Madan Mohan Malaviya University of Technology, Gorakhpur (U.P.) India

मदन मोहन मालवीय प्रौद्योगिकी विश्वविद्यालय, गोरखपुर (उ.प्र.) भारत

Established by U.P. Act No. 22 of 2013 of U.P. Government
(Formerly Madan Mohan Malaviya Engineering College)

Department of Computer Science and Engineering
Minor Degree Courses (Started from Session 2025-26)

The department of Computer Science and Engineering offers the following minor degree courses for the students of undergraduate B. Tech. program of other departments in MMMUT Gorakhpur.

Minor Degree Courses offered by CSED (For the B.Tech. students of other Department)

1. For holistic development of the students and as per NEP-2020 and AICTE guideline, the students may earn additional 20 credits through the minor degree courses offered by different departments of the University from Semester IV to VIII.
2. Minor degree courses are optional, but it will be helpful to align the need of industries.
3. Students can only opt for one minor degree course during his/her studies of the B. Tech. program
4. If students complete all 5 PE (professional elective) category courses offered for the minor degree (total 20 credit) from the other department for minor degree, he/shell get a B. Tech. degree in his/her own branch.
5. No extra fee for a minor degree course will be charged by the students
6. In case if a student is unable to complete all 5 PE courses as offered by the other department for minor degree at the time of completion of B. Tech. program in his/her own branch then student will get B. Tech. degree in his/her own branch without completing the minor degree course from other Department.
7. The minor degree course may be offered by the department through MOOC, as per the guidelines in B. Tech. ordinance 3.0 for the MOOC course.

Minor Degree 1: Data Science

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-103	Introduction to Data Science	3	1	0	4
PE3	ECS-303	Computing using Python	3	0	2	4
PE5	ECS-503	Data Visualization	3	1	0	4
PE7	ECS-703	Statistical Learning for Reliability Analysis	3	1	0	4
PE9	ECS-903	Reinforcement Learning	3	0	2	4
	ECS-913	Deep Learning	3	0	2	4
Total						20

Minor Degree 2: Artificial Intelligence & Machine Learning

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-104	Introduction to Artificial Intelligence	3	1	0	4
PE3	ECS-304	Introduction to Machine Learning	3	0	2	4
PE5	ECS-504	Artificial Intelligence: Knowledge Representation and Reasoning	3	1	0	4
PE7	ECS-704	Natural Language Processing	3	0	2	4
PE9	ECS-904	Optimization and Machine Learning	3	1	0	4
	ECS-914	Deep Learning and Neural Network	3	0	2	4
Total						20

Minor Degree 3: Cyber Security & Cyber Physical Systems (CPS)

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-105	Cryptography and Network Security	3	1	0	4
PE3	ECS-305	Practical Cyber Security for Cyber Security Practitioners	3	0	2	4
PE5	ECS-505	Cyber Security and Privacy	3	1	0	4
PE7	ECS-705	Introduction to Security of Cyber-Physical Systems	3	1	0	4
PE9	ECS-905	Ethical Hacking	3	0	2	4
	ECS-915	Computer Forensics	3	0	2	4
Total						**20**

Minor Degree 4: Computer Graphics and Virtual Reality

Category	**Subject Code**	**Name of Subject**	**Credit**			**Total Credit**
			L	**T**	**P**	
PE1	ECS-106	Computer Graphics for Virtual Reality	3	1	0	4
PE3	ECS-306	Computer Graphics and Game Programming	3	0	2	4
PE5	ECS-506	Concepts of Virtual and Augmented Reality	3	1	0	4
PE7	ECS-706	Scientific and Engineering Data Visualization	3	1	0	4
PE9	ECS-906	Interactive 3D Graphics	3	0	2	4
	ECS-916	Sensors and actuators in AR/VR system	3	0	2	4
Total						20

Minor Degree 5: Blockchain and Cryptocurrency

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-107	Fundamental of Blockchain Technology and Applications	3	1	0	4
PE3	ECS-307	Private and Consortium Blockchain for Enterprise	3	1	0	4
PE5	ECS-507	Smart Contracts and Decentralized Applications	3	1	0	4
PE7	ECS-707	Introduction to Solidity Language	3	0	2	4
PE9	ECS-907	Blockchain for Social Impact: Democracy & Governance	3	1	0	4
	ECS-917	Blockchain Security and Next Generation Applications	3	1	0	4
					Total	20

Minor Degree 6: Software Engineering

Category	**Subject Code**	**Name of Subject**	**Credit**			**Total Credit**
			L	**T**	**P**	
PE1	ECS-108	Introduction to C Programming	3	0	2	4
PE3	ECS-308	Data Structures and Algorithms using C	3	0	2	4
PE5	ECS-508	Introduction to Software Engineering	3	1	0	4
PE7	ECS-708	Software Product Management	3	1	0	4
PE9	ECS-908	Software Testing and Quality Assurance	3	1	0	4
	ECS-918	Agile Software Processes	3	1	0	4
Total						20

Minor Degree 7: Spatial Information Technology

Category	**Subject Code**	**Name of Subject**	**L**	**T**	**P**	**Total Credit**
PE1	ECS-109	Global Positioning System	3	1	0	4
PE3	ECS-309	Principle of Remote Sensing	3	0	2	4
PE5	ECS-509	Spatial Information Technology	3	0	2	4
PE7	ECS-709	Web Technology for Spatial Information Mapping	3	0	2	4
PE9	ECS-909	Satellite Image Processing	3	1	0	4
	ECS-919	Spatial Data Quality Assessment	3	1	0	4
					Total	20

Minor Degree 1: Data Science

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-103	Introduction to Data Science	3	1	0	4
PE3	ECS-303	Computing using Python	3	0	2	4
PE5	ECS-503	Data Visualization	3	1	0	4
PE7	ECS-703	Statistical Learning for Reliability Analysis	3	1	0	4
PE9	ECS-903	Reinforcement Learning	3	0	2	4/
	ECS-913	Deep Learning	3	0	2	4
Total						20

INTRODUCTION TO DATA SCIENCE

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-103	Introduction to Data Science	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:

1. To provide the knowledge and expertise to become a proficient data scientist.
2. Demonstrate an understanding of statistics and machine learning concepts that are vital for data science.
3. Produce Python code to statistically analyse a dataset.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. To explain how data is collected, managed, and stored for data science.
2. To understand the key concepts in data science, including their real-world applications and the toolkit used by data scientists.
3. To implement data collection and management scripts using MongoDB.
4. Critically evaluate data visualisations based on their design and use for communicating stories from data.
5. Evaluate outcomes and make decisions based on data
6. Effectively communicate results

TOPIC COVERED

UNIT-I **9L**

Introduction to Data Science, Different Sectors using Data science, Purpose and Components of Python in Data Science. Applications of Data Science, Data Science and Ethical Issues- Discussions on privacy, security, ethics- A look back at Data Science- Next-generation data scientists.

UNIT-II **9L**

Data Analytics Process, Knowledge Check, Exploratory Data Analysis (EDA), EDA- Quantitative technique, EDA- Graphical Technique, Data Analytics Conclusion and Predictions.

UNIT-III **9L**

Feature Generation and Feature Selection (Extracting Meaning from Data)- Motivating application: user (customer) retention- Feature Generation (brainstorming, role of domain expertise, and place for imagination)- Feature Selection algorithms.

UNIT-IV **9L**

Data Visualization- Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects- Exercise: create your own visualization of a complex dataset.

