses. To be able to apply machine learning algorithms to solve problems of moderate complexity. To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models. Working on projects to gain practical knowledge about data science along with teamwork.
Course Outcome:	Oncompletion ofthe course, astudent should beable to: . Appreciate the importance of visualization in the data analytics solution Apply structured thinking to unstructured problems Understand a very broad collection of machine learning algorithms and problems Learn algorithmic topics of machine learning and mathematically deep enough to introduce the required theory Develop an appreciation for what is involved in learning from data.

Syllabus

1) Basic exercises on Python Machine Learning Packages such as Numpy, Pandas, and matplotlib.
2) WAP to take a input of a dataset. Split the independent and dependent variable to X and Y.
3) Create a data frame with columns at least 5 observations
    a. Retrieve a particular column from the DataFrame
    b. Summarize the data frame and observe the statistics of the DataFrame created
    c. Observe the mean and standard deviation of the data frame and print the values
    d. Perform the Data Preprocessing (i.e. Remove Duplicates, Check the missing Values, Standardization, Normalization, data balancing).
    e. Split the dataset into training and testing. X_train, X_test, Y_train and Y_test.
4) Write a program to implement perceptron for different learning tasks.
5) Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
6) Given a set of sample points in N-dimensional feature space. Write a program to fit the points with a hyperplane using Linear Regression. Calculate the sum of residual error.
7) Write a program that provides an option to compute different distance measures between two pointsin the N-dimensional feature space. Consider some sample datasets for computing distances amongsample points.
8) Write a program to implement the Logistic Regression for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier.
9) Write a program to demonstrate the working of the decision tree algorithm. Use an appropriate dataset for building the decision tree and apply this knowledge to classify a new sample.
10) Write a program to construct a Regression tree for cost estimation by assuming any numerical dataset.
11) Given a dataset. Write a program to implement k-Nearest Neighbour algorithm for classification. Print both correct and wrong predictions and the parameters of the confusion matrix.
12) Write a program to implement the Naive Bayesian classifier for a sample training data set. Computethe accuracy, precision and recall of the classifier, considering few test data sets.
13) Given a dataset for classification task. Write a program to implement Support Vector Machine and estimate its test performance through accuracy, AUC and ROC curve.
14) Write a program to implement feature reduction using Principle Component Analysis.
15) Write a program to implement the K means clustering algorithm. Select your own dataset to test theprogram. Demonstrate the nature of output with varying values of K.
16) Given a dataset. Implement Ensemble learning (Bagging and Booting) different architectures and compare their performance.
17) Write a program to preprocess a customer review text data followed by sentimental analysis.

	18) Implement dimensionality reduction –PCA upon a large dataset and perform the analysis. 19) Implement dimensionality reduction - UMAP upon a large dataset and perform the analysis. 20) Implement Reinforcement Learning on a dataset and compute the accuracy.
SuggestedBooks: [Minimum 4/5 Books] (IncludeE-books also, if any)	1.Vijayvargia, Abhishek, *Machine Learning with Python: An Approach to Applied Machine Learning*, BPB Publications, 1st edition,2018. 2. Aurelien Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, Oreilly, March 2017. 3. Dr. M Gopal, *Applied Machine Learning*, 1st Edition, McGraw-Hill,2018
Evaluation:	Quizzes/Projects:15% MidTerm:30% EndTermExam:50% Teacher'sAssessment:5%

8th Sem CSE (Research Project)

	Research Project
Credits:10 (0-0-0)	
Pre-requisitefor this course:	**Shouldhaveprior experienceinoneprogramming language, sound knowledge of Mathematics.**
Course Objective:	In the final semester (i.e. in 8th Semester CSEbranch), each and every student has to perform one Academic Research Project under the supervision of one department assigned faculty member. The project is to be completed in six months duration. Students are to appear the Mid Sem and End Sem evaluation of 100 marks comprising of 10 credits. They may continue to work on the same research topic as chosen in 7th Semester Technical Writing Seminar topic or the new topics may be assigned by the concerned faculty member. They have to work continuously under the supervision of the assigned faculty, have to write a major project report and may go for research paper publication in a conference or journal.
Course Outcome:	Students should be able to write a research report at the end of the study.
Syllabus **Students are advised to choose one recent research aper in any domain of their choice within last two years, try to understand, implement, present and submit their Thesis report.**	
Evaluation:	MidTerm:30% EndTermExam:70%

