AC –
Item No. –

# As Per NEP 2020

# University of Mumbai

| Syllabus for Major<br>Vertical – 1, 4 & 5 | | |
|---|---|---|
| Name of the Programme – B.E. (Artificial Intelligence & Data Science) | | |
| Faulty of Engineering | | |
| Board of Studies in Computer Engineering | | |
| U.G. Third Year Programme | Exit Degree | B.Sc. Tech Engineering- Artificial Intelligence & Data Science |
| Semester | | V & VI |
| From the Academic Year | | 2026-27 |

# University of Mumbai

**(As per NEP 2020)**

| Sr. No. | Heading | Particulars |
|---|---|---|
| 1 | **Title of program**<br>**O:_____** | **B.E. (Artificial Intelligence & Data Science)** |
| 2 | **Exit Degree** | **B.Sc. Tech Engineering- Artificial Intelligence & Data Science** |
| 3 | **Scheme of Examination**<br>**R: _____** | NEP<br>40% Internal<br>60% External, Semester End Examination<br>Individual Passing in Internal and External Examination |
| 4 | **Standards of Passing**<br>**R: _____** | **40%** |
| 5 | **Credit Structure**<br>**R: _____** | **Attached herewith** |
| 6 | **Semesters** | **Sem. V & VI** |
| 7 | **Program Academic Level** | **5.5** |
| 8 | **Pattern** | **Semester** |
| 9 | **Status** | **New** |
| 10 | **To be implemented from Academic Year** | **2026-27** |



| | | |
|---|---|---|
| **Sd/-** | **Sd/-** | **Sd/-** |
| **Dr. Subhash K. Shinde** | **Dr. Deven Shah** | **Prof. Shivram S. Garje** |
| **BoS-Chairman-Computer Engineering** | **Associate Dean** | **Dean** |
| **Faculty of Technology** | **Faculty of Science & Technology** | **Faculty of Science & Technology** |

# Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The third-year engineering course consists of a core and elective training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, program elective course, multidisciplinary minor, IKS, PCE and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program, elective program, communication skills and skills required for the Computer Engineering Branch of engineering in the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends. Elective courses are offered to the learners to understand the concepts of emerging technology. This will help with the holistic development of learners through interdisciplinary skills development and mobility among other departments as real-time solutions. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Computer Engineering in Engineering for open electives and multidisciplinary minor courses in sem V and VI. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

Program Core Course and Program Elective Courses Cover Computer Engineering core and elective courses. Also, MDM is where a pool of subjects is given for selection. In addition to this learner will have IKS and PCE courses. Considering the present scenario, diverse choices need to be made available to fulfil the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. The IKS course will be program-specific, which can be achieved in Undergraduate training by giving an objective viewpoint to the learning technology, from ascent time and how it was learned and the process and transitioning a learner from a rote learner to a creative professional. For this purpose, PCE is introduced in the First Semester to orient a communication skills to become skilled professionals for placement and higher education entrance exams. Considering the NEP-2020 structure of award of Certificate after U.G. First Year, U. G. Diploma after the Second Year and B.Sc. Tech Engineering after the Third Year at multiple exit-point pools of Vocational skills, is arranged to give exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every core and elective course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught to the learners, which will enhance the learners' learning process. NEP 2020 grading system enables a much-needed shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on the holistic development of the learners through mobility and continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020; however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, self learning, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the third-year syllabus must not be heavily loaded for the learner and it is of utmost importance that the learner entering into the third year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Third Year of Engineering from the academic year 2026-27. Subsequently, this system will be carried forward for Final Year Engineering in the academic years 2027-28, respectively.

| | | |
|---|---|---|
| **Sd/-** | **Sd/-** | **Sd/-** |
| **Dr. Subhash K. Shinde** | **Dr. Deven Shah** | **Prof. Shivram S. Garje** |
| **BoS-Chairman-Computer Engineering** | **Associate Dean** | **Dean** |
| **Faculty of Technology** | **Faculty of Science & Technology** | **Faculty of Science & Technology** |

## B.Sc. Tech Engineering- Artificial Intelligence & Data Science

### Credit Structure (Sem. V & VI)

| | R: _____C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Level** | **Semester** | **Major** | | **Minor** | **OE** | **VSC, SEC (VSEC)** | **AEC, VEC, IKS** | **OJT, FP, CEP, CC,RP** | **Cum. Cr. / Sem.** | **Degree/ Cum. Cr.** |
| | | **Mandatory** | **Electives** | | | | | | | |
| **5.5** | V | PCC501:3<br>PCC502:3<br>PCC503:3<br>PCL501: 1<br>PCL502:1<br>PCL503:1 | PEC501:3<br>PEL501:1 | MDC:3<br>MDL:1 | OE:2 | -- | IKS: 2 | -- | 24 | **B.Sc. Tech Engineering 48** |
| | R:_____D | | | | | | | | | |
| | VI | PCC601:3<br>PCC602:3<br>PCL601:1<br>PCL602:1 | PEC601:3<br>PEC602:3<br>PEL601:1<br>PEL602:1 | MDC: 3<br>MDL:1 | -- | VSEC:2 | AEC: 2 | -- | 24 | |
| | **Cum Cr.** | 20 | 12 | 8 | 2 | 2 | *2+2* | -- | 48 | |

**Exit option:** Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

**[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project ]**

# T.E.
# Scheme

**Program Structure for Third Year of Artificial Intelligence & Data Science**
**UNIVERSITY OF MUMBAI (With Effect from 2026-2027)**

**SEMESTER V**

| Course Code | Course Description | Teaching Scheme (Contact Hours) | | | Credit Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Practical | Tutorial | Theory | Tutorial | Practical | Total Credits |
| 2015111 | Statistics for Machine Learning and Data Science | 3 | -- | -- | 3 | -- | -- | 3 |
| 2015112 | Artificial Intelligence and Soft Computing | 3 | -- | -- | 3 | -- | -- | 3 |
| 2015113 | Agile Software Development and DevOps | 3 | -- | -- | 3 | -- | -- | 3 |
| 201511# | Program Electives -I | 3 | -- | -- | 3 | -- | -- | 3 |
| 2015211 | Multi-disciplinary Minor | 3 | -- | -- | 3 | -- | -- | 3 |
| 2015117 | Statistics for Machine Learning and Data Science Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 2015118 | Artificial Intelligence and Soft Computing Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 2015119 | Agile Software Development and DevOps Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 201512# | Program Electives -I Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 2015212 | Multi-disciplinary Minor Lab | -- | 2@ | -- | -- | -- | 1 | 1 |
| 2015311 | To be taken from the bucket provided by the University from other Faculty | 2# | -- | --- | 2 | -- | -- | 2 |
| 2015511 | Indian Knowledge System (Syllabus common to all Branch). | -- | 2*+2 | -- | -- | -- | 2 | 2 |
| **Total** | | **17** | **14** | **00** | **17** | **00** | **07** | **24** |

***** Two hours of practical class to be conducted for full class as demo/discussion.
Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

# Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

@ Institute shall offer a course for MDM from other Engineering Boards Group.

| **Program Electives:** | |
|---|---|
| Course Code | Course Description |
| 2015114 | Web and Mobile Application Development |
| 2015115 | User Experience Design with VR |
| 2015116 | Computer Network |
| Lab Code | Lab Description |
| 2015120 | Web and Mobile Application Development Lab |
| 2015121 | User Experience Design with VR Lab |
| 2015122 | Computer Network Lab |

**Program Structure for Third Year of Artificial Intelligence & Data Science**

**UNIVERSITY OF MUMBAI (With Effect from 2026-2027)**

**SEMESTER V**

| Course Code | Course Description | Examination Scheme | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Internal Assessment Test (IAT) | | | End Sem. Exam Marks | End Sem. Exam Duration (Hrs) | Term Work (Tw) | Oral & Pract. | Total |
| | | IAT-I | IAT-II | Total (IAT-I) + IAT-II) | | | | | |
| 2015111 | Statistics for Machine Learning and Data Science | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2015112 | Artificial Intelligence and Soft Computing | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2015113 | Agile Software Development and DevOps | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 201511# | Program Electives -I | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2015211 | Multi-disciplinary Minor | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2015117 | Statistics for Machine Learning and Data Science Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 2015118 | Artificial Intelligence and Soft Computing Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 2015119 | Agile Software Development and DevOps Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 201512# | Program Electives -I Lab | -- | -- | -- | -- | -- | 25 | - | 25 |
| 2015212 | Multi-disciplinary Minor Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 2015311 | To be taken from the bucket provided by the University from other Faculty | -- | -- | -- | -- | -- | 50 | -- | 50 |
| 2015511 | Indian Knowledge System (Syllabus common to all Branch). | -- | -- | -- | -- | -- | 50 | -- | 50 |
| Total | | **100** | **100** | **200** | **300** | **10** | **225** | **100** | **825** |

| **Program Electives:** | |
|---|---|
| Course Code | Course Description |
| 2015114 | Web and Mobile Application Development |
| 2015115 | User Experience Design with VR |
| 2015116 | Computer Network |
| Lab Code | Lab Description |
| 2015120 | Web and Mobile Application Development Lab |

| | |
|---|---|
| 2015121 | User Experience Design with VR Lab |
| 2015122 | Computer Network Lab |

**Program Structure for Third Year of Artificial Intelligence & Data Science**

**UNIVERSITY OF MUMBAI (With Effect from 2026-2027)**

**SEMESTER VI**

| Course Code | Course Description | Teaching Scheme (Contact Hours) | | | Credit Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Practical | Tutorial | Theory | Tutorial | Practical | Total Credits |
| 2016111 | Machine Learning and Swarm Intelligence | 3 | -- | -- | 3 | -- | -- | 3 |
| 2016112 | Data Analytics and Visualization | 3 | -- | -- | 3 | -- | -- | 3 |
| 201611# | Program Electives -II | 3 | -- | -- | 3 | -- | -- | 3 |
| 201611# | Program Electives -III | 3 | -- | -- | 3 | -- | -- | 3 |
| 2016211 | Multi-disciplinary Minor | 3 | -- | -- | 3 | -- | -- | 3 |
| 2016411 | Mini Project-II | -- | 4 | -- | -- | -- | 2 | 2 |
| 2016511 | Professional Skill | -- | 2*+2 | -- | -- | -- | 2 | 2 |
| 2016119 | Machine Learning and Swarm Intelligence Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 2016120 | Data Analytics and Visualization Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 201612# | Program Electives -II Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 201612# | Program Electives -III Lab | -- | 2 | -- | -- | -- | 1 | 1 |
| 2016212 | Multi-disciplinary Minor Lab | -- | 2@ | -- | -- | -- | 1 | 1 |
| Total | | 15 | 18 | 00 | 15 | 00 | 09 | 24 |

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

@ Institute shall offer a course for MDM from other Engineering Boards.

| | Course Code | Course Description | Lab Code | Lab Description |
|---|---|---|---|---|
| Program Electives -II | 2016113 | Internet of Things | 2016121 | Internet of Things Lab |
| | 2016114 | Cyber Security | 2016122 | Cyber Security Lab |
| | 2016115 | Nature-Inspired Optimization Algorithms | 2016123 | Nature-Inspired Optimization Algorithms Lab |
| Program Electives -III | 2016116 | Edge Computing | 2116124 | Edge Computing Lab |
| | 2116117 | Digital Imaging Techniques and Analysis | 2116125 | Digital Imaging Techniques and Analysis Lab |
| | 2016118 | Blockchain Technology | 2016126 | Blockchain Technology Lab |