TEXT BOOKS

1. Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly Media (2015).
2. An introduction to Data Science, Jeffrey Stanton.
3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 2nd edition, William McKinney, O'Reilly Media (2017)
4. Business Analytics: The Science of Data - Driven Decision Making, U Dinesh Kumar, John Wiley & Sons.
5. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, John Wiley & Sons.
6. Data Science from Scratch, Joel Grus, O'Reilly Publisher Media.

REFERENCE BOOKS

1. Mining of Massive Datasets. v2.1, Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. , Cambridge University Press.
2. Python Data Science Handbook, Jake VanderPlas, Shroff Publisher/O'Reilly Publisher Media.
3. Data Analysis with Open Source Tools, Philipp Janert, Shroff Publisher/O'Reilly Publisher Media.

Computing Using Python

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-303	Computing using Python	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Objectives: The course will enable students to:
Learn the syntax and semantics of Python Programming Language.
Write Python functions to facilitate code reuse and manipulate strings.

Illustrate the process of structuring the data using lists, tuples and dictionaries.
Demonstrate the use of built-in functions to navigate the file system.
Appraise the need for working on web scraping

Course Outcomes:

1. Understand and apply basic Python syntax, data types, variables, and flow control structures.
2. Develop functions with parameters, return values, and manage variable scope and exceptions.
3. Perform operations on lists using built-in methods and augmented assignment operators.
4. Use dictionaries and strings to structure and manipulate textual and real-world data.
5. Implement pattern matching using regular expressions for text processing tasks.
6. Build Python programs integrating functions, data structures, and regular expressions.

TOPIC COVERED

UNIT-I **9L**

Introduction, Python Basics: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program. Flow control: Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit().

UNIT-II **9L**

Functions: def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling. Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods.

UNIT-III **9L**

Dictionaries and Structuring Data: The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things. Manipulating Strings - Working with Strings, Useful String Methods.

UNIT-IV **9L**

Pattern Matching with Regular Expressions: Finding Patterns of Text without Regular Expressions, Finding Patterns of Text with Regular Expressions, More Pattern Matching with Regular Expressions, Greedy and Nongreedy Matching, The findall() Method, Character Classes, Making Your Own Character Classes, The Caret and Dollar Sign Characters, The WildcardCharacter, Review of Regex Symbols, Case-Insensitive Matching, Substituting Strings with the sub() Method, Managing Complex Regexes, Combining re.IGNORECASE, re .DOTALL, and re .VERBOSE.

Text Books:

1. Al Sweigart, “Automate the Boring Stuff with Python”, William Pollock, 2015,
ISBN: 978-1593275990.

Reference Books:

1. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd Edition,Green Tea Press, 2015, ISBN: 978-9352134755.
2. Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014.
3. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson EducationIndia, 2015. ISBN-13: 978-9332555365.
4. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, "Data Structures and Algorithms in Python", 1st Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126562176.
5. ReemaThareja, "Python Programming using problem solving approach", Oxford University press, 2017. ISBN-13: 978-0199480173
6. Charles R. Severance, "Python for Everybody: Exploring Data Using Python 3", 1st Edition, Shroff Publishers, 2017. ISBN: 978-9352136278.

List of Experiments:

1. Write a Python program to demonstrate basic data types, expressions, variable assignments, and string operations (concatenation, replication).
2. Develop a Python program using flow control statements (if-else, while, for) and Boolean logic to solve a real-life problem.
3. Create functions with parameters and return values, demonstrate local and global scope, and implement basic exception handling.
4. Write a Python program to perform list operations such as indexing, slicing, appending, inserting, deleting, and using list methods.
5. Implement a program using dictionaries to store and retrieve data, and use string methods for data formatting and manipulation.
6. Develop a Python script that uses regular expressions to find, match, and replace patterns in a given text input.

Data Visualization

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-503	Data Visualization	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Identify and interpret basic plots using Matplotlib, Seaborn, and Plotly.
2. Analyze dataset types and apply suitable visualization techniques.
3. Create dynamic and animated visualizations using modern tools.
4. Evaluate visualization effectiveness using validation techniques and design rules.
5. Design spatial and network visualizations using visual encoding principles.
6. Develop interactive dashboards using multi-view and focus-context methods.

TOPIC COVERED

UNIT-I **9L**

Line plot - Bar plot - Pie Chart - Scatter Plot - Histogram - Stacked Bar Charts - Sub Plots - Matplotlib, Searborn, Plotly - Seaborn Styles, What is Visualization and Why do it: External representation – Interactivity – Difficulty in Validation. Data Abstraction: Dataset types – Attribute types – Semantics. Task Abstraction – Analyze, Produce

UNIT-II **9L**

Box plot - Density Plot - Area Chart - Heat map - Tree map - Graph Networks, Value of Visualization –, Search, Query. Four levels of validation – Validation approaches – Validation examples. Marks and Channels

UNIT-III **9L**

Dynamic charts - Dynamic maps - Animation types - 2D, 3D, Motion Animation - Animation Principles - Altair Package - Statistical Visualizations Rules of thumb – Arrange tables: Categorical regions – Spatial axis orientation – Spatial layout density. Arrange spatial data: Geometry – Scalar fields – Vector fields – Tensor fields. Arrange networks and trees: Connections, Matrix views – Containment. Map color: Color theory, Color maps and other channels.

UNIT-IV **9L**

Visual Perception and Cognition - Gestalt's Principles - Tufte's Principles - Applications of Principles of Information Visualization - Dashboard Design, Manipulate view: Change view over time – Select elements – Changing viewpoint – Reducing attributes. Facet into multiple views: Juxtapose and Coordinate views – Partition into views – Static and Dynamic layers – Reduce items and attributes: Filter – Aggregate. Focus and context: Elide – Superimpose – Distort – Case studies

References:

1. "Fundamentals of Data Visualization" by Claus O. Wilke
2. “Information Visualization: Perception for Design" by Colin Ware

Statistical Learning for Reliability Analysis

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-703	Statistical Learning for Reliability Analysis	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Understand the fundamentals of reliability analysis and apply probability distributions in reliability estimation.
2. Apply sampling distribution techniques for reliability prediction and perform parametric statistical inference.

3. Use hypothesis testing and confidence intervals for reliability analysis using parametric methods.
4. Analyse the underlying relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
5. Plan and execute successful machine learning and big data projects, including selecting an adequate process for the specific task and avoiding the machine learning pit falls.
6. Evaluate the issues raised by current research in the field of machine learning

TOPIC COVERED

UNIT-I **9L**

Introduction to reliability, reliability estimation, concept of statistical learning, advanced techniques to reliability analysis. Probability distribution techniques: discrete and continuous probability distributions and their applications to reliability estimation modeling.

UNIT-II **9L**

Sampling distribution techniques and their different applications for reliability prediction., Statistical inference technique-I (Parametric-based approaches: Hypothesis testing, Confidence interval estimation). Case studies for reliability analysis with parametric-based approaches.

UNIT-III **9L**

What is Machine Learning, Introduction to ML's three approaches: Supervised, Unsupervised and Reinforcement Learning.Linear Regression, Logistic Regression, Decision Trees, Random Forest, Support Vector Machine, K-Nearest Neighbours, CN2 Algorithm, Naive Bayes

UNIT-IV **9L**

K-means, Silhoutte Scores, Hierarchical Clustering, Fuzzy c-means, DBScan, Low Variance Filter, High Correlation Filter, Backward Feature Elimination,Forward Feature Selection, Principal Component Analysis, Projection Methods.