7th Sem CSE (SE, SE Lab, ML, ML Lab, Technical Writing)

CS125	SOFTWARE ENGINEERING
Credits: 3 (3-0-0)	
Pre-requisite for this course:	Fundamentals of programming languages.
Course Objective:	The course aims to: • A general understanding of software process models and an understanding of the role of project management including planning, scheduling, risk management, etc. • An understanding of software requirements and effectively document the SRS. • An understanding of implementation issues such as modularity and coding standards. • An understanding of approaches to verification and validation including static analysis, and reviews. • An understanding of software testing approaches such as unit testing and integration testing. • An understanding on quality control and how to ensure good quality software. • An understanding of some ethical and professional issues that are important for software engineers. • Skill development for significant teamwork and project-based experience.
Course Outcome:	On completion of the course, a student should be able to: • The ability to analyze, design, verify, validate, implement, apply, and maintain software systems. • Apply the process of analysis and design using the object-oriented approach. • learn to work as an effective and productive team member in a range of professional and social situations. • To communicate effectively with others, especially regarding the progress of the system development and the content of the design by means of reports and presentations. • The ability to work in one or more significant application domains. • The ability to manage the development of software systems.

Syllabus	
Module I **[10] Hours**	**Software Process Models:** Software Product, Software crisis, Handling complexity through Abstraction and Decomposition, Overview of software development activities, Process Models, Classical waterfall model, iterative waterfall model, prototyping mode, evolutionary model, spiral model, concurrent development model, RAD model, Agile models: Extreme Programming, and Scrum. **Software Requirements Engineering:** Requirement Gathering and Analysis, Functional and Non-functional requirements, Software Requirement Specification(SRS), IEEE 830 guidelines, Decision tables and trees.
Module II **[12] Hours**	**Structured Analysis & Design:** Overview of design process: High-level and detailed design, Cohesion and coupling, Modularity and layering, Function–Oriented software design: Structured Analysis using DFD Structured Design using Structure Chart, Basic concepts of Object-Oriented Analysis & Design. User interface design, Command language, menu and iconic interfaces., **Coding and Software Testing Techniques:** Coding, Code Review, documentation. Testing: Unit testing, Black-box Testing, Whitebox testing, Cyclomatic complexity measure, coverage analysis, mutation testing, Debugging techniques, Integration testing, System testing, Regression testing.
Module III **[8] Hours**	**Software Reliability and Software Maintenance:** Basic concepts in software reliability, reliability measures, reliability growth modeling, Quality SEI CMM, Characteristics of software maintenance, software reverse engineering, software reengineering, software reuse. **Emerging Topics:** Client-Server Software Engineering, Service-oriented Architecture (SOA), Software as a Service (SaaS).
Recommended Books:	1. Fundamentals of Software Engineering, Rajib Mall , PHI, 2014. 2. Software Engineering, A Practitioner's Approach, Roger S. Pressman, TMG Hill. 3. Software Engineering, I. Sommerville, 9th Ed., Pearson Education.
Evaluation:	• Quizzes: 15% • Mid Term: 30% • End Term Exam: 50% • Teacher's Assessment: 5%