**Program Structure for Third Year of Artificial Intelligence & Data Science**

**UNIVERSITY OF MUMBAI (With Effect from 2026-2027)**

**SEMESTER VI**

| Course Code | Course Description | Examination Scheme | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Internal Assessment Test (IAT) | | | End Sem. Exam Marks | End Sem. Exam Duration (Hrs) | Term Work (Tw) | Oral & Pract. | Total |
| | | IAT-I | IAT-II | Total (IAT-I) + IAT-II) | | | | | |
| 2016111 | Machine Learning and Swarm Intelligence | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2016112 | Data Analytics and Visualization | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 201611# | Program Electives -II | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 201611# | Program Electives -III | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2016211 | Multi-disciplinary Minor | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |
| 2016411 | Mini Project-II | -- | -- | -- | -- | -- | 50 | -- | 50 |
| 2016511 | Professional Skill | -- | -- | -- | -- | -- | 25 | -- | 25 |
| 2016119 | Machine Learning and Swarm Intelligence Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 2016120 | Data Analytics and Visualization Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 201612# | Program Electives -II Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 201612# | Program Electives -III Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| 2016212 | Multi-disciplinary Minor Lab | -- | -- | -- | -- | -- | 25 | 25 | 50 |
| Total | | **100** | **100** | **200** | **300** | **10** | **200** | **125** | **825** |

| | Course Code | Course Description | Lab Code | Lab Description |
|---|---|---|---|---|
| | 2016113 | Internet of Things | 2016121 | Internet of Things Lab |
| | 2016114 | Cyber Security | 2016122 | Cyber Security Lab |

| | | | | |
|---|---|---|---|---|
| Program Electives -II | 2016115 | Nature-Inspired Optimization Algorithms | 2016123 | Nature-Inspired Optimization Algorithms Lab |
| Program Electives -III | 2016116 | Edge Computing | 2116124 | Edge Computing Lab |
| | 2116117 | Digital Imaging Techniques and Analysis | 2116125 | Digital Imaging Techniques and Analysis Lab |
| | 2016118 | Blockchain Technology | 2016126 | Blockchain Technology Lab |



# Vertical – 1 Major

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015111 | Statistics for Machine Learning and Data Science | 3 | - | - | 3 | - | - | 3 |

| Course Code | Course Name | Theory | | | | | Term Work | Pract / Oral | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | Internal Assessment | | | End Sem Exam | Exam Duration (in Hrs) | | | |
| | | IAT -I | IAT-II | IAT-I + IAT-II | | | | | |
| 2015111 | Statistics for Machine Learning and Data Science | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |

**Rationale:** In modern data-driven systems, statistical understanding forms the backbone of Machine Learning and Data Science. This course integrates foundational statistical concepts with practical statistical learning methods such as regression, classification, resampling, unsupervised learning, Bayesian reasoning, and ethical AI practices. Students will learn to perform sound data analysis, build interpretable ML models, evaluate them rigorously, and understand the implications of data-driven decisions.

**Course Objectives:** By the end of the course, the learner will be able to:

1. Understand fundamental concepts of probability, descriptive statistics, and sampling techniques for data analysis.
2. Apply estimation and hypothesis testing methods, including classical and resampling approaches.
3. Build and interpret regression models (linear, nonlinear, and regularized) for predictive analytics.
4. Implement supervised and unsupervised statistical learning algorithms to model real-world data.
5. Evaluate models using cross-validation, MLE, and statistical performance metrics.
6. Analyze ethical implications, interpret model decisions, and apply Bayesian reasoning for explainable ML.

**Course Outcomes:** After successful completion of the course, students will be able to:

1. Summarize and interpret datasets using descriptive measures, sampling and distributional analysis.
2. Perform statistical inference using estimation, hypothesis testing, and resampling approaches.
3. Build, evaluate, and interpret regression and regularized models.
4. Apply supervised and unsupervised learning algorithms effectively.
5. Evaluate models using appropriate metrics, Bayesian reasoning, and cross-validation.
6. Analyze fairness, model interpretability, and ethical implications in ML workflows.

**Prerequisite:** Basic Probability, Linear Algebra, Python Programming

**DETAILED SYLLABUS:**

| Sr. No. | Name of Module | Detailed Content | Hours | CO Mapping |
|---|---|---|---|---|
| 0 | **Prerequisite** | Basic Probability, Linear Algebra, Python Programming | 00 | |
| I | **Statistical Foundations, Covariance, and Sampling** | Types of data: categorical, numerical, ordinal. Descriptive statistics: central tendency, dispersion, skewness, kurtosis. Data visualization: histograms, boxplots, scatter plots, pair plots. Probability distributions: Normal, Binomial, Poisson, Exponential, Uniform, Beta, Gamma. Covariance, correlation, confounding, correlation matrices. Distance measures: Euclidean, Manhattan, Cosine. Sampling techniques: simple random, stratified, cluster. Central Limit Theorem (CLT) and Law of Large Numbers. Data preprocessing: handling missing values, outlier treatment, encoding, normalization.<br>**Self-learning Topics:** Explore "An Introduction to Statistical Learning (ISLR2)" YouTube videos and Kaggle datasets for descriptive analytics and sampling distribution exercises.<br>https://onlinecourses.nptel.ac.in/noc24_ma03/preview<br>https://onlinecourses.nptel.ac.in/noc24_ma64/preview? | 07 | CO1, CO6 |
| II | **Statistical Inference, Estimation & Resampling** | Point estimates, properties of estimators: bias, variance, MSE. Maximum Likelihood Estimation (MLE) and Method of Moments. Confidence intervals (mean, proportion, variance). Hypothesis testing: z-test, t-test, chi-square, ANOVA. Nonparametric tests: Mann–Whitney U, Kruskal–Wallis. Resampling methods: bootstrap CI, permutation tests. Likelihood Ratio Tests (LRT). Multiple testing correction: Bonferroni & FDR. Introduction to Monte Carlo simulation for sampling distribution approximation.<br>**Self-learning Topics:** Explore bootstrapping via Python tutorials. Read "Practical Statistics for Data Scientists", Ch. 5 on inference.<br>https://onlinecourses.nptel.ac.in/noc25_ma55/preview? | 06 | CO2, CO5 |
| III | **Regression Models, Regularization & Nonlinear Models** | Simple & multiple linear regression.<br>Model assumptions, residual analysis, multicollinearity (VIF), leverage, Cook's distance. Polynomial regression and interaction effects. Nonlinear regression models. | 08 | CO3, CO6 |

| | | | | |
|---|---|---|---|---|
| | | Regularization: Ridge, Lasso, **Elastic Net.** Spline regression, LOESS (conceptual). Generalized Linear Models (GLM) – logistic (binary), Poisson (count data). Model evaluation metrics: MAE, MSE, RMSE, $R^2$, Adjusted $R^2$. Subset selection: forward, backward (overview). Model calibration and interpretability basics.<br>**Self-learning Topics:** Practice polynomial/spline regression using scikit-learn. Use open datasets to implement regularization.<br>https://onlinecourses.nptel.ac.in/noc24_cs20/preview | | |
| IV | **Classification & Supervised Learning** | Classification metrics: accuracy, precision, recall, F1, ROC–AUC, PR curves. Logistic regression, odds ratio, decision boundary. Naïve Bayes (Frequentist and Bayesian views). k-NN classifier. Decision Trees & pruning. Ensemble methods: Bagging, Random Forest, Gradient Boosting (overview). Support Vector Machines (SVM): linear & kernel trick– (conceptual). Handling imbalanced data: SMOTE, class weights, stratified splits. Confusion matrix analysis & error trade-offs. Model calibration: Platt scaling & isotonic regression (overview).<br>**Self-learning Topics:** Implement logistic and k-NN in Python. Explore metric trade-offs using real datasets. | 07 | CO4, CO6 |
| V | **Unsupervised Learning & Dimensionality Reduction** | PCA: covariance matrix, Eigen decomposition, explained variance. LDA (supervised dimensionality reduction). Kernel PCA & ICA (conceptual overview). Clustering: K-means, hierarchical, DBSCAN. Cluster evaluation: silhouette score, Davies–Bouldin index. Dimensionality reduction for visualization: t-SNE, UMAP (conceptual). Outlier detection methods: Z-score, IQR, Isolation Forest (intro), Local Outlier Factor (LOF) – conceptual overview.<br>**Self-learning Topics:** Visualize PCA/t-SNE clusters. Compare clustering methods with real-world datasets. | 05 | CO4, CO6 |
| VI | **Model Evaluation, Bayesian Inference & Ethics** | Model selection: AIC, BIC.Bias–variance trade-off. Cross-validation: k-fold, LOOCV, repeated CV. Calibration curves and Brier score (conceptual). Bayesian fundamentals: prior, likelihood, posterior. MAP vs MLE, conjugate priors (Beta–Bernoulli, Normal–Normal). Naïve Bayes from Bayesian perspective. Interpretability tools (conceptual): SHAP, LIME, PDP, ICE. Fairness in ML: equalized odds, demographic parity. Ethical considerations: model bias, transparency, data | 06 | CO5, CO6 |

| | | | | |
|---|---|---|---|---|
| | | leakage, Data Privacy Overview: anonymization, differential privacy. Reproducible ML: dataset versioning, MLflow / Weights & Biases (W&B conceptual).<br>**Self-learning Topics:** Read beginner-friendly blogs on Bayesian stats. Interpretability tools: SHAP, LIME (high-level); permutation importance, PDPs, ICE plots. | | |

**Text Books:**


1. Aggarwal, Charu C. *Probability and Statistics for Machine Learning*. Springer, 2024.
2. James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert. *An Introduction to Statistical Learning: with Applications in Python* (2nd ed.). Springer, 2023.
3. Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
4. *Practical Statistics for Data Scientists — Peter Bruce et al., 2020*


**References:**


1. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer, 2009.
2. DasGupta, Anirban. *Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics*. Springer, 2011.
3. Wasserman, Larry. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
4. McElreath, Richard. Statistical Rethinking: *A Bayesian Course with Examples in R and Stan*. CRC Press, 2020 (2nd ed.).
5. *Before Machine Learning, Volume 3: Probability & Statistics for AI*. Packt Publishing,


**Online References:** Students are encouraged to use the following verified online learning platforms for additional learning and self-study.

| Sr. No. | Website Name |
|---|---|
| **1.** | https://www.statlearning.com/ |
| **2.** | https://scikit-learn.org/stable/ |
| **3.** | https://developers.google.com/machine-learning/crash-course |
| **4.** | https://distill.pub/2016/misread-tsne/ |
| **5.** | https://christophm.github.io/interpretable-ml-book/ |
| **6** | https://onlinecourses.nptel.ac.in/noc25_ma20/preview? |
| **7.** | https://onlinecourses.nptel.ac.in/noc25_ma21/preview? |

**Assessment:**

**Internal Assessment (IA) for 20 marks:**

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

**Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

| | | |
|---|---|---|
| **NB:**<br>(1) Question 1 is compulsory.<br>(2) Solve any three questions out of five. | | |
| **Q.1** | **Solve any three questions out of five: (05 marks each)** | **15** |
| a) | | 05 |
| b) | | 05 |
| c) | | 05 |
| d) | | 05 |
| e) | | 05 |
| **Q.2,3, 4,5,6** | | **15** |
| a) | Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module. | 07or 08 |
| b) | | 08 or 07 |

**Note for QP Setters:**

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015112 | Artificial Intelligence & Soft Computing | 3 | - | - | 3 | - | - | 3 |

| Course Code | Course Name | Theory | | | | | Term Work | Pract / Oral | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | Internal Assessment | | | End Sem Exam | Exam Duration (in Hrs) | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | | |
| 2015112 | Artificial Intelligence & Soft Computing | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |

**Course Objectives: The course aims to:**

1. To introduce the fundamental concepts, evolution, and significance of Artificial Intelligence and Soft Computing, highlighting their roles in solving real-world problems.
2. To enable students to understand and apply classical AI search strategies such as uninformed, informed, and local search for goal-driven problem solving.
3. To develop the ability to represent and reason with knowledge using logical and probabilistic frameworks for handling uncertainty in intelligent systems.
4. To familiarize learners with fuzzy set theory, fuzzy rules, and inference mechanisms for approximate reasoning and intelligent decision-making.
5. To impart knowledge of neural network architectures and learning algorithms for modelling cognitive and pattern-recognition processes.
6. To integrate fuzzy logic and neural networks into hybrid intelligent systems and analyze their comparative performance across applications.