Books and References

1. An Introduction to Statistical Learning: with Applications in R, G James, D. Witten, T Hastie, and R. Tibshirani, (Springer)
2. Software for Data Analysis: Programming with R (Statistics and Computing), John M. Chambers (Springer)
3. Advances in Complex Data Modeling and Computational Methods in Statistics, Anna Maria Paganoni and Piercesare Secchi, (Springer)
4. Machine Learning For Dummies, John Paul Mueller and Luca Massaron
5. A Course in Machine Learning, Hal Daumé III.
6. Programming Collective Intelligence: Building Smart Web 2.0 Applications, Toby Segaran

Reinforcement Learning

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-903	Reinforcement Learning	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcomes:

1. Describe the history, scope, and interdisciplinary connections of Reinforcement Learning.
2: Apply foundational probability concepts relevant to Reinforcement Learning models.
3: Explain key RL concepts including Markov property, MRP, and MDP.
4: Derive and analyze Bellman equations for reward and decision processes.
5: Implement Monte Carlo methods for model-free reinforcement learning tasks.
6: Apply policy gradient and actor-critic methods to optimize RL policies.

TOPIC COVERED

UNIT-I **9L**

Course logistics and overview. Origin and history of Reinforcement Learning research. Its connections with other related fields and with different branches of machine learning.

UNIT-II **9L**

Brush up of Probability concepts - Axioms of probability, concepts of random variables, PMF, PDFs, CDFs, Expectation. Concepts of joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence.

UNIT-III **9L**

Introduction to RL terminology, Markov property, Markov chains, Markov reward process (MRP). Introduction to and proof of Bellman equations for MRPs along with proof of existence of solution to Bellman equations in MRP. Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.

UNIT-IV **9L**

Overiew of Monte Carlo methods for model free RL, First visit and every visit Monte Carlo, Monte Carlo control, On policy and off policy learning, Importance sampling. Getting started with policy gradient methods, Log-derivative trick, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods.

Reference Book

1. "Reinforcement Learning: An Introduction", Richard S. Sutton and Andrew G. Barto, 2nd Edition
2. "Probability, Statistics, and Random Processes for Electrical Engineering", 3rd Edition, Alberto Leon-Garcia
3. "Machine Learning: A Probabilistic Perspective", Kevin P. Murphy Link

List of Experiments:

1. Study and present the history and evolution of reinforcement learning and its connections with supervised and unsupervised learning using a case-based presentation or report.

2. Implement a simulation to demonstrate probability concepts (e.g., generating PMFs, PDFs, CDFs) using Python and visualize the relationship between joint, marginal, and conditional distributions.
3. Create a simulation of a Markov Reward Process (MRP), compute the state value function using Bellman equations, and visualize state transitions and rewards.
4. Implement a Markov Decision Process (MDP) environment and compute state-value and action-value functions using Bellman expectation and optimality equations.
5. Develop Monte Carlo simulation for a simple grid world or custom environment using first-visit and every-visit approaches for estimating value functions.
6. Implement the REINFORCE algorithm for a small environment (like Cart Pole or a custom environment), and demonstrate the effect of baselines and variance reduction techniques using actor-critic methods.

Deep Learning

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-913	Deep Learning	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Objectives: The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Course Outcomes:
1. Understand deep learning
2. Use different types of classifiers.
3. Describe the feed-forward and deep networks.
4. Design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
5. Implement deep neural networks to solve a problem
6. Analyse performance of deep networks.

TOPIC COVERED

UNIT-I **9L**

Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive sub-gradient method.

UNIT-II **9L**

Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyper parameter tuning.Convolution Neural Networks: Introduction to convolution neural networks: stacking, stridingand pooling, applications like image, and text classification.

UNIT-III **9L**

Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks, LSTM networks.

UNIT-IV **9L**

Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders,denoising autoencoders, representational power, layer, size, and depth of autoencoders,stochastic encoders and decoders

Reference Books:

1. Bunduma, N. (2017). Fundamentals of Deep Learning
2. Heaton, J.(2015). Deep Learning and Neural Networks, Heaton Research Inc.
3. Goodfellow, I. (2016). Deep Learning. MIT Press.
4. Deng, L., & Yu, D. (2009). Deep Learning: Methods and Applications (Foundations andTrends in Signal Processing). Publishers Inc.

List of Experiments:

1. Implement logistic regression for binary classification using gradient descent and evaluate performance.

2. Apply stochastic gradient descent and momentum to optimize a supervised classification task.

3. Build and train a feedforward neural network on a dataset like MNIST.

4. Apply dropout and L2 regularization to prevent overfitting in a deep neural network.

5. Design and train a CNN for image classification using convolution, pooling, and striding.

6. Perform text classification using a 1D CNN on a sentiment analysis dataset.

7. Implement a simple RNN for character-level sequence prediction.

8. Build a bidirectional RNN and compare performance with a unidirectional RNN.

9. Construct an encoder-decoder LSTM model for machine translation.

10. Train a denoising autoencoder to reconstruct images with added noise.

Minor Degree 2: Artificial Intelligence & Machine Learning

Category	**Subject Code**	**Name of Subject**	**Credit**			**Total Credit**
			L	**T**	**P**	
PE1	ECS-104	Introduction to Artificial Intelligence	3	1	0	4
PE3	ECS-304	Introduction to Machine Learning	3	0	2	4
PE5	ECS-504	Artificial Intelligence: Knowledge Representation and Reasoning	3	1	0	4
PE7	ECS-704	Natural Language Processing	3	0	2	4
PE9	ECS-904	Optimization and Machine Learning	3	1	0	4/
	ECS-914	Deep Learning and Neural Network	3	0	2	4
Total						20

Introduction to Artificial Intelligence

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-104	Introduction to Artificial Intelligence	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:
To review and strengthen important mathematical concepts required for AI & ML.
Introduce the concept of learning patterns from data and develop a strong theoretical foundation for understanding state of the art Machine Learning algorithms.

COURSE OUTCOMES:

1. Explain AI fundamentals and represent knowledge using predicate logic and rules.
2. Differentiate between procedural and declarative knowledge and apply logic programming basics.
3. Describe machine learning types and classify problems into regression and classification.
4. Implement and optimize linear regression using gradient descent.
5. Apply logistic regression for classification and handle multi-class and overfitting issues.
6. Understand and implement clustering algorithms with their real-world applications.

TOPIC COVERED

UNIT-I 9L

Defining Artificial Intelligence, Defining AI techniques, Using Predicate Logic and Representing Knowledge as Rules, Representing simple facts in logic, Computable functions and predicates, Procedural vs Declarative knowledge, Logic Programming, Mathematical foundations: Matrix Theory and Statistics for Machine Learning.

UNIT-II 9L

Idea of Machines learning from data, Classification of problem –Regression and Classification, Supervised and Unsupervised learning.

UNIT-III **9L**

Linear Regression: Model representation for single variable, Single variable Cost Function, Gradient Decent for Linear Regression, Gradient Decent in practice.

UNIT-IV **9L**

Logistic Regression: Classification, Hypothesis Representation, Decision Boundary, Cost function, Advanced Optimization, Multi-classification (One vs All), Problem of Overfitting. Discussion on clustering algorithms and use-cases cantered around clustering and classification.

Text Books/References:

1. Saroj Kaushik, Artificial Intelligence, Cengage Learning, 1st Edition 2011.
2. Anindita Das Bhattacharjee, "Practical Workbook Artificial Intelligence and Soft Computing for beginners, Shroff Publisher-X team Publisher.
3. Yuxi (Hayden) Liu, "Python Machine Learning by Example", Packet Publishing Limited, 2017.
4. Tom Mitchell, Machine Learning, McGraw Hill, 2017.
5. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2011.
6. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2011.

Introduction to Machine Learning

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-304	Introduction to Machine Learning	3	0	2	4

Course Assessment Methods: Continuous assessment through assignments, quizzes, tutorials, one minor test, one major examination, and lab-based project evaluations.