CSxxx	SOFTWARE ENGINEERING LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	Fundamentals of programming languages.
Course Objective:	The course aims to: • A general understanding of software process models and an understanding of the role of project management including planning, scheduling, risk management, etc. • An understanding of software requirements and effectively document the SRS. • An understanding of implementation issues such as modularity and coding standards. • An understanding of approaches to verification and validation including static analysis, and reviews. • An understanding of software testing approaches such as unit testing and integration testing. • An understanding on quality control and how to ensure good quality software. • An understanding of some ethical and professional issues that are important for software engineers. • Skill development for significant teamwork and project-based experience.
Course Outcome:	On completion of the course, a student should be able to: • The ability to analyze, design, verify, validate, implement, apply, and maintain software systems. • Apply the process of analysis and design using the object-oriented approach . • learn to work as an effective and productive team member in a range of professional and social situations. • To communicate effectively with others, especially regarding the progress of the system development and the content of the design by means of reports and presentations. • The ability to work in one or more significant application domains. • The ability to manage the development of software systems.

Syllabus	
SE Lab Programs	Experiment1: Develop requirements specification for a given problem (Document SRS) Experiment 2: Develop DFD Model (Level 0, Level 1 DFD and data dictionary) of the sample problem (Use of a CASE tool required) Experiment 3: Develop structured design for the DFD model developed Experiment 4: Develop UML Use case model for a problem (Use of a CASE tool any of Rational rose, Argo UML, or Visual Paradigm etc. is required) Experiment 5: Develop Sequence Diagrams. Experiment 6: Develop Class diagrams. Experiment 7: Develop code for the developed class model using Java. Experiment 8: Use testing tool such as Junit. Experiment 9: Use a configuration management tool. Experiment 10: Use any one project management tool such as Microsoft Project or Gantt Project, etc.
Recommended Books:	1. Fundamentals of Software Engineering, Rajib Mall , PHI, 2014. 2. Software Engineering, A Practitioner's Approach, Roger S. Pressman, TMG Hill. 3. Software Engineering, I. Sommerville, 9th Ed., Pearson Education.
Evaluation:	• Quizzes: 15% • Mid Term: 30% • End Term Exam: 50% • Teacher's Assessment: 5%

	Technical Writing
Credits:3(0-0-0)	
Pre-requisitefor this course:	**Shouldhaveprior experienceinoneprogramming language, sound knowledge of Mathematics.**
Course Objective:	Undergraduate students are expected to develop a strong foundation in technical writing as it is a crucial skill for their academic and professional success. Technical writing involves the ability to convey complex engineering concepts, designs, and processes clearly and concisely to a diverse audience. Students are expected to be proficient in analyzing research papers. They must adhere to the principles of clarity, precision, and brevity, while also ensuring the correct use of technical jargon and industry-specific terminology. The incorporation of visual aids such as charts, diagrams, and tables are encouraged to enhance understanding. Furthermore, students should be familiar with citation styles commonly used in engineering reports to properly attribute sources. Mastery of these skills not only fulfills academic requirements but also prepares students for the rigorous communication demands of the engineering field.
Course Outcome:	Students should be able to write a research report at the end of the study.
Syllabus **Students are advised to choose one recent research aper in any domain of their choice within last two years, try to understand, implement, present and submit their report.**	
Evaluation	MidTerm:30% EndTermExam:70%

	Technical Writing
Credits:3(0-0-0)	
Students Shouldhaveprior experienceinoneprogramming language, sound knowledge of Mathematics.	
Undergraduate students are expected to develop a strong foundation in technical writing as it is a crucial skill for their academic and professional success. Technical writing involves the ability to convey complex engineering concepts, designs, and processes clearly and concisely to a diverse audience. Students are expected to be proficient in analyzing research papers. They must adhere to the principles of clarity, precision, and brevity, while also ensuring the correct use of technical jargon and industry-specific terminology. The incorporation of visual aids such as charts, diagrams, and tables are encouraged to enhance understanding. Furthermore, students should be familiar with citation styles commonly used in engineering reports to properly attribute sources. Mastery of these skills not only fulfills academic requirements but also prepares students for the rigorous communication demands of the engineering field.	
Students should be able to write a research report at the end of the study.	
Syllabus **Students are advised to choose one recent research aper in any domain of their choice within last two years, try to understand, implement, present and submit their report.**	
Evaluation:	MidTerm:30% EndTermExam:70%