**Course Outcomes: After successful completion of the course, the learner will be able to:**

1. Describe the core concepts, constituents, and applications of Artificial Intelligence and Soft Computing.
2. Apply various uninformed, informed, and local search techniques to solve real-world and optimization problems.

3. Represent and reason with knowledge using propositional and predicate logic, and manage uncertainty using probabilistic reasoning.
4. Develop and analyze fuzzy inference systems for decision-making and control applications.
5. Design, train, and evaluate neural network models using supervised and unsupervised learning algorithms.
6. Integrate fuzzy and neural models into hybrid intelligent systems such as ANFIS and evaluate their performance.

**DETAILED SYLLABUS:**

| Sr. No | Name of Module | Detailed Content | Hours | CO Mapping |
|---|---|---|---|---|
| 0 | **Prerequisite** | Mathematical Foundations: Linear Algebra, Programming Skills: Python (preferred), Basic understanding of data structures, Discrete Mathematics | -- | -- |
| I | **Introduction to Artificial Intelligence & Soft Computing** | Introduction to Artificial Intelligence:<br>Artificial Intelligence (AI), AI Perspectives, Applications of AI, The present state of AI<br>Introduction to Soft Computing:<br>Soft computing Constituents, Characteristics of Neuro Computing and Soft Computing, Difference between Hard Computing and Soft Computing, Concepts of Learning and Adaptation.<br>**Self-learning Topics:** Recent advances in Generative AI, ChatGPT-like models, ethical challenges in AI. | 05 | CO1 |
| II | **Solving Problems by Searching** | Definition, State space representation, Problem as a state space search, Problem formulation, Uninformed Search: Depth First Search, Breadth First Search, Informed Search: Greedy Best First Search, A* Search. Local Search: Hill Climbing Search, Simulated Annealing Search,<br>**Self-learning Topics:** Game Playing, Adversarial Search Techniques | 07 | CO2 |
| III | **Knowledge and Reasoning** | Knowledge Representation Systems, Propositional Logic (PL): Syntax, Semantics, Formal logic-connectives, Predicate Logic: FOPL, Syntax, Semantics, Quantification, Inference rules in FOPL, Forward Chaining, Backward Chaining and Resolution in FOPL.<br>Handling Uncertain Knowledge, Random Variables, Prior and Posterior Probability<br>**Self-learning Topics:** Ontologies and Semantic Web reasoning | 07 | CO3 |
| IV | **Fuzzy Set Theory & Fuzzy Rules, Reasoning and** | Fuzzy Set Theory: Fuzzy Sets, Fuzzy relations, Fuzzification and Defuzzification. Features of the membership Functions<br>Fuzzy Rules, Reasoning and Inference System: Fuzzy Rules: Fuzzy If-Then Rules, Fuzzy Reasoning Fuzzy | 07 | CO4 |

| | | | | |
|---|---|---|---|---|
| | **Inference System** | Inference System (FIS): Mamdani FIS, Sugeno FIS, Comparison between, Mamdani and Sugeno FIS<br>**Self-learning Topics:** Adaptive fuzzy systems and membership optimization | | |
| V | **Neural Networks** | Basics of Neural Networks: Introduction to Neural Networks, Biological Neural Networks, McCulloch Pitt model<br>Supervised Learning algorithms: Perceptron (Single Layer, Multi-layer), Linear separability, Delta learning rule, Back Propagation algorithm<br>Un-Supervised Learning algorithms: Hebbian Learning, Winner take all, Self Organizing Maps, Learning Vector Quantization.<br>**Self-learning Topics:** CNNs and RNNs — introductory overview and comparison | 08 | CO5 |
| VI | **Hybrid System** | Introduction to Hybrid Systems, Key applications of hybrid systems (industrial control, robotics, forecasting), Adaptive Neuro-Fuzzy Inference System (ANFIS) , Comparative evaluation of Fuzzy, Neural, and Neuro-Fuzzy models.<br>**Self-learning Topics:** Sustainability and governance in hybrid AI systems. | 05 | CO6 |

**Text Books:**


1. Stuart Russell and Peter Norvig**,** *Artificial Intelligence – A Modern Approach*, 4th Edition, Pearson Education, 2021.
2. S. Rajaraman and G. Vijayalakshmi Pai**,** *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications,* PHI Learning, 2017.
3. S. N. Sivanandam and S. N. Deepa**,** *Principles of Soft Computing, 3rd Edition*, Wiley India, 2018.


**References:**


1. Elaine Rich, Kevin Knight, and Shivashankar B. Nair**,** *Artificial Intelligence,* 3rd Edition, McGraw-Hill Education, 2017
2. Simon Haykin**,** *Neural Networks and Learning Machines,* 3rd Edition, Pearson, 2017.
3. Bart Kosko**,** *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence,* Prentice Hall, 1992.


**Online References:**

| Sr. No. | Website Name |
|---|---|
| **3.** | https://onlinecourses.nptel.ac.in/noc25_cs48/preview? |
| **4.** | https://nptel.ac.in/courses/106105077 |
| **3.** | https://stanford-cs221.github.io/spring2024/ |

**Assessment:**

**Internal Assessment (IA) for 20 marks:**

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

- **Question paper format:**
- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.



University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

| | | |
|---|---|---|
| **NB:**<br>(1) Question 1 is compulsory.<br>(2) Solve any three questions out of five. | | |
| **Q.1** | **Solve any three questions out of five: (05 marks each)** | **15** |
| a) | | 05 |
| b) | | 05 |
| c) | | 05 |
| d) | | 05 |
| e) | | 05 |
| **Q.2,3, 4,5,6** | | **15** |
| a) | Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module. | 07or 08 |
| b) | | 08 or 07 |

**Note for QP Setters:**

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

.....

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015113 | Agile Software Development & DevOps | 3 | - | - | 3 | - | - | 3 |

| Course Code | *Course Name* | *Theory* | | | | | Term Work | Pract / Oral | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | *Internal Assessment* | | | End Sem Exam | Exam Duration (in Hrs) | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | | |
| 2015113 | Agile Software Development & DevOps | 20 | 20 | 40 | 60 | 2 | -- | -- | 100 |

**Course Objectives:**

1. Understand software engineering frameworks, SDLC models, and the need for Agile software development.
2. Recognize different Agile development models, their methods, values, roles, artifacts, and business benefits.
3. Understand Agile requirements elicitation, modeling, prioritization, and Agile design principles.
4. Learn Agile project planning, estimation techniques, forecasting, and risk management.
5. Understand Agile approaches to software quality, testing principles, automation, and continuous testing.
6. Understand DevOps principles, tools, lifecycle, and integration with Agile for business agility.

**Course Outcomes: After learning this course, students will be able to**

1. Describe SDLC process models, differentiate traditional vs. Agile models, and summarize Agile principles and stakeholder roles.
2. Compare Agile development models (Scrum, XP, Kanban, FDD, Lean, Crystal, ASD), and apply Agile values and tools to practical scenarios.
3. Develop and prioritize Agile requirements using product and sprint backlogs, user stories, story cards, and apply Agile design principles.
4. Estimate, plan, and track Agile projects using techniques such as Planning Poker, story points, velocity, burndown charts, and forecasting methods**.**
5. Implement Agile testing strategies including TDD, BDD, acceptance and exploratory testing, and analyze quality metrics using automation tools**.**
6. Describe DevOps principles and lifecycle, implement CI/CD pipelines using DevOps tools, and evaluate the impact on software delivery and team collaboration.

**DETAILED SYLLABUS:**

| Sr. No. | Name of Module | Detailed Content | Hours | CO Mapping |
|---|---|---|---|---|
| 0 | **Prerequisite** | Basics of Programming and concepts in Software development. | | |
| I | **Introduction to Agile Software Development** | Software Engineering-process framework, Software Development Life Cycle (SDLC) Process Models: The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models, Agile Process Model, Component-Based Development.<br>Agile Process model, Need of Agile software development, Agile Manifesto and Principles, Stakeholders and Challenges, History of Agile Software Development, Traditional Model vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Teams, Team Meetings and Ceremonies, Agile Drivers, Capabilities and Values.<br><br>**Self-learning Topics:** explore advanced Agile concepts such as Agile scaling frameworks (SAFe, LeSS), Agile estimation techniques, and metrics like velocity and burndown charts. | 6 | CO 1 |
| II | **Agile Processes** | Agile Development Models: Scrum, Extreme Programming, Feature Driven Development, Crystal, Kanban, Adaptive Software Development, Test Driven Development, and Lean Software Development, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility.<br><br>**Self-learning Topics:** To Study Agile metrics like cycle time, lead time, and cumulative flow diagrams along with tools such as Jira, Azure DevOps, and Trello for practical implementation. | 8 | CO 2 |
| III | **Agile Requirements** | Impact of Agile Processes in RE, Product Backlog, Sprint Backlogs, Current Agile | 7 | CO 3 |

| | | | | |
|---|---|---|---|---|
| | **Engineering & Design** | Practices, Variance, Managing Unstable Requirements, Requirements Elicitation, Agile Requirements Prioritization, Agile Requirements Modeling and Generation, Concurrency in Agile Requirements Generation, User Stories – Specification and Formats, Story-Cards. Agile Design: Symptoms of poor design, Agile design principles<br><br>**Self-learning Topics:** Explore advanced backlog management techniques such as user story mapping, impact mapping, and MoSCoW prioritization to deepen their understanding of Agile requirements. | | |
| IV | **Agile Planning & Estimation** | Planning levels; User stories and estimation; Planning Poker; Velocity and Burndown Charts;<br>Agile risk management.<br>Agile forecasting and project Management, velocity, progress tracking, Track Done pattern, Project forecasting, Estimation: Level of Effort, Ideal Time, Hours, Story Points, Team Participation, Sprint Planning and Estimation.<br>**Self-learning Topics:** Advanced Agile forecasting techniques such as Monte Carlo simulations, probabilistic planning, and throughput-based forecasting for more accurate predictions. | 6 | CO 4 |
| V | **Agile Quality Assurance & Testing** | Agile Metrics, Agile Approach to Quality Assurance.<br>Testing: Functionality Testing, UI Testing, Performance Testing, Security Testing, Agile Testing: Principles of Agile Testers; The Agile Testing Quadrants, Test Driven Development, Acceptance Tests, Refactoring, Agile Automation, Test Automation Pyramid. Case Study – Selenium/ Jira<br>**Self-learning Topics:** Continuous Testing, BDD, exploratory testing, and modern automation tools such as Selenium, Cypress, Postman, and JMeter. | 6 | CO 5 |
| VI | **DevOps** | DevOps: Introduction, DevOps - Importance and Benefits, DevOps Principles and Practices, 7-C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, Choosing the right DevOps tools and challenges in DevOps Implementation. Mapping App to DevOps-Assessment. Study of various Open Source Tools in DevOps.<br>**Self-learning Topics:** CI/CD pipelines, cloud-based DevOps tools, and real-world case studies on DevOps adoption and automation practices. | 6 | CO 6 |