Course Outcomes: Upon successful completion of this course, students will be able to:

1. Explain basic machine learning concepts and types of learning (supervised, unsupervised, reinforcement).
2. Apply linear algebra concepts like vectors and matrices in machine learning algorithms.
3. Use simple optimization techniques like gradient descent for model training.
4. Implement supervised learning algorithms such as linear regression and logistic regression.
5. Apply unsupervised learning techniques like clustering for data analysis.
6. Evaluate machine learning models using metrics like accuracy, precision, and recall.

TOPIC COVERED

Unit-I: (9 L)

Overview of machine learning: Definition, history, and applications.Types of machine learning: Supervised, unsupervised, and reinforcement learning.Basic concepts: Features, labels, training and test datasets.Linear algebra for machine learning: Vectors, matrices, and their operations.Introduction to Python for machine learning: NumPy, pandas, scikit-learn.

Unit-II (9 L)

Linear regression: Model, cost function, and gradient descent.Logistic regression: Binary classification and probability estimation.Overfitting and underfitting: Bias-variance tradeoff.Evaluation metrics: Accuracy, precision, recall, F1-score.Data preprocessing: Normalization, encoding categorical variables.

Unit-III: (9 L)

Introduction to unsupervised learning: Goals and applications.K-means clustering: Algorithm and implementation.Hierarchical clustering: Concepts and dendrograms.Principal Component Analysis (PCA): Dimensionality reduction.Applications: Customer segmentation, anomaly detection.

Unit-IV (9 L)

Decision trees and random forests: Concepts and implementation.Support Vector Machines: Linear and kernel-based approaches.Model evaluation techniques: Cross-validation, confusion matrix.Regularization: L1 (Lasso) and L2 (Ridge) regularization.Introduction to neural networks: Perceptron and basic architecture.

Reference Books

1. "Mathematics for Machine Learning" by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong — Cambridge University Press, 2020.
2. "Machine Learning Yearning" by Andrew Ng — deeplearning.ai, 2018.
3. "Pattern Recognition and Machine Learning" by Christopher M. Bishop — Springer, 2006.

List of Experiments

1. Perform vector and matrix operations using NumPy for machine learning applications.
2. Load and explore a dataset using pandas; identify features, labels, and handle missing values.
3. Split data into training and testing sets and perform basic visualizations using matplotlib/seaborn.
4. Implement Linear Regression using scikit-learn and visualize the regression line.
5. Implement Logistic Regression for binary classification and evaluate the model.
6. Demonstrate overfitting and underfitting using polynomial regression and analyze the bias-variance tradeoff.
7. Apply data preprocessing techniques: normalization, encoding categorical variables, and feature scaling.
8. Implement K-means clustering and visualize clusters for unsupervised learning.
9. Perform Principal Component Analysis (PCA) for dimensionality reduction and visualize the output.
10. Build and evaluate a Decision Tree classifier using cross-validation and a confusion matrix.

Artificial Intelligence: Knowledge Representation and Reasoning

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-504	Artificial Intelligence: Knowledge Representation and Reasoning	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1: Understand and apply propositional logic, truth tables, and direct proof methods.
2: Explain first-order logic concepts and implement unification, forward, and backward chaining.
3: Utilize the resolution refutation method for automated theorem proving.
4: Demonstrate knowledge of Horn clauses, logic programming, and Prolog basics.
5: Explain rule-based systems, description logics, and default reasoning.
6: Apply advanced knowledge representation techniques like frames

TOPIC COVERED

UNIT-I **9L**

Symbolic Reasoning. Truth, Logicand Provability. Propositional Logic. Direct Proofs. The Tableau Method.

UNIT-II **9L**

First Order Logic. Universal Instantiation. The Unification Algorithm. Forward and Backward Chaining. The Resolution Refutation Method.

UNIT-III **9L**

Horn Clauses and Logic Programming. Prolog. Rule Based Systems. The OPS5 Language. The Rete Algorithm.

UNIT-IV **9L**

Representation in First Order Logic. Conceptual Dependency. Frames. Description Logics and the Web Ontology Language, Taxonomies and Inheritance. Default Reasoning. Circumscription. Auto-epistemic Reasoning. Event Calculus Epistemic Logic. Knowledge and Belief

Textbooks and Reference Books

1. Stuart Russell and Peter Norvig,*Artificial Intelligence: A Modern Approach*, 3rd/4th Edition, Pearson Education.
2. Elaine Rich and Kevin Knight,*Artificial Intelligence*, 3rd Edition, McGraw-Hill.
3. Ivan Bratko,*Prolog Programming for Artificial Intelligence*, 4th Edition, Pearson Education.

Natural Language Processing

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-704	Natural Language Processing	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objectives:
By the end of this course, students will:

Understand fundamental concepts and techniques in Natural Language Processing.
Develop the ability to analyze and model linguistic data computationally.
Gain experience using NLP tools and libraries (e.g., NLTK, spaCy).
Apply NLP methods to solve real-world text processing problems.

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

1 Understand fundamental concepts of NLP, including text preprocessing and language modeling techniques.
2 Apply parsing techniques such as context-free grammar, dependency, and constituency parsing.
3 Implement models for sequential tagging and perform text classification tasks.
4 Explain and use word embeddings and lexical semantics for meaning representation.
5 Develop systems for named entity recognition, machine translation, and question answering.
6 Apply NLP techniques for text summarization, sentiment analysis, and address ethical considerations in NLP.

TOPIC COVERED

Unit-I: Introduction and Core Concepts (9 L)

Overview of NLP and its applications, Text Pre-processing: Tokenization, Stemming, Lemmatization, Stop-word Removal, Regular Expressions: Basic syntax and semantics, Spelling Correction, Language Modeling: Basics and advanced smoothing techniques, Part-of-Speech (POS) Tagging

Unit-II: Syntax and Parsing (9 L)

Context-Free Grammar (CFG), Constituency Parsing and Dependency Parsing, Syntax-Driven Translation, Sequential Tagging Models, Text Classification

Unit-III: Semantics and Word Representations (9 L)

Word Sense Disambiguation, Word Embeddings: Word2Vec, GloVe, fastText, Language Modeling: N-grams, Perplexity, Distributional and Lexical Semantics, Topic Modeling

Unit-IV: Advanced Applications and Ethics (9 L)

Named Entity Recognition (NER), Machine Translation: Seq2Seq Overview, Question Answering, Text Summarization, Sentiment Analysis and Opinion Mining, Information Extraction, Ethical Considerations in NLP

Textbooks

1. Jurafsky, D., & Martin, J. (2009). *Speech and Language Processing* (2nd ed.). Prentice Hall.
2. Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

List of practical

1. Implement tokenization, stemming, lemmatization, and stop-word removal on a text dataset (e.g., news articles) using Python, NLTK, and spaCy. Output cleaned tokens, stems, and lemmas.
2. Use Python's re module to extract patterns like emails, phone numbers, and URLs from a text corpus. Test on a sample dataset.
3. Build a POS tagger using NLTK or spaCy to label words in sentences from a dataset (e.g., Brown Corpus). Compare results with ground truth tags.
4. Train a classifier for sentiment analysis (positive/negative) on a dataset (e.g., IMDB reviews) using scikit-learn or PyTorch/Keras.
5. Generate word embeddings using Word2Vec (Gensim) on a text corpus and visualize similarities with t-SNE or PCA.
6. Implement NER using spaCy or Hugging Face Transformers to identify entities (e.g., person, organization) in a news dataset.
7. Build an extractive or abstractive summarization model using Hugging Face Transformers (e.g., BERT) on a dataset like CNN/Daily Mail.
8. Fine-tune a pretrained transformer (e.g., BERT) for sentiment analysis on a dataset like Twitter sentiments using PyTorch/Keras or TensorFlow.

Optimization and Machine Learning

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-904	Optimization and Machine Learning	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Understand key concepts of linear algebra for machine learning.
2. Demonstrate functions relevant to high-dimensional data spaces.
3. Analyze convex and non-convex optimization problems.
4. Implement gradient descent and its variants.
5. Apply and evaluate advanced optimization techniques.
6. Utilize eigenvalue and singular value decomposition.

TOPIC COVERED

UNIT-I **9L**

Basics of Linear Algebra and Calculus: Subspaces, EigenValue Decomposition, Singular Value Decomposition -Algorithms and Methods, PSD Matrices and Kernel Functions, Vector Calculus

UNIT-II **9L**

Basics of Linear Algebra and Calculus: Subspaces, EigenValue Decomposition, Singular Value Decomposition -Algorithms and Methods, PSD Matrices and Kernel Functions, Vector CalculusConvex Functions, First and Second Order Conditions for Optimizations, Convex and Non Convex Optimization problems in Machine Learning

UNIT-III **9L**

Gradient Descent: math, programming basic optimisation problems and their solutions Variants of Gradient Descent: Projected, Stochastic, Proximal, Accelerated, Coordinate Descent, Training a NeuralNetwork

UNIT-IV **9L**

Variants of Gradient Descent: Projected, Stochastic, Proximal, Accelerated, Coordinate Descent, Training a Neural Network: Theory Newton's Method, Optimization for ML in practice: Pytorch/Tensor Flow.

Reference Books:

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing, 3rd Edition, Pearson, 2022.
2. Christopher D. Manning, Prabhakar Raghavan, and HinrichSchütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
3. Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python, O'Reilly Media, 2009.

Deep Learning and Neural Network

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-914	Deep Learning and Neural Network	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes

1. Understand deep learning basics and models.
2. Use linear algebra and optimization for neural networks.
3. Build models with PyTorch/TensorFlow.
4. Tune and apply models to real-world tasks.
5. To learn the basics of Generative AI
6. Explore AI ethics and emerging trends.

Unit-I: Deep Learning Basics (9 L)

Intro to Deep Learning: Uses, Key Ideas, Linear Algebra: Vectors, Matrices, Neural Networks: Neurons, Loss Functions, Backpropagation and Gradient Descent, Basic introduction to PyTorch/TensorFlow/Keras

Unit-II: Core Deep Learning Models (9 L)

Feedforward Networks: Build and Train, CNNs: Image Processing Basics, RNNs: Sequences with LSTMs, Optimization: SGD, Adam, Preventing Overfitting: Dropout, Batch Norm

Unit-III: Advanced Deep Learning Models (9 L)

Transformers: Attention, BERT Basics, Transfer Learning: Using Pretrained Models, Generative Models: Intro to GANs, VAEs, Model Tuning: Metrics, Hyperparameters

Unit-IV: AI Trends and Ethics (9 L)

Large Language Models: GPT, Prompt Basics, Generative AI: Diffusion Models, Multimodal AI: Vision + Text (e.g., CLIP), AI Ethics: Bias, Fairness, What’s Next: Neuromorphic AI, Federated Learning

References

1. Goodfellow et al. (2016). *Deep Learning*.
2. Deisenroth et al. (2020). *Mathematics for Machine Learning*.
3. Zhang et al. (2023). *Dive into Deep Learning*.

List of Practicals:

1. Implement matrix operations (e.g., dot products) using NumPy and build a simple perceptron with Keras to compute loss functions on a synthetic dataset.
2. Build and train a feedforward neural network using Keras to classify the MNIST dataset. Implement gradient descent and evaluate model accuracy.
3. Create a CNN using Keras for image classification on CIFAR-10. Apply dropout and batch normalization to prevent overfitting and evaluate performance.

4. Implement an LSTM model using Keras for a sequence task (e.g., sentiment analysis on IMDB reviews). Train and visualize loss/accuracy trends.
5. Fine-tune a pretrained BERT model using Keras (via Hugging Face’s transformers with Keras API) for text classification on a dataset like Twitter sentiments.
6. Build a simple GAN using Keras to generate synthetic MNIST digits. Implement generator and discriminator networks and visualize generated images.
7. Use a pretrained GPT-2 model with Keras (via Hugging Face’s Keras API) for text generation. Experiment with prompt engineering on a small text dataset.

Minor Degree 3: Cyber Security & Cyber Physical Systems (CPS)

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-105	Cryptography and Network Security	3	1	0	4
PE3	ECS-305	Practical Cyber Security for Cyber Security Practitioners	3	0	2	4
PE5	ECS-505	Cyber Security and Privacy	3	1	0	4
PE7	ECS-705	Introduction to Security of Cyber-Physical Systems	3	1	0	4
PE9	ECS-905	Ethical Hacking	3	0	2	4
	ECS-915	Computer Forensics	3	0	2	4
Total						**20**

Cryptography and Network Security

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-105	Cryptography and Network Security	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

After successful completion of the course, the learners would be able to:

1. Explain security attacks, services, mechanisms.
2. Describe modern block cipher designs including DES, AES.
3. Understand principles of public key cryptography, RSA, Diffie-Hellman.
4. Apply message authentication, hash functions, and analyze their security against attacks.
5. Explain digital signature algorithms, authentication protocols.
6. Describe IP security architecture, including authentication headers.

TOPIC COVERED

UNIT-I **9L**

Introduction to security attacks - services and mechanism, Introduction to cryptography - Conventional Encryption: Conventional encryption model, Classical encryption techniques - substitution ciphers and transposition ciphers – cryptanalysis, Steganography.

UNIT-II **9L**

Stream and block ciphers - Modern Block Ciphers: Block ciphers principals - Shannon's theory of confusion and diffusion - Feistel structure - data encryption standard (DES) - strength of DES – differential and linear crypt analysis of DES - block cipher modes of operations - triple DES – AES.

UNIT-III **9L**

Principles of public key crypto systems - RSA algorithm, Security of RSA - key management, Diffie-Hellman key exchange algorithm, Introductory idea of Elliptic curve cryptography, Elgamal encryption, Message Authentication and Hash Function, Authentication requirements & functions, message authentication code, birthday attacks, security of hash functions. MD5 message digest algorithm - Secure hash algorithm (SHA).

UNIT-IV **9L**

Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security-pretty good privacy (PGP) - S/MIME, IP Security: Architecture - Authentication header - Encapsulating security payloads – combining security associations - key management.

Text Books:

1. William Stallings, "Crpyptography and Network security Principles and Practices", Pearson/PHI.
2. Wade Trappe, Lawrence C Washington, " Introduction to Cryptography with coding theory", Pearson.

Reference Books

1. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger – Security in computing – Prentice Hall of India.

Practical Cyber Security for Cyber Security Practitioners

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-305	Practical Cyber Security for Cyber Security Practitioners	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcome:

After successful completion of the course, the learners would be able to:

1. Understand cyber security posture, maturity models.
2. Analyze attacker behaviour and continuous improvement strategies.
3. Explain Cyber Kill Chains including Lockheed Martin.
4. Implement defence strategies using the MITRE DEF3ND framework.
5. Secure system configurations using STIG benchmarks.
6. Develop cyber security policies, risk management plans.