**Text Books:**

1. Roger Pressman, Software Engineering: A Practitioner's Approach, 6th edition, McGraw Hill International Publication.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Introduction to Agile Methods, Sondra Ashmore
4. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint.
5. Mike Cohn – Agile Estimating and Planning (Pearson)

**Reference Books:**

1. David J. Anderson and Eli Schragenheim, ―Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results‖, Prentice Hall, 2003.
2. Craig Larman, Agile and Iterative Development: A Manager's Guide, Addison-Wesley, 2004.
3. Sricharan Vadapalli , DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive, , Packt, 2018.

**Online References:**

| Sr. No. | Website Name |
|---|---|
| 1. | https://www.edx.org/course/agile-software-development |
| 2. | https://www.coursera.org/learn/agile-software-development |
| **3.** | https://devops.com/most-popular-open-source-devops-tools/ |
| **4.** | https://github.com/jidibinlin/Free-DevOps-Books-1/blob/master/book/Practical%20DevOps.pdf |

**Assessment:**

**Internal Assessment (IA) for 20 marks:**

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

**NB:**
(1)Question 1 is compulsory.
(2) Solve any three questions out of five.

| | | |
|---|---|---|
| **Q.1** | **Solve any three questions out of five: (05 marks each)** | **15** |

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015116 | Computer Network | 3 | - | - | 3 | - | - | 3 |

| Course Code | Course Name | Theory | | | | | Term Work | Pract / Oral | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | Internal Assessment | | | End Sem Exam | Exam Duration (in Hrs) | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | | |
| 2015116 | Computer Network | 20 | 20 | 40 | 60 | 2 | - | -- | 100 |

**Rationale:**
This course provides foundational knowledge of computer networks, covering architectures, topologies, transmission media, and devices. It develops skills in error control, IP addressing, routing, and socket programming while introducing advanced concepts like VPN, SDN, NFV, and data center networks to prepare students for modern networking applications.

**Course Objectives:**
1. Understand the fundamentals of computer networks, their types, architectures, topologies, and communication models.
2. Explain the design issues, error control, and media access mechanisms used in the Data Link Layer.
3. Analyze addressing, routing techniques, and protocols implemented in the Network Layer for data delivery.
4. Describe transport layer mechanisms including connection establishment, flow control, and congestion management.
5. Illustrate the functionality of common application layer protocols and their role in network communication.
6. Explore emerging networking technologies.

**Course Outcomes:** On successful completion of course, learner will be able to
1. Explain the basic concepts of computer networks, their architectures, topologies, transmission media, and network devices.
2. Demonstrate understanding of Data Link Layer design issues and implement error detection, correction, and flow control mechanisms.
3. Apply IPv4 and IPv6 addressing, subnetting, and routing algorithms for efficient packet delivery across networks.
4. Describe the working of transport layer protocols and implement basic socket programming for process-to-process communication.
5. Analyze the structure and operation of major application layer protocols such as HTTP, DNS, FTP, and SMTP.
6. Evaluate and discuss the concepts of VPN, SDN, NFV, and Data Center Networks as part of advanced networking technologies.

**Prerequisite:** Operating System Fundamentals

**DETAILED SYLLABUS:**

| Sr. No. | Name of Module | Detailed Content | Hours | CO Mapping |
|---|---|---|---|---|
| I | **Introduction To Computer Networks** | Definition, **Types of Networks**: Local area networks (LAN), Metropolitan area networks (MAN), Wide area networks (WAN), Wireless networks, Networks Software, Protocol, Design issues for the Network layers. **Network Models**: The OSI Reference Model, TCP/IP Model, Network Topologies, Types of Transmission Medium. **Network Architectures**: Client-Server, Peer to Peer, Hybrid. **Network Devices**: Bridge, Switch, Router, Gateway, Access Point.<br>**Self-learning Topics:** Study of Campus wide networking | **6** | CO1 |
| II | **Data Link Layer** | Introduction, Design Issues (Services to Network Layer, Framing. Error control, Flow control), Error Detection and correction, Parity Bits, Checksum, Hamming Codes and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocols.<br>Elementary Data Link protocols<br>MAC Sub layer: Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, WDMA, Introduction to IEEE standards (IEEE 802.3, IEEE 802.11).<br>Self-learning Topics: Demonstration of DLL protocols on Simulator | **8** | CO2 |
| III | **Network Layer** | Network Layer design issues, **Switching Techniques**: Circuit switching, Message Switching, Packet Switching. **IP Protocol**: Classes of IP (Network addressing), IPv4 , IPv6, Network Address Translation, Sub-netting, Supernetting, CIDR. **Network layer Protocols**: ARP, RARP, ICMP, IGMP. **Network Routing and Algorithms**: Static Routing, Dynamic Routing, Distance Vector Routing, Link State Routing. **Routing Protocols**: RIP, OSPF, BGP<br>**Self-learning Topics:** Count-to-Infinity Problem, AODV AND DSR routing protocol | **9** | CO 3 |
| IV | **Transport Layer** | Process to Process Delivery, Services, Socket Programming. **Elements of Transport Layer Protocols**: Addressing, Connection establishment, Connection release, CP timers and TCP state transition diagram, Flow control and buffering, Multiplexing, | **07** | CO4 |

| | | | | |
|---|---|---|---|---|
| | | Congestion Control: slow start protocol. **Transport Layer Protocols**: TCP and UDP, Quality of Service (QoS). **Self-learning Topics:** Demonstration of Transport layer protocols on Simulator. | | |
| V | **Application Layer** | Introduction, Web and HTTP, Web Caching, DNS, Email: SMTP, POP3, Webmail, FTP, TELNET, DHCP, SNMP. **Self-learning Topics:** GOOGLE DNS | **04** | CO5 |
| VI | **Emerging & Advanced Topics** | VPN (Virtual Private Networks) - Types and basic principle. **Software-Defined Networking (SDN):** Concepts, Control and Data Planes, OpenFlow protocols and controllers. **Network Function Virtualization (NFV):** Concepts and Benefits. **Data Center Networks (DCNs):** Topologies (Fat Tree - Concept) **Self-learning Topics:** SD-WAN | **05** | CO6 |

**Text Books:**


1. Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw-Hill, Publications, ISBN:0–07 – 058408 – 7
2. Andrew S. Tanenbaum, Computer Networks, 5th Edition, Pearson India, 2012.
3. A. S. Godbole and A. Kahate, Data Communications & Networks, 2nd ed. New Delhi: McGraw Hill Education, 2011.


**References:**


1. Kurose, Ross, "Computer Networking a Top Down Approach Featuring the Internet", Pearson, ISBN-10: 0132856204
2. L. Peterson and B. Davie, "Computer Networks: A Systems Approach", 5th Edition, Morgan-Kaufmann, 2012.
3. Douglas E. Comer & M.S Narayanan, "Computer Network & Internet", Pearson Education


**Online References:**

| Sr. No. | Website Name |
|---|---|
| 1. | NPTEL Swayam Course: https://onlinecourses.nptel.ac.in/noc22_cs19/ |
| 2. | CCNA: Introduction to Networks https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US |

**Assessment:**

**Internal Assessment (IA) for 20 marks:**

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question

must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).

- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

| | | |
|---|---|---|
| **NB:**<br>(1) Question 1 is compulsory.<br>(2) Solve any three questions out of five. | | |
| **Q.1** | **Solve any three questions out of five: (05 marks each)** | **15** |
| a) | | 05 |
| b) | | 05 |
| c) | | 05 |
| d) | | 05 |
| e) | | 05 |
| **Q.2,3, 4,5,6** | | **15** |
| a) | Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module. | 07or 08 |
| b) | | 08 or 07 |

**Note for QP Setters:**

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015117 | Statistics for Machine Learning and Data Science Lab | - | 2 | - | - | 1 | - | 1 |

| Course Code | Course Name | Examination Scheme | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory Marks | | | | Term Work | Practical/ Oral | Total |
| | | Internal assessment | | | End Sem. Exam | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | |
| 2015117 | Statistics for Machine Learning and Data Science Lab | -- | -- | -- | -- | 25 | 25 | 50 |

**Rationale-** The laboratory component reinforces theoretical understanding through hands-on data experiments using real-world datasets and tools such as Python (NumPy, Pandas, Scikit-learn). Students gain skills in exploratory analysis, sampling, hypothesis testing, regression, clustering, and model evaluation. It encourages experiential learning, data storytelling, and critical interpretation, building confidence to apply statistical learning methods for decision-making and AI development.

**Lab Objectives:** After completing the lab, students will be able to:

1. Apply descriptive statistics and sampling to analyze and visualize data.
2. Perform estimation, hypothesis testing, and resampling-based inference.
3. Implement regression and classification models and interpret diagnostic results
4. Apply unsupervised learning methods including PCA and clustering.
5. Evaluate ML models using appropriate metrics and cross-validation.
6. Interpret models with transparency, fairness, and ethical considerations.

**Lab Outcomes:** Upon successful completion, the student will be able to:

1. Perform EDA and sampling-based data exploration.
2. Conduct hypothesis testing and resampling analysis.

3. Implement regression and classification models.
4. Apply clustering and PCA for unsupervised analysis.
5. Evaluate statistical learning models using performance metrics.
6. Ensure interpretability and ethical awareness in statistical modeling.