TOPIC COVERED

UNIT-I **9L**

Cyber Security Posture, Maturity, and Continuous Improvement Strategies, Knowing your attacker: MITRE ATT&CK Framework

UNIT-II **9L**

Cyber Kill Chains: Lockheed Martin Cyber Kill Chain and Unified Kill Chain, Defending Yourself: MITRE DEF3ND Framework, Securing your Configurations: STIG Benchmarking

UNIT-III **9L**

Sharing Threat Intelligence: STIX and TAXII 2.0, Cyber Security Risk Management, Cyber Resilience Assessment

UNIT-IV **9L**

Cyber Security Policy Formulation, Cyber Crisis Management Plan and Cyber Incident Management Plan.

References:

1. "Cybersecurity and Cyberwar: What Everyone Needs to Know" by P.W. Singer and Allan Friedman — Oxford University Press, 2014.
2. "The Cybersecurity Playbook: How Every Leader and Employee Can Contribute to a Culture of Security" by Allison Cerra — McGraw-Hill Education, 2020.
3. "Cybersecurity Operations Handbook" by John R. Vacca — Academic Press, 2014.

List of Experiments:

1. Explore and map attacker techniques using the MITRE ATT&CK framework.
2. Analyze case studies based on the Lockheed Martin Cyber Kill Chain stages.
3. Perform STIG benchmarking on sample system configurations.
4. Conduct a cybersecurity risk assessment on a simulated environment.
5. Develop a cybersecurity policy document tailored for a hypothetical organization.
6. Prepare a Cyber Crisis Management Plan and simulate incident handling.
7. Create a Cyber Incident Management Plan including communication workflows.
8. Conduct a tabletop exercise simulating a cybersecurity incident response.

Cyber Security and Privacy

Subject Code	**Name of Subject**	**Credit**			**Total Credit**
		L	**T**	**P**	
ECS-505	Cyber Security and Privacy	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

After successful completion of the course, the learners would be able to:

1. Understand the core principles of cybersecurity including.
2. Explain confidentiality, integrity, and availability (CIA).
3. Explain governance, risk management, compliance frameworks
4. Analyze cyber security policies and their role in organizational security.
5. Describe industry defense technologies, attack methods, and cybersecurity standards.
6. Understand foundations of information privacy, measurement theories, and privacy regulations.

TOPIC COVERED

UNIT-I **9L**

Introduction to cyber security, Confidentiality, integrity, and availability. Foundations - Fundamental concepts, CIA, CIA triangle, data breach at target.

UNIT-II **9L**

Security management, Governance, risk, and compliance (GRC)- GRC framework, security standards Control strategies and protection mechanisms, Cryptography for security. Cyber security policy - ESSP, ISSP, SYSSP. Risk Management - Cyber Risk Identification, Assessment, and Control.

UNIT-III **9L**

Cyber security: Industry perspective - Defense Technologies, Attack, Exploits. Cyber security technologies - Access control, Encryption, Standards. Foundations of privacy - Information privacy, Measurement, Theories. Privacy regulation - Privacy, Anonymity, Regulation, Data Breach.

UNIT-IV **9L**

Privacy regulation in Europe, Privacy: The Indian Way - Data Protection, GDPR, DPDP, Aadhar. Information privacy: Economics and strategy, Economic value of privacy, privacy valuation

Reference Books:

1. "Principles of Information Security" by Michael E. Whitman and Herbert J. Mattord – Cengage Learning, 6th Edition, 2017.
2. "Cybersecurity for Beginners" by RaefMeeuwisse – Cyber Simplicity Ltd., 2021.
3. "Information Privacy: Statutes and Regulations" by Richard M. Smith – Aspen Publishing, Latest Edition.

INTRODUCTION TO SECURITY OF CYBER-PHYSICAL SYSTEMS

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	

ECS-705	Introduction to Security of Cyber-Physical Systems	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes:
After successful completion of the course, the learners would be able to:

1. To apply basics of security and issues related to it.
2. To use biometric techniques available and how they are used in today's world.
3. To investigate Security issues in web and how to tackle them.
4. To Learn mechanisms for transport and network security
5. Understand abstraction in system designs
6. Express pre- and post-conditions and invariants for CPS models

TOPIC COVERED

UNIT-I **9L**
Overview of Security and Privacy in Information System, Applied Cryptography & Intrusion Detection, Architecture of Applied Cryptography, One Way Hash Function and Integrity, Encryption Algorithms and Confidentiality, Digital Signature and Authentication (DH, RSA, 2 class), Intrusion Detection and Information Theory.

UNIT-II **9L**
Internet of Things Security, Security and Privacy for IoT Case Study: Smart Home, Smart Grid Network, Modern Vehicle, Wearable Computing & BYOD, Mobile HealthCare.

UNIT-III **9L**
Software-Defined Networks, Introduction of Software-Defined Networks, Security for Software-Defined Networks, Privacy Leakages for Software-Defined Networks, Case Studies: How to Attack Software-Defined Networks.

UNIT-IV **9L**
Cyber-Physical Systems (CPS), CPS - Platform components, CPS implementation issues, Intelligent CPS, Secure Deployment of CPS.

TEXT BOOKS

1. Cyber Security, Nina Godbole, John Wiley & Sons.
2. Li Da Xu, Shancang Li, "Securing the Internet of Things", Syngress.
3. Alasdair Gilchrist, "IoT Security Issues", De Gruyter

REFERENCE BOOKS

1. Sean Smith, "The Internet of Risky Things", Sean Smith, Shroff Publisher/O'Reilly Publisher

Ethical Hacking

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	

ECS-905	Ethical Hacking	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcome:

After successful completion of the course, the learners would be able to:
1. Understand the fundamentals of ethical hacking.
2. Configure computer networking, and vulnerability scanning tools.
3. Know secure hacking environments and perform system-level attacks.
4. Analyze advanced security topics such as steganography, biometric authentication.
5. Perform traffic analysis and security testing using tools like Wireshark.
6. Demonstrate ethical hacking techniques.

TOPIC COVERED

UNIT-I 9L

Introduction to ethical hacking. Fundamentals of computer networking. TCP/IP protocol stack. IP addressing and routing. TCP and UDP. IP subnets. Routing protocols. IPv- 6, Installation of attacker and victim system. Information gathering using advanced google search, archive.org, netcraft, whois, host, dig, dnsenum and NMAP tool. Vulnerability scanning using NMAP and Nessus.

UNIT-II 9L

Creating a secure hacking environment. System Hacking: password cracking, privilege escalation, application execution. Malware and Virus. ARP spoofing and MAC attack.Introduction to cryptography, private-key encryption, public-key encryption.

UNIT-III 9L

Cryptographic hash functions, digital signature and certificate, applications. Steganography, biometric authentication, network-based attacks, DNS and Email security. Packet sniffing using wireshark and burpsuite, password attack using burp suite.

UNIT-IV 9L

Social engineering attacks and Denial of service attacks. Elements of hardware security: side-channel attacks, physical inclinable functions, hardware trojans. Different types of attacks using Metasploit framework: password cracking, privilege escalation, remote code execution, etc. Attack on web servers: password attack, SQL injection, cross site scripting.

Text Books and References

1. Data and Computer Communications -- W. Stallings.
2. Data Communication and Networking -- B. A. Forouzan
3. TCP/IP Protocol Suite -- B. A. Forouzan
4. UNIX Network Programming -- W. R. Stallings
5. Introduction to Computer Networks and Cybersecurity -- C-H. Wu and J. D. Irwin
6. Cryptography and Network Security: Principles and Practice -- W. Stallings

List of Experiments:

1. Information Gathering and Scanning Using tools like whois, dig, dnsenum, and Nmap.
2. Perform a vulnerability scan using on a virtual machine and generate a vulnerability assessment report.
3. Password Cracking and Privilege Escalation and demonstrate privilege escalation on a Linux system.
4. Packet Sniffing and Analysis to analyze network packets using Wireshark.
5. Cryptography and Hashing SHA-256 using Python.
6. Web Server Exploitation attacks such as SQL injection or remote code execution.

Computer Forensics

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-915	Computer Forensics	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcome:

1. Understand the principles of Computer Forensics and how resultant evidence can be applied within legal cases.
2. Apply the concepts in evidence recovering files, data forensics investigation.
3. Analyze the concept of incidence response methodology and fake mail concept in forensic investigation.
4. Evaluate relevant technical and legal information and emerging industry trends.

TOPIC COVERED

UNIT-I **9L**

Introduction to Computer Forensics and Its Applications: Use of Computer Forensics in Law Enforcement, Computer Forensics Assistance in Human Resources and Employment Proceedings, Computer Forensics Services, Benefits of Professional Forensics Methodology, Steps Taken by Computer Forensics Specialists.

UNIT-II **9L**

Computer Forensics Technology and Evidence Handling: Types of Computer Forensics Technology-Business Computer Forensics Technology, Military Computer Forensics Technology, Law Enforcement Computer Forensics Technology.

Computer Forensics Evidence and Capture: Daubert Guidelines, Data Recovery-Definition, Backup and Recovery, Role of Backup in Data Recovery, The Data Recovery Solution.

UNIT-III **9L**

Evidence Collection, Preservation, and Analysis:

Evidence Collection and Data Seizure-Collection Options and Obstacles, Types of Evidence, Rules of Evidence and Chain of Custody, ACPO Principles.

Duplication and Preservation of Digital Evidence-Preserving the Digital Crime Scene, Computer Evidence Processing Steps, Legal Aspects of Collecting and Preserving Computer Forensic Evidence

Computer Forensic Analysis and Validation-Determining What Data to Collect and Analyze, Validating Forensic Data, Addressing Data-Hiding Techniques, Performing Remote Acquisitions.

UNIT-IV **9L**

Network Forensics, Crime Scene Processing, and Forensic Tools: Network Forensics Overview-Performing Live Acquisitions, Developing Standard Procedures for Network Forensics, Using Network Tools, Examining the Honeynet Project.Processing Crime at Incident Scenes-Identifying and Collecting Digital Evidence, Processing Private-Sector and Law Enforcement Crime Scenes, preparing for a Search and Securing the Scene, Seizing and Storing Digital Evidence, obtaining a Digital Hash and Reviewing a Case.

Computer Forensic Tools- Evaluating Tool Requirements, Software and Hardware Forensic Tools, Validating and Testing Forensic Software

Email Forensics: Role of Email in Investigations, Understanding Client and Server Roles in Email, Investigating Email Crimes and Violations, Using Specialized Email Forensic Tools.

Text Book:

1. Computer Forensics, Computer Crime Investigation by John R,Vacca, Firewall Media, New Delhi.

Reference Books:

1. Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning.
2. Real Digital Forensics by Keith J.Jones, Richard Bejitlich, Curtis W.Rose, Addison Wesley Pearson Education.

List of Experiments:

1. Study the basic concepts of computer forensics and familiarize with various forensic tools used in investigations.
2. Implement buffer over flow attack using C Language.
3. File Recovery Using Forensics Tools
4. Hiding and Extracting Text Behind an Image File.
5. Hiding and Extracting Text Behind an Audio File.
6. Analyze image files for metadata using ExifReader software.
7. Create a bit-by-bit forensic image of a hard drive using EnCase Forensics.
8. Restore and analyze a previously created forensic image of a hard disk.
9. Identify and collect relevant email data from a target system.
10. Simulate fake email scenarios using services like fakesend.com, fakemailer.net, etc.
11. Analyze and extract data from web browsers (history, cookies, cache, etc.)

12. Determine the last activities performed on the system using forensic methods.
13. Compare two files to detect tampering or modifications using Compare IT.
14. Perform live forensic investigation using Autopsy forensic browser.
15. Simulate a phishing attack and analyze how attackers deceive users.
16. Demonstrate SQL Injection vulnerability and its exploitation.

Minor Degree 4: Computer Graphics and Virtual Reality

Category	**Subject Code**	**Name of Subject**	**Credit**			**Total Credit**
			L	**T**	**P**	
PE1	ECS-106	Computer Graphics for Virtual Reality	3	1	0	4
PE3	ECS-306	Computer Graphics and Game Programming	3	0	2	4
PE5	ECS-506	Concepts of Virtual and Augmented Reality	3	1	0	4
PE7	ECS-706	Scientific and Engineering Data Visualization	3	1	0	4
PE9	ECS-906	Interactive 3D Graphics	3	0	2	4/
	ECS-916	Sensors and actuators in AR/VR system	3	0	2	4
Total						20

COMPUTER GRAPHICS FOR VIRTUAL REALITY

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-106	Computer Graphics for Virtual Reality	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:

1. To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.
2. To learn the basic principles of 3-dimensional computer graphics.
3. Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. To list the basic concepts used in computer graphics.
2. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.
3. To define the fundamentals of animation, virtual reality and its related technologies.
4. To design an application with the principles of virtual reality.
5. To understand a typical graphics pipeline
6. To design an application with the principles of virtual reality

TOPIC COVERED

UNIT-I **9L**

Graphics system and models: applications of computer graphics, graphics system, physical and synthetic images, imaging systems, graphics architectures, Geometric objects and transformations: scalars, points and vectors, three-dimensional primitives, coordinate systems and frames, frames in OpenGL, matrix and vector classes, modelling a colored cube, affine transformations - translation, rotation and scaling, transformations in homogeneous coordinates, concatenation of transformations, transformation matrices in OpenGL, interfaces to 3D applications, quaternion.

Vertices to fragments: basic implementation strategies, four major tasks, clipping - line clipping, polygon clipping, clipping of other primitives, clipping in three dimensions, polygon rasterization, hidden-surface removal, antialiasing, display considerations.

UNIT-II **9L**

Lighting and shading: light and matter, light sources, the Phong reflection model, computation of vectors, polygonal shading, approximation of a sphere by recursive subdivision, specifying lighting parameters, implementing a lighting model, shading of the sphere model, per-fragment lighting, global illumination. Hierarchical modelling: symbols and instances, hierarchical models, a robot arm, trees and traversal, use of tree data structures, other tree structures, scene graphs, open scene graph.

UNIT-III **9L**

Discrete techniques: buffers - digital images - writing into buffers - mapping methods - texture mapping - texture mapping in OpenGL - texture generation - environment maps - reflection map - bump mapping - compositing techniques - sampling and aliasing. Advanced rendering: going beyond pipeline rendering - ray tracing - building a simple ray tracer - the rendering equation - radiosity - Renderman - parallel rendering -volume rendering - Isosurfaces and marching cubes - mesh simplification - direct volume rendering - image-based rendering.

UNIT-IV **9L**

Fractals: modelling - Sierpinski Gasket - coastline problem - fractal geometry - fractal dimension - recursively defined curves - Koch curves - c curves - dragons - space filling curves - turtle graphics - grammar based models - Graftals - volumetric examples - k-midpoint subdivision - fractal Brownian motion - fractal mountains - iteration in the complex plane - Mandelbrot set.

Virtual reality modelling language: introduction, exploring and building a world, building object, lighting, sound and complex shapes, animation and user interaction, colors, normals and textures, nodes references. Special applications: stereo display programming, multiport display systems, multi-screen display system, fly mode navigation, walk through navigation, virtual track ball navigation.