**Prerequisite:** Python

**Suggested List of Experiments (10-12 Experiments must be conducted)**

| Sr No | List of Experiments | Hrs |
|---|---|---|
| **01** | Exploratory Data Analysis: descriptive stats, histograms, correlation matrix | 02 |
| **02** | Data preprocessing: missing values, encoding, scaling, outlier detection. | 02 |
| 03 | Sampling methods: SRS, stratified, cluster sampling + dataset demonstration | 02 |
| 04 | Demonstration of Central Limit Theorem using simulations. | 02 |
| 05 | Hypothesis testing (t-test, chi-square, ANOVA) on real dataset. | 02 |
| 06 | Bootstrap confidence intervals & permutation tests. | 02 |
| 07 | MLE estimation for Normal & Bernoulli distributions. | 02 |
| 08 | Linear regression & diagnostics (VIF, residual plots, Cook's distance) | 02 |
| 09 | Ridge, Lasso, Elastic Net with hyperparameter tuning. | 02 |
| 10 | Classification models: Logistic regression, KNN, Naïve Bayes, ROC analysis. | 02 |
| 11 | Decision Trees & Random Forest; feature importance. | 02 |
| 12 | PCA + clustering (K-means, hierarchical) | 02 |
| 13 | Interpretability: SHAP/LIME/PDP/ICE (conceptual + coding) | 02 |
| 14 | Fairness & bias detection metrics (equalized odds, parity) | 02 |

| Sr No | List of Assignments / Tutorials | Hrs |
|---|---|---|
| 01 | Mini project on complete ML pipeline (EDA → model → evaluation) | 02 |
| 02 | Comparison of resampling methods (bootstrap vs. permutation) | 02 |
| 03 | Regularization comparison (Ridge / Lasso / ElasticNet) on multiple datasets | 02 |
| 04 | Bias–variance tradeoff study | 02 |
| 05 | Ethical ML case study (fairness, interpretability, transparency) | 02 |

**Students must use at least 3 real-world datasets from Kaggle/UCI/OpenML**
**Common recommended datasets:**

- Iris
- Titanic
- Breast Cancer Wisconsin
- California Housing

- Mall Customers
- Bank Marketing

**Tools Requirement-** Python libraries: NumPy, Pandas, Matplotlib/Seaborn, Scikit-Learn, SciPy, SHAP, LIME
All implementations must follow responsible and ethical data practices.

**Text Books (For Lab Use):**

1. Wes McKinney – *Python for Data Analysis*, O'Reilly.
2. Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly.
3. Peter Bruce & Andrew Bruce – *Practical Statistics for Data Scientists*.
4. James et al. – Introduction to Statistical Learning (ISLR2).

**References:**

1. Charu Aggarwal – *Probability & Statistics for ML*.
2. Scikit-learn documentation (https://scikit-learn.org/stable/).

**Assessment :**
**Term Work:** Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.
**Term Work Marks:** 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

**Practical & Oral Exam:** An Oral & Practical exam will be held based on the above syllabus**.**

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015118 | Artificial Intelligence & Soft Computing Lab | | 2 | - | - | 1 | - | 1 |

| Course Code | Course Name | Examination Scheme | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory Marks | | | | Term Work | Practical/ Oral | Total |
| | | Internal assessment | | | End Sem. Exam | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | |
| 2015118 | Artificial Intelligence & Soft Computing Lab | -- | -- | -- | -- | 25 | 25 | 50 |

**Lab Objectives: Six Lab Objectives**

1. To introduce practical applications of AI fundamentals including intelligent agents, PEAS-based problem formulation, and basic PROLOG programming for knowledge representation and reasoning.
2. To implement classical search algorithms such as BFS, DFS, and heuristic-based A* for solving real-world planning and navigation problems.
3. To simulate foundational Neural Network models such as perceptron classification and explore their role in advanced neural architectures including Transformer-based systems.
4. To explore Generative AI applications including LLM-based text generation along with RAG and CAG enhancements.
5. To design and analyze Soft Computing models including fuzzy sets, membership functions, and fuzzy logic controllers for approximate reasoning and adaptive decision-making.
6. To develop and evaluate Hybrid Neuro-Fuzzy systems integrating neural learning with fuzzy inference mechanisms for intelligent automated solutions.

**Lab Outcomes: Six Lab Outcomes (Based on Blooms Taxonomy)**

1. Explain AI fundamentals and formulate problems using the PEAS model and PROLOG.
2. Apply search algorithms to achieve goal-oriented problem solving.

3. Implement basic neural network learning using perceptron models and interpret Transformer evolution conceptually.
4. Construct LLM-based text generation and enhance it using RAG & CAG techniques.
5. Design and simulate Fuzzy Logic and Neural Network models for adaptive decision making.
6. Create and evaluate a Hybrid Neuro-Fuzzy system for intelligent learning.

**DETAILED SYLLABUS: Syllabus related Lab experiment must be considered and mapped with Blooms Taxonomy.**

**Total six modules for each subject lab (13 weeks) to be distributed among six modules.**

| Sr. No. | Module | Detailed Content | LO Mapping |
|---|---|---|---|
| 0 | **Prerequisite** | **Engineering Mathematics:** Matrix Algebra, **Discrete Structures:** Set Theory and Logic | -- |
| I | **Introduction to Artificial Intelligence and Problem Formulation** | Introduction to AI — applications in engineering; Intelligent agents and PEAS model; Problem formulation in AI systems. Introduction to Data Centers and Cloud Services for AI deployment. Basics of PROLOG — facts, rules, queries; Ethical and Explainable AI. **Self-Learning Topics:** Applications of AI in autonomous systems, PEAS modeling. | **LO1** |
| II | **Search Strategies and Classical AI Problem Solving** | Search-based problem-solving techniques: uninformed (BFS, DFS) and informed (Best-First, A*) algorithms. Problem representation and use of heuristics in goal-driven tasks. Adversarial search: Mini-max Search. **Self-Learning Topics:** Bidirectional & iterative deepening search. | **LO2** |
| III | **Neural Networks Fundamentals** | Biological vs. Artificial neuron; Activation functions; Single-layer Perceptron and forward propagation; Introduction to advanced neural architectures with the basic concept of Transformer models as an evolution of neural network design. **Self-Learning Topics:** Gradient Descent intuition. | **LO3** |
| IV | **Generative AI and Advanced Architectures (LLM + RAG & CAG)** | Introduction to Generative AI workflow. Transformer architecture — Attention mechanism. Language Model Classification: Foundation Models (FM), Large Language Models (LLM), Small Language Models (SLM), Micro Language Models (μLM). Retrieval-Augmented Generation (RAG) using vector databases (FAISS/ChromaDB). Cache-Augmented Generation (CAG) for performance improvement. Text generation/summarization using GPT-2/Mistral/LLaMA-2via HuggingFace/Ollama. **Self-Learning Topics:** Prompt engineering and memory-optimized GenAI pipelines. | **LO4** |
| V | **Soft Computing Techniques — Fuzzy Logic** | Soft computing concepts, uncertainty and approximate reasoning. Fuzzy sets and membership functions; Design of Fuzzy Logic Controller (Mamdani/Sugeno). | **LO5** |

| | | | |
|---|---|---|---|
| | | Implementation of a basic McCulloch–Pitts Neural Network for logic operations (AND/OR).<br>**Self-Learning Topics:** Adaptive fuzzy systems and rule optimization. | |
| VI | **Learning Rules and Hybrid AI Systems** | Basic learning rules: Perceptron, Delta, Hebbian. Concept of Neuro-Fuzzy System- architecture and benefits (Hybrid Neuro-Fuzzy System integrating fuzzy logic with neural learning). Comparative evaluation with standalone models.<br>**Self-Learning Topics:** Applications of hybrid systems in decision support. | **LO6** |



**Text Books:**

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach —Second Edition" Pearson Education.
2. Elaine Rich and Kevin Knight —Artificial Intelligence‖ Third Edition, Tata McGraw-Hill Education Pvt. Ltd., 2008.
3. S. Rajasekaran & G. A. Vijayalakshmi Pai — *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*, PHI, Second Edition
4. J. S. R. Jang, C. T. Sun & E. Mizutani — *Neuro-Fuzzy and Soft Computing,* Pearson Education.

**References:**

1. Ivan Bratko —PROLOG Programming for Artificial Intelligence‖, Pearson Education, Third Edition.
2. S. N. Sivanandam & S. N. Deepa — *Principles of Soft Computing*, Wiley India.
3. Sebastian Raschka — *Build a Large Language Model (From Scratch) (*MIT Press Draft).
4. OpenAI / Hugging Face — *Introduction to Transformers and LLMs (Free Online).*
5. Jay Alammar — *The Illustrated Transformer* (jalammar.github.io).

**Online Resources:**

| Sr. No. | Website Name |
|---|---|
| **1.** | https://www.coursera.org/learn/neural-networks-deep-learning |
| **2.** | https://www.coursera.org/courses?query=generative+ai |
| **3.** | https://www.coursera.org/learn/introduction-to-ai |

**List of Experiments.**

| Sr No | List of Experiments | Hrs |
|---|---|---|
| 01 | Select a problem statement relevant to Artificial Intelligence: Identify the problem, describe it using PEAS (Performance, Environment, Actuators, Sensors), and perform problem formulation | 02 |
| 02 | Introduction to AI Programming Language: Write and execute basic programs in PROLOG. | 02 |
| 03 | Implement uninformed search algorithms (BFS, DFS) to reach a goal state. | 02 |
| 04 | Implement any one of the Informed search techniques E.g. A* algorithm for 8 puzzle problem | 02 |

| | | |
|---|---|---|
| 05 | Implement adversarial search using min-max algorithm. | 02 |
| 06 | Creating/building a neural network using basic language (C Programming/Java Programming) | 02 |
| 07 | Train a perceptron to learn AND, OR, XOR gates. | 02 |
| 08 | To implement Basic Supervised / Unsupervised Neural Network learning rules for a problem | 02 |
| 09 | Exploration and Implementation of Generative AI Techniques using Open-Source Tools and Frameworks | 02 |
| 10 | Developing Language Models ( Small Language Models (SLM), Micro Language Models (μLM).) | 02 |
| 11 | Design and Implementation of a Retrieval-Augmented Generation (RAG) Framework for Contextual Question Answering | 02 |
| 12 | Development of a Cache-Augmented Generation (CAG) System for Efficient AI Responses | 02 |
| 13 | Implement a simple Fuzzy Logic Controller for real-world control problems. | 02 |
| 14 | Implement Mamdani/Sugeno Fuzzy Inference Systems (FIS) using MATLAB / Python. | 02 |
| 15 | Developing a Hybrid Neuro-Fuzzy System | 02 |

| Sr No | List of Assignments / Tutorials | Hrs |
|---|---|---|
| 01 | Applications and Impact of Artificial Intelligence in Modern Engineering | 02 |
| 02 | Role of Heuristics in Enhancing Search Efficiency in AI Systems | 02 |
| 03 | Open-Source Frameworks and Tools for Generative AI Development | 02 |
| 04 | Retrieval-Augmented Generation (RAG): Architecture and Applications | 02 |
| 05 | Cache-Augmented Generation (CAG): Concept and Performance Optimization in AI Systems | 02 |
| 06 | Fuzzy Logic Controllers and Neural Network Approaches for Adaptive Decision-Making | 02 |
| 07 | Design and Applications of Hybrid Neuro-Fuzzy Systems in Intelligent Automation | 02 |
| 08 | Comparative Study of Learning Approaches in Neural Networks and Fuzzy Logic Systems | 02 |

**Assessment:**
**Term Work:** Term Work shall consist of at least 10 to 12 practical's based on the above list.
Also, Term work Journal must include at least 2 assignments.
**Term Work Marks:** 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
**Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.**

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| 2015119 | Agile Software Development & DevOps Lab | - | 2 | - | - | 1 | - | 1 |

| Course Code | Course Name | Examination Scheme | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory Marks | | | | | | |
| | | Internal assessment | | | End Sem. Exam | Term Work | Practical/ Oral | Total |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | |
| 2015119 | Agile Software Development & DevOps Lab | -- | -- | -- | -- | 25 | 25 | 50 |

**Lab Objectives: The student will be able:**

1. To understand and implement the Agile Software Development Lifecycle using Jira and integrate it with DevOps tools for continuous project management.
2. To perform version control operations using Git and apply collaboration workflows for software development.
3. To design and configure Continuous Integration / Continuous Deployment pipelines using Jenkins, GitHub Actions, and containerization tools.
4. To automate provisioning, configuration, container orchestration, and cloud resource management using Ansible, Docker, Kubernetes, and Terraform.

5. To study and implement MLOps workflows including model tracking, deployment, and data pipeline automation using MLflow and Airflow.
6. To integrate and implement a complete end-to-end DevOps pipeline through a mini project covering planning, building, testing, deployment, and monitoring.

**Lab Outcomes: Upon completion, the student will be able to:**

1. Apply Agile methodologies and use Jira for sprint planning, task tracking, and integrating with DevOps workflows.
2. Use Git effectively for version control, source code management, and collaborative team development.
3. Build and manage automated CI/CD pipelines for application development, testing, packaging, and deployment.
4. Containerize, orchestrate, and deploy applications using Docker, Docker Compose, Kubernetes, and cloud infrastructure tools.
5. Implement ML lifecycle management and automated workflows using MLflow, Airflow, and monitoring tools like Prometheus/Grafana/Nagios.
6. Develop and demonstrate an end-to-end DevOps pipeline mini-project integrating all stages of DevOps from code commit to deployment and monitoring.

**DETAILED SYLLABUS:**

| Sr. No. | Module | Detailed Content | LO Mapping |
|---|---|---|---|
| 0 | **Prerequisite** | Knowledge of Programming, Computer Networks, Web Programming. | |
| I | **Agile Project Management & DevOps Integration** | Fundamentals of Agile and Scrum, Jira for Agile project lifecycle management, Creating epics, user stories, tasks, sprints, Planning, assigning, tracking sprint progress, Integrating Jira with CI/CD pipelines Agile reporting: Burndown chart, velocity chart.<br><br>**Self-learning Topics:** Kanban vs Scrum, Agile ceremonies: Sprint review, retrospective | LO 1 |
| II | Version Control & Collaboration using Git | Introduction to Git, GitHub, Basic operations: clone, commit, push, pull, Branching strategies and merging, Git workflows: Feature branching, Git Flow, Handling merge conflicts, GitHub Actions basics<br><br>**Self-learning Topics:** GitHub Issues & Project Boards Webhooks in GitHub | LO 2 |
| III | CI/CD Automation using Jenkins & Containerization using Docker | Understanding CI/CD and DevOps pipelines, Jenkins installation and job creation, GitHub webhook integration, CI pipeline: build → test → package, Docker fundamentals: images, containers, Creating Dockerfiles, Multi-service deployment using Docker Compose<br>**Self-learning Topics:** Jenkinsfile (Pipeline as Code), Publishing images to Docker Hub/ECR | LO 3 |
| IV | Configuration Automation, Orchestration & Cloud Provisioning | Ansible automation: inventory, playbooks, roles, Provisioning and configuring applications, Kubernetes concepts: Pods, Deployment, Service, Ingress, Managing containerized apps in Kubernetes, Terraform basics: providers, resources, state files, Automating cloud infrastructure creation.<br><br>**Self-learning Topics:** Helm Charts, Terraform Cloud / Remote | LO 4 |

| | | | |
|---|---|---|---|
| | | state management | |
| V | Monitoring, Serverless Computing & MLOps Fundamentals | Monitoring tools: Prometheus, Grafana, Nagios, Exporters, dashboards, alerting, AWS Lambda fundamentals and deployment, MLOps lifecycle overview, MLflow for tracking experiments, versioning & deployment, Airflow basics: DAGs, tasks, operators, scheduling<br><br>**Self-learning Topics:** CloudWatch metrics for serverless apps, Airflow for ML/ETL automation | LO 5 |
| VI | End-to-End DevOps Pipeline | Designing a full DevOps automation pipeline, Integrating Agile → Git → CI → CD → Containers → Kubernetes → Monitoring, Cloud deployment using AWS/GCP/Azure, Pipeline security basics (DevSecOps), Documentation, demonstration & evaluation. **Self-learning Topics:** Industry-grade DevOps architecture, Cost-efficient cloud deployment strategies | LO 6 |

**Text Books:**


1. The DevOps handbook : how to create world-class agility, reliability, and security in technology organizations by Kim, Gene.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint.


**References**


1. DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive . By Sricharan Vadapalli, Packt, 2018


**Online Resources:**

| Sr. No. | Website Name |
|---|---|
| 1. | https://devops.com/most-popular-open-source-devops-tools/ |
| 2. | https://github.com/bregman-arie/devops-resources? |
| 3. | https://seblum.github.io/mlops-engineering-book/? |

**List of Experiments.**

| Sr No | List of Experiments | Hrs |
|---|---|---|
| 01 | To study and Implement Agile Software Project lifecycle using Jira with DevOps Pipeline Integration. | 2 |
| 02 | To perform version control operations using Git. | 2 |
| 03 | To set up a Continuous Integration Pipeline using Jenkins/GitHub Actions. | 2 |
| 04 | To automate software provisioning and configuration using Ansible | 2 |
| 05 | To containerize applications using Docker. | 2 |
| 06 | To deploy multi-service applications using Docker Compose. | 2 |
| 07 | To deploy and manage containers using Kubernetes. | 2 |
| 08 | To automate application deployment using Github/ Jenkins. | 2 |
| 09 | To provision cloud infrastructure using Terraform. | 2 |
| 10 | To deploy a serverless function using AWS Lambda. | 2 |
| 11 | To implement monitoring using Prometheus / Grafana / Nagios. | 2 |

| | | |
|---|---|---|
| 12 | To study and Implement machine learning lifecycle with model development & deployment using MLFlow. | 2 |
| 13 | To study and Implement designing, scheduling, and monitoring workflows and data pipelines using AirFlow. | 2 |
| 14 | To design and implement an end-to-end DevOps pipeline through a mini project. | 2 |

**List of Assignments (Based on Theory Syllabus):**

| Sr No | List of Assignments / Tutorials | Hrs |
|---|---|---|
| | **Assignment 1** | |
| 01 | Choose any one application domain (E-commerce / Healthcare / College ERP / Banking / Food Delivery) and Create Agile Requirements using User Stories & Acceptance Criteria. | 2 |
| 02 | For the user stories created in Assignment 1 do Sprint Planning, Story Point Estimation & Release Planning | |
| 03 | Test Case Design using Agile QA Techniques | |
| | **Assignment 2** | |
| 01 | DevOps Pipeline – CI/CD Setup using GitHub Actions / Jenkins. | 2 |
| 02 | Simulate a Full Agile Process Implementation Using Jira. | |
| 03 | Integrate Agile Planning with DevOps Automation. | |

**Assessment :**

**Term Work:** Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.

**Term Work Marks:** 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

**Practical & Oral Exam:** An Oral & Practical exam will be held based on the above syllabus**.**

| Subject Code | Subject Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Practical | Tutorial | Theory | Practical & Oral | Tutorial | Total |
| 2015120 | Web and Mobile Application Development Lab | -- | 2 | -- | -- | 1 | -- | 1 |

| Subject Code | Subject Name | Examination Scheme | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory Marks | | | | Term Work | Practical / Oral | Total |
| | | Internal assessment | | | End Sem. Exam | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | |
| 2015120 | Web and Mobile Application Development Lab | -- | -- | -- | -- | 25 | -- | 25 |

| | | |
|---|---|---|
| | i. a cube, plane and sphere, apply transformations on the 3 game objects.<br>ii. add a video and audio source | |
| 13 | Understand basic VR environment design using Unity's *Play Mode* (desktop simulation). | 02 |
| 14 | Prototype VR Interaction using Mouse and Keyboard | 02 |
| 15 | Evaluate VR Usability using Video/Screen-based Heuristic Analysis | 02 |

| **Useful Links** | |
|---|---|
| 1 | https://www.coursera.org/professional-certificates/google-ux-design |
| 2 | https://nptel.ac.in/courses/124107008 |
| 3 | https://www.coursera.org/learn/develop-augmented-virtual-mixed-extended-reality-applicatio ns-webxr-unity-unreal |
| 4 | https://tih.iitr.ac.in/AR-VR.html |

| **Sr No** | **Suggested List of Assignments / Tutorials** | **Hrs** |
|---|---|---|
| | **Assignment 1** | |
| 01 | While creating your project proposal, how did you identify the **main user problem** and define measurable **user goals**? Give an example from your project. | 2 hrs |
| 02 | How did the **personas** you created influence your design decisions and the features included in your prototype? | |
| 03 | When designing your **wireframe**, what layout or navigation issues did you face, and how did you resolve them? | |
| | **Assignment 2** | |
| 01 | Describe the process you followed to create your **low-fidelity prototype in Figma**. How did it help before moving to the high-fidelity version? | 2 hrs |
| 02 | During **usability testing**, what kind of feedback did you receive from users, and what changes did you make based on that feedback? | |
| 03 | In your **Unity VR project**, explain how you applied transformations (position, rotation, scale) to objects and added multimedia components like audio or video. | |

**Assessment:**
**Term Work:** Term Work shall consist of at least 10 to 12 practical's based on the above list.
**Term Work Marks:** 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)



| **Course Code** | **Course Name** | **Teaching Scheme (Contact Hours)** | | | **Credits Assigned** | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Theory** | **Pract.** | **Tut.** | **Theory** | **Pract.** | **Tut.** | **Total** |
| **2015122** | **Computer Network Lab** | | **2** | **-** | | **1** | **-** | **1** |

| Course Code | Course Name | Examination Scheme | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory Marks | | | | Term Work | Practical/ Oral | Total |
| | | Internal assessment | | | End Sem. Exam | | | |
| | | IAT-I | IAT-II | IAT-I + IAT-II | | | | |
| 2015122 | Computer Network | -- | -- | -- | -- | 25 | -- | 25 |

**Lab Objectives:**

1. To introduce students to networking components, commands, cables, and connectors essential for building and troubleshooting networks.
2. To enable students to design and simulate various network topologies and test their connectivity using modern simulation tools.
3. To develop understanding of data communication protocols and techniques such as sliding window and error detection/correction methods.
4. To provide hands-on experience in configuring routing protocols and core network services like FTP and DNS.
5. To enhance students' skills in network analysis, traffic monitoring, and performance evaluation using tools like Wireshark.
6. To acquaint students with advanced networking concepts such as VLANs, NAT, and Software Defined Networking (SDN).

**Lab Outcomes:**

1. Identify and use different network cables, connectors, and commands for establishing and troubleshooting network connections.
2. Design and simulate wired LANs and complex topologies, performing connectivity tests and file sharing between devices.
3. Implement and analyze data transmission techniques including sliding window protocols and error detection/correction algorithms.
4. Configure and verify static and dynamic routing protocols (RIP, OSPF) and network services like FTP and DNS.
5. Capture, analyze, and interpret network traffic using Wireshark to evaluate protocol behavior and network performance.
6. Design and simulate VLANs, NAT, and SDN topologies to understand modern network architectures and their applications.

**List of Experiments.**

| Sr No | List of Experiments | LO |
|---|---|---|
| 01 | To study different networking commands and network cables & connectors. | LO1 |
| 02 | Simulate a wired LAN using networking devices and test connectivity using PING utility. | LO1 |
| 03 | Create different network topologies using any network simulator (e.g. Cisco Packet Tracer, GNS3). For each topology perform connectivity tests such as ping, file sharing between devices. | LO2 |
| 04 | Simulate sliding window protocols using any packet tracer tool (any one from Stop and Wait, Go-Back-N, Selective Repeat) | LO3 |
| | | |
| 05 | Implementation of error detection and correction techniques (any one from CRC, Hamming Code, Checksum) using any programming language. | LO3 |
| 06 | To analyze network traffic using the Wireshark tool. Perform following activities: | LO2, LO4, LO5 |

| | | |
|---|---|---|
| | • Packet capture and filtering<br>• Analyze ethernet frames<br>• IP packet analysis<br>• ICMP analysis (PING test)<br>• HTTP/HTTPs request analysis<br>• Analyze header of TCP /UDP dump.. | |
| 07 | To configure and verify static routing and dynamic routing protocols (RIP and OSPF) in a simulated network environment using Cisco Packet Tracer. | LO4 |
| 08 | To implement Socket programming using any programming language. | LO4, LO5 |
| 09 | To configure FTP server and transfer files to demonstrate the working of the same. | LO5 |
| 10 | To configure the DNS server and to demonstrate the working of the same. | LO5 |
| 11 | To design and simulate VLANs on switch/router using any network simulator (e.g. Cisco Packet Tracer, GNS3) | LO6 |
| 12 | To design and simulate NAT on router using any network simulator (e.g. Cisco Packet Tracer, GNS3) | LO6 |
| 13 | Implementation of small SDN based topology using Mininet. | LO6 |

**Online Resources:**

| Sr. No. | Website Name |
|---|---|
| 1. | NPTEL Swayam Course: https://onlinecourses.nptel.ac.in/noc22_cs19/ |
| 2. | CCNA: Introduction to Networks https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US |

| Sr No | List of Assignments / Tutorials | Hrs |
|---|---|---|
| **01** | Assignment covers the topics from the first three units. | CO1,CO2,CO3 |
| **02** | Assignment covers the topics from the last three units. | CO4,CO5,CO6 |

**Assessment :**
**Term Work:** Term Work shall consist of at least 10 to 12 practicals based on the above list. Also, Term work Journal must include at least 2 assignments.
**Term Work Marks:** 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

# IKS

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract. | Tut. | Theory | Pract. | Tut. | Total |
| IKS501 | Indian Knowledge System | -- | 2*+2 | -- | -- | 2 | - | 2 |

| Course Code | Course Name | Theory | | | | | Term work | Pract / Oral | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | Internal Assessment | | | End Sem Exam | Exam Duration (in Hrs) | | | |
| | | Test 1 | Test 2 | Total | | | | | |
| IKS501 | Indian Knowledge System | -- | -- | -- | -- | -- | 50 | -- | 50 |

*** Two hours of practical class to be conducted for full class as demo/discussion.**

**Rationale:**

The proposed IKS modules contextualize core engineering principles across Computer, IT, EXTC, Electronics, and Electrical engineering—within scientifically analyzable Indian knowledge traditions. By mapping indigenous practices to modern technical frameworks, students critically examine foundational engineering concepts through interdisciplinary and systems-based perspectives. This approach strengthens conceptual understanding, promotes analytical reinterpretation of traditional knowledge using contemporary tools, and aligns engineering education with culturally rooted yet technically rigorous inquiry.

**Course Objectives:**

1. Provide foundational knowledge of Indian acoustic practices.
2. To Understand Ancient Indian use of renewable energy.
3. Introduce Sanskrit grammar in simple language, with special attention to its rule-based structure.
4. Learners should be able to explain key contributions of ancient Indian mathematics in simple language.
5. Understand how Indian traditions of logic, language, knowledge organization, design thinking, governance, sustainability, and wellbeing can support modern IT solutions.
6. Understand the basic structure of Sanskrit grammar and relate selected grammatical ideas to computational modules used in information technology.

**Course Outcomes:**

**On successful completion of the course learner will be able to:**

1. Analyze traditional Indian acoustic practices and interpret their relevance to frequency analysis.
2. Apply the traditional knowledge to modern energy solutions and sustainable engineering.
3. Apply Indian Knowledge Systems in linguistics, Sanskrit-based NLP, traditional number systems, measurement practices, and ancient rainfall prediction techniques using calendars, nakshatras, and natural indicators.
4. Apply basic modular arithmetic, pattern recognition, and place-value ideas to simple IT examples such as coding, data representation, and checksums.
5. Apply basic concepts of Indian Knowledge Systems such as logic, language, knowledge organization, ethics, sustainability, and responsible decision-making to design simple technology-based solutions like apps, chatbots, databases, or awareness tools.
6. Apply basic Sanskrit grammar concepts and Pāṇinian rule-based principles to understand formal grammar, algorithms, and simple computational language processing applications in IT.

**DETAILED SYLLABUS:**

| Sr. No. | Name of Module | Detailed Content | Hours | CO Mapping |
|---|---|---|---|---|
| I | **Acoustic Science in Vedic Chanting and Indian Music** | I.1 Concept of Nāda (Sound) and Shruti<br>• Physical interpretation: frequency, pitch, harmonic content<br>• Shruti system (22 microtones) and frequency discrimination | **10** | **CO1** |

| | | | | |
|---|---|---|---|---|
| | | I.2 Resonance and Architectural Acoustics<br>• Temple architecture and echo control<br>• Natural acoustic amplification<br>• Relation to reverberation time and standing waves<br>I.3 Structured Sound Encoding in Vedic Recitation<br>• Pada, Krama, Ghana patha recitation methods<br>• Built-in redundancy → error detection and correction<br>• Parallel to digital communication redundancy schemes<br><br>**Self-Study:**<br>Study of sounds and analysis in time and frequency domains. | | |
| **II** | **Tradition** | 1. Introduction to IKS in renewable energy<br>2. Ancient Indian use of renewable energy<br>a) Solar Energy: Solar Architecture, Solar Heating Systems<br>b) Wind Energy: Ancient Windmills, Wind-Powered Water Lifting<br>c) Water (Hydro) Energy: Ancient Watermills, Irrigation Systems<br>d) Bioenergy: Biomass Fuel, Bioenergy Production<br>3. Integration of ancient Traditional knowledge of renewable energy with Modern Technologies.<br><br>**Students shall study the following to understand the use of ancient technologies in Renewable Energy:**<br>• Gujarat Windmills<br>• Watermills in the Indus Valley<br>• Konark Sun Temple | **10** | **CO2** |
| **III** | **Linguistics, Number Systems, and rainfall Prediction** | Linguistics:<br>• Components of language<br>• Panini's work on Sanskrit grammar<br>• Role of Sanskrit in Natural Language Processing<br>Number System and Units of Measurement<br>• Number system in India- Historical Evidence<br>• Decimal System<br>• Unique Approaches to represent Numbers-Katapayadi System<br>• Pingala and the Binary System<br>Ancient Indian techniques for predicting Rainfall<br>• Introduction<br>• Elements of the Indian Calendar<br>• Panchang and Nakshtras<br>• Non-bio indicators and Bio indicators<br>**Self-Learning:**<br>• An Ecosystem for Sanskrit Language Processing<br>• Salient Features of the Indian Numeral System<br>• Salient Features of the Indian Numeral System<br>• Measurement for Time Distance and Weight<br>• Measurement of Time- An Illustration from Purana | **10** | **CO3** |
| **IV** | **Mathematics Foundations in Ancient India and Its Relevance to Information Technology** | 4.1 Historical context and number ideas: Indian knowledge systems, the development of numeration, zero as placeholder and number, decimal place value.<br>4.2 Major contributors and procedural mathematics: Aryabhata, Brahmagupta, Bhaskara II, and examples of rule-based or sutra-like calculation methods in plain language. | **10** | **CO4** |

| | | | | |
|---|---|---|---|---|
| | | 4.3 Algorithmic thinking and computation: Historical mathematical procedures connect to algorithms, flowcharts, decomposition, and computational logic.<br>4.4 Modular arithmetic, binary-style ideas, and data representation: Clock arithmetic, remainders, cyclic reasoning, two-state representation, bits, bytes, and simple binary conversion.<br>4.5 Relevance to IT and synthesis: Coding, checksums/parity, simple cryptography idea, data representation, and how ancient mathematical thinking supports modern IT problem-solving.<br><br>**Self-Learning Topics and Activities:**<br>• Zero and place value in daily life<br>• Algorithmic thinking<br>• Modular arithmetic<br>• Binary representation | | |
| **V** | **Indian Logic and its applications in Information Technology** | 5.1 Introduction to IKS and its relevance to contemporary technology.<br>5.2 Indian logic, language, and computation with simple examples.<br>5.3 Knowledge organization, data, and decision-making.<br>5.4 Ethics, society, sustainability, and responsible digital systems.<br>5.5 Mini activity where students design a small app, chatbot, database, or awareness tool using one IKS principle.<br><br>**Self-Learning Topics and Activities:**<br>• Basic ideas of Nyaya logic: observation, inference, debate, and evidence.<br>• Introduction to Paninian grammar and why structured language matters in NLP.<br>• Traditional knowledge classification and its connection with databases and knowledge graphs.<br>• Technology for Indian languages, heritage preservation, digital museums, and public services.<br>• Design thinking activity: identify a local problem and propose an IKS-inspired IT solution. | **10** | **CO5** |
| **VI** | **Sanskrit Grammar and Computational Modules** | 6.1 Introduction to Indian Knowledge Systems, Sanskrit as structured language, and relevance to IT.<br>6.2 Basic Sanskrit grammar concepts: sounds, words, roots, suffixes, sandhi, and sentence structure.<br>6.3 Pāṇinian grammar as a rule-based system; relation with algorithms and formal grammar.<br>6.4 Computational modules: tokenization, morphology, parsing, language rules, and NLP examples.<br>6.5 Demonstration of Sanskrit digital tools, reflection activity, and discussion on interdisciplinary applications.<br><br>**Self-Learning Topics and Activities:**<br>• Basic Sanskrit alphabet and sound classification.<br>• Simple sandhi examples and word joining rules.<br>• Introduction to Pāṇini and the Aṣṭādhyāyī.<br>• Dhātu, pratyaya, pada, and kāraka in simple terms.<br>• Basics of natural language processing: tokenization, morphology, and parsing.<br>• Exploration of Sanskrit digital dictionaries and Sanskrit reader tools. | **10** | **CO6** |

| | | | | |
|---|---|---|---|---|
| | | • Comparison of grammar rules and computer program rules. | | |

**Text and Reference Books or Articles:**


1. Acoustics of Chants, Conch-Shells, Bells & Gongs in Worship Spaces. Available: https://www.academia.edu/38208102/Acoustics_of_Chants_pdf?utm_source=chatgpt.com

2. V. R. S. Nalluri, V. J. K. S. Sonti, and G. Sundari, "Analysis of frequency dependent Vedic chanting and its influence on neural activity of humans," International Journal of Reconfigurable and Embedded Systems, vol. 12, no. 2, pp. 230–239, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp230-239.

3. The Garland Encyclopedia of World Music: South Asia. Available: https://api.pageplace.de/preview/DT0400.9781351544399_A30892028/preview-9781351544399_A30892028.pdf?utm_source=chatgpt.com

4. J. Pattanayak, Y. Belur, and J. Ilavarasu, "Acoustic and psychological effects of Omkar chanting in novice Indian adults: A quasi-experimental study," Journal of Clinical and Diagnostic Research, 2025, doi: 10.7860/JCDR/2025/77365.21111. Available: https://www.jcdr.net/articles/PDF/21111/77365_CE%5BRa1%5D_F%28IS%29_QC%28PS_SS%29_PF1%28AB_IS%29_redo_PFA%28IS%29_PB%28AB_IS%29_PN%28IS%29.pdf?utm_source=chatgpt. com

5. "Indian Knowledge System: Integrating Heritage with Engineering" by Gagan Bansa, Deep Science Publishing; ISBN: 978-93-49307-77-3 E-ISBN: 978-93-49307-29-2;

6. The textbook of Indian Knowledge Systems, University of Mumbai, ISBN: 978-93-5915-553-1

7. Introduction to Indian Knowledge System: Concepts and Applications, Mahadevan, B., Bhat, Vinayak Rajat, Nagendra Pavana R.N. by PHI Learning Pvt Ltd. Delhi, ISBN: 978-93-91818-20-3

8. Echoes of the Monsoon: Ancient Indian techniques for predicting Rainfall, Janak Joshi, Kandarp

9. Parmar, Lakhan Jain, Sharmistha Bhowmik and Bindu Bhatt, DOI: https://www.doi.org/10.22271/27067483.2025.v7.i7a.386.

10. Kim Plofker, Mathematics in India: A clear historical overview of Indian mathematical developments, suitable for faculty use and advanced self-learners.

11. George Gheverghese Joseph, The Crest of the Peacock: An accessible introduction to non-European mathematical traditions, including India.

12. Charles Petzold, Code: The Hidden Language of Computer Hardware and Software: A beginner-friendly bridge from numbers and symbols to data representation and computing.

13. Simon Singh, The Code Book: Useful for simple explanations of ciphers, encryption, and the mathematics behind communication security.

14. T. A. Sarasvati Amma, Geometry in Ancient and Medieval India: A more specialized reference for deeper exploration of traditional mathematical methods.

15. Mahadevan, B., Bhat, V. R., and Nagendra Pavana R. N. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning.

16. Kapur, K., and Singh, A. K. (Eds.). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla.

17. Bora, M. C. Bharatiya Knowledge Systems. Khanna Publishing House.

18. Rao, S. S. Engineering and Technology in Ancient India. Indian National Science Academy / related academic editions.

19. Selected open articles or chapters from IKS Division / AICTE / Ministry of Education resources may be used as supplementary reading.

20. S. D. Joshi and J. A. F. Roodbergen, The Aṣṭādhyāyī of Pāṇini.

21. V. N. Jha, The Philosophy of Sanskrit Grammar.

22. Peter M. Scharf, Sanskrit Syntax and Discourse Structures.

23. Daniel Jurafsky and James H. Martin, Speech and Language Processing — for introductory NLP concepts.

24. Rajeev Sangal, Akshar Bharati, and Vineet Chaitanya, Natural Language Processing: A Paninian Perspective.

**Other Online Resources:**

| Sr. No. | Website Name |
|---|---|
| **1.** | NPTEL course page: For a structured academic overview of mathematics in India. https://nptel.ac.in/courses/111101080 |
| **2.** | NPTEL YouTube playlist: For lecture-based learning that can support flipped or self-paced study. https://www.youtube.com/playlist?list=PLbMVogVj5nJThf31TNSQzuN7zqxe7HdRN |
| **3.** | IKS India: For broader Indian Knowledge Systems context and additional learning resources. https://www.nist.gov/cryptography |
| **4.** | Responsible AI and India-context resources: https://indiaai.gov.in/ |
| **5.** | IKS Division, Ministry of Education / AICTE: https://iksindia.org/ |
| **6.** | NPTEL: Introduction to Pāṇinian Grammar<br>URL: https://onlinecourses.nptel.ac.in/noc24_hs44/preview |
| **7.** | Sanskrit Heritage Reader URL: https://sanskrit.inria.fr/ |

**List of Practical and List of Assignments:**

| Sr No | List of Experiments | Hrs |
|---|---|---|
| 01 | Record a temple bell sound, conch sound, or classical "Sa" note and analyze it by plotting its time-domain waveform and frequency spectrum, identifying the fundamental frequency and harmonics, and comparing the observed harmonic structure with theoretical frequency ratios of Just Intonation and Equal Temperament. | 04 |
| 02 | Observe the movement of sunlight and shadow around a building, monument, or open courtyard at different times of the day, record the shadow directions and lengths, and analyze how solar position can guide orientation, time estimation, and passive architectural planning in the context of traditional Indian solar design. | 04 |
| 03 | Examine images, plans, or a simple model of the Konark Sun Temple or any sun-oriented structure, identify how geometry, orientation, and sunlight interaction are used in the design, and prepare a visual explanation of how ancient solar architecture supports modern sustainable building concepts. | 04 |
| 04 | Construct a small paper or cardboard windmill, test its rotation under natural wind or fan airflow at different speeds, and document how blade movement demonstrates the basic principle of converting wind energy into mechanical motion, linking it to traditional wind-powered systems and modern turbines. | 04 |
| 05 | Build a simple working model using cups, straws, thread, or cardboard to simulate water lifting or channel-based irrigation, observe how water can be moved with minimal energy input, and relate the model to ancient Indian irrigation systems and watermill-based practices. | 04 |
| 06 | Survey homes or nearby community spaces to identify traditional biomass fuels such as wood, agricultural residue, or cow-dung cakes, record their current use, benefits, and limitations, and compare the observed practices with modern ideas of bioenergy and | 04 |

| | | |
|---|---|---|
| | sustainable fuel management.. | |
| 07 | Pronounce and classify selected Sanskrit sounds into vowels and consonants using a simple sound chart, group them according to articulation patterns, and analyze how this systematic sound organization supports pronunciation training, language learning, and speech-processing applications. | 04 |
| 08 | Create all possible combinations of short and long syllabic patterns of a fixed length, tabulate them systematically, and compare the resulting patterns with binary 0/1 combinations to understand how ancient combinatorial thinking connects with digital representation. | 04 |
| 09 | Solve real-life problems based on clocks, calendars, weekdays, or repeated events using modular arithmetic, record the remainder-based reasoning used in each case, and explain how cyclic patterns support time management, scheduling, and introductory cryptographic logic. | 04 |
| 10 | Convert selected decimal numbers into binary form, arrange them in bit patterns, append parity bits for even or odd parity, and interpret how binary representation and simple error-detection methods are used in digital systems for reliable data storage and transmission. | 04 |
| 11 | Use a basic Sanskrit dictionary, transliteration tool, or reader tool to input selected words, observe the output such as root, meaning, or grammatical analysis, and describe how such tools support language technology and digital preservation. | 04 |
| 12 | Select a real-life problem such as fake-news verification, waste segregation, or campus awareness, identify observation, evidence, inference, and conclusion using the basic framework of Indian logic, and represent the reasoning process as a decision tree, chatbot flow, or awareness tool concept. | 04 |
| 13 | Take a simple historical or rule-based calculation procedure associated with Indian mathematics, rewrite it as a sequence of input-process-output steps, and convert it into a flowchart to demonstrate the connection between procedural mathematics and modern algorithmic thinking. | 04 |
| 14 | Represent selected large and small numbers using a place-value table, deliberately insert or remove zero in different positions, and analyze how zero functions both as a number and as a placeholder, showing its importance in arithmetic, accounting, and digital data representation. | 04 |

| Sr No | List of Assignments | Hrs |
|---|---|---|
| 01 | Study the concepts of Nāda and Shruti, temple acoustics, and Vedic recitation methods, and briefly explain their relation to modern concepts of frequency, resonance, reverberation, and redundancy/error correction in digital communication systems. Also write a short note on how Indian Knowledge Systems demonstrate scientific understanding of sound and communication. | 01 |
| 02 | Study the ideas of solar architecture in traditional Indian structures such as the Konark Sun Temple or courtyard-based houses, and explain how sunlight, orientation, shading, and ventilation were used for thermal comfort and energy efficiency. Also write a short note on how these traditional design ideas are relevant to modern sustainable architecture and climate-responsive planning.. | 01 |
| 03 | Study one traditional Indian renewable energy practice such as wind-powered water lifting, watermills, solar heating, or biomass use, and compare it with one modern renewable energy technology. Explain how traditional knowledge systems supported sustainable living and discuss how these ideas can inspire present-day energy conservation and green technology. | 01 |
| 04 | Study the basic ideas of Pāṇinian grammar as a structured rule-based system, and explain how concepts such as sounds, roots, suffixes, and word formation relate to modern ideas in natural language processing, machine translation, and language analysis. Also write a short note on why Sanskrit grammar is often considered relevant to computational linguistics. | 01 |
| 05 | Study the historical development of zero as both a number and a placeholder, and explain how the decimal place-value system simplified arithmetic and record keeping. Also write a short note on how zero and place value continue to be essential in digital computation, data storage, and computer-based numerical processing. | 01 |

| | | |
|---|---|---|
| 06 | Study the basic ideas of Indian logic, especially observation, inference, debate, and evidence in the Nyāya tradition, and explain how these ideas support careful reasoning and decision-making. Also write a short note on how such logical methods can be applied in present-day contexts such as data analysis, fake-news verification, ethical decision-making, and responsible digital systems. | 01 |
| 07 | Identify one idea from Indian Knowledge Systems in the syllabus, such as renewable energy, Indian numeration, Indian logic, or Sanskrit grammar, and propose a simple real-life solution such as an app, chatbot, awareness campaign, digital archive, poster, database, or educational tool based on that idea. Explain the problem being addressed, the IKS principle used, and how the proposed solution can be useful in present-day society. | 01 |
| 08 | Study the basic Sanskrit grammar concepts of sounds, words, roots, suffixes, sandhi, and sentence structure, and explain how grammar rules work like structured instructions. Also compare grammar rules with computer program rules, and briefly discuss how tokenization, morphology, and parsing in computational systems resemble traditional rule-based language analysis. | 01 |

**Assessment:**

**Term Work: Term work shall consist of at least 10-12 practical's based on above list. Also Term Work Journal must include at least 4 assignments**
**Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 20 Marks (Assignments) + 5 Marks (Attendance)**