TEXT BOOKS

1. Rajesh K. Maurya, Computer Graphics with Virtual Reality System, John Wiley & Sons.
2. Edward Angel, "Interactive Computer Graphics: A Top-Down Approach Using OpenGL", Addison-Wesley.
3. Foley James D, Van Dam, Feiner and Hughes, "Computer Graphics: Principles and Practice", Pearson Education.

REFERENCE BOOK

Computer Graphics with Virtual Reality System, 3Ed, Rajesh K. Maurya, Wiley India

Computer Graphics and Game Programming

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-306	Computer Graphics and Game Programming	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcome:

1 Understand the timing mechanisms, input handling, and basic physics gameplay.

2 Implement collision detection, resolution techniques skeletal animation systems.

3 Design and manage game scenes, object hierarchies for immersive environments.

4 Use scripting languages to define object behaviors and implement event-driven game logic.

5 Create responsive UI/UX designs and integrate AI techniques into games.

6 Develop save/load systems and apply optimization techniques.

TOPIC COVERED

UNIT-I **9L**

Game Loop and Timing Central structure of a game (initialization, update, render) Delta time, frame rate control Fixed vs variable timestep logic Keyboard, mouse, touch, controller inputs Input mapping and event handling Input buffering and polling, Collision detection (AABB, Circle, Pixel-based) Collision resolution (impulse, overlap correction) Basic physics (velocity, acceleration, gravity, jump mechanics) Sprite sheets and frame-based animation Skeletal animation basics Blend trees (for 3D character animation in engines)

UNIT-II **9L**

Scene and Object Management Game world representation Object hierarchies and parent-child relationships Level design, tilemaps, and layers Audio Systems Playing sound effects and background music 2D/3D positional audio Audio mixing and resource management Scripting and Behavior Programming Scripting languages (C#, GDScript, Blueprints) Attaching behavior to game objects Event-driven game logic

UNIT-III **9L**

UI/UX Design in Games HUD (Health bars, score) Menu systems and transitions Feedback elements: popups, visual effects, sound cues Artificial Intelligence in Games Finite State Machines (FSM) Pathfinding (A*, Dijkstra) Decision trees, Behavior Trees (introductory level)

UNIT-IV **9L**

Save Systems & Game States Serialization of game data (JSON/XML/binary) Save/load mechanics Checkpoints, progress tracking Game Engine Internals components, Scenes, Prefabs Resource loading and management Optimization techniques (culling, LOD, batching) Multiplayer and Networking (Optional/Advanced) Local multiplayer (split-screen or shared screen) Client-server model basics Synchronization and lag compensation (intro level)

Reference Books:

“Game Programming Patterns” by Robert Nystrom – Genever Benning, 2014.

“Fundamentals of Game Design” by Ernest Adams – New Riders, 4th Edition, 2019.

“Introduction to Game Development” by Steve Rabin – Cengage Learning, 2nd Edition, 2010.

List of Experiments:

1. Develop a basic 2D game loop with update, render, and timing control.
2. Implement player input handling using keyboard/mouse/touch and integrate movement physics.
3. Design a tile-based level with layered scene management and integrate object hierarchy.
4. Show sound effects and background music using positional audio and test audio mixing.
5. Create a game UI with health bar and implement simple AI using a finite state machine.
6. Implement a save/load system using JSON or XML.

Concepts of Virtual and Augmented Reality

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-506	Concepts of Virtual and Augmented Reality	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective: This course is designed to give historical and modern overviews and perspectives on virtual reality. It describes the fundamentals of sensation, perception, technical and engineering aspects of virtual reality systems.

Course Outcomes:
After completion of course, students would be able:
1. To analyse the hardware and software requirements.
2. To use the different intersection techniques.
3. To design 3D interfaces.
4. To understand the basic concept and framework of virtual reality.
5. To understand the principles and multidisciplinary features of virtual reality.

6. To understand the technology for multimodal user interaction and perception in VR, the visual, audial, and haptic interface and behaviour.

TOPIC COVERED

UNIT-I 9L

Virtual reality and virtual environments: the historical development of VR, scientific landmarks computer graphics, real-time computer graphics, virtual environments, requirements for VR, benefits of virtual reality. Hardware technologies for 3D user interfaces: visual displays, auditory displays, haptic displays, choosing output devices for 3D user interfaces.

UNIT-II 9L

3D user interface input hardware: input device characteristics, desktop input devices, tracking devices, 3d mice, special purpose input devices, direct human input, home - brewed input devices, choosing input devices for 3D interfaces. Software technologies: database - world space, world coordinate, world environment, objects - geometry, position / orientation, hierarchy, bounding volume, scripts and other attributes, VR environment - VR database, tessellated data, LODs, Cullers and Occluders, lights and cameras, scripts, interaction - simple, feedback, graphical user interface, control panel, 2D controls, hardware controls, room / stage / area descriptions, world authoring and playback, VR toolkits, available software in the market.

UNIT-III 9L

3D interaction techniques: 3D manipulation tasks, manipulation techniques and input devices, interaction techniques for 3D manipulation, design guidelines – 3D travel tasks, travel techniques, design guidelines - theoretical foundations of wayfinding, user centered wayfinding support, environment centered wayfinding support, evaluating wayfinding aids, design guidelines - system control, classification, graphical menus, voice commands, Gestrual commands, tools, mutimodal system control techniques, design guidelines, case study: mixing system control methods, symbolic input tasks, symbolic input techniques, design guidelines, beyond text and number entry.

UNIT-IV 9L

Designing and developing 3D user interfaces: strategies for designing and developing guidelines and evaluation. Advances in 3D user interfaces: 3D user interfaces for the real world, AR interfaces as 3D data browsers, 3D augmented reality interfaces, augmented surfaces and tangible interfaces, agents in AR, transitional AR-VR interfaces - the future of 3D user interfaces, questions of 3D UI technology, 3d interaction techniques, 3d UI design and development, 3D UI evaluation and other issues.

Textbooks/References:

1. Paul Mealy, Virtual & Augmented Reality for Dummies, John Wiley & Sons.

2. Alan B Craig, William R Sherman and Jeffrey D Will, “Developing Virtual Reality Applications: Foundations of Effective Design”, Morgan Kaufmann.

3. Jan Erik Solem, Programming Computer Vision with Python, Shroff Publisher/O’Reilly Publisher

4. Gerard Jounghyun Kim, “Designing Virtual Systems: The Structured Approach”.

5. Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, “3D User Interfaces, Theory and Practice”, Addison Wesley, USA

Scientific and Engineering Data Visualization

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-706	Scientific and Engineering Data Visualization	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes: After completion of course, students would be able:

1. To design processes to develop visualization methods and visualization systems, and methods for their evaluation.
2. To understand various types of data for visualization
3. To complete preparation and processing of data.
4. To complete visual mapping and the visualization.
5. To analyze large-scale abstract data.
6. To understands various related tools and technologies: tensors, pre-processor, solver, post processor, etc.

TOPIC COVERED

UNIT-I **9L**

Visualization - Scientific and engineering perspective - Impact of Visualization in product design, an overview of computer graphics for visualization –Types of data for visualization, Introduction to tensors. role of pre-processor, solver and post processor in solving engineering problems. Overview of massive data visualization: Simplification methods, Multi-resolution methods, External memory methods, Visual scalability. Scalar visualization techniques:

UNIT-II **9L**

Visualization Goals, Representation of mesh and results data, mapping analysis results to Visualizations, one dimensional, two dimensional and 3D Scalar fields - Element face color coding - contour display -Isosurface techniques - Marching Cubes algorithm - Particle sampling.Visualization of flow data: Visualization mappings of flow data.

UNIT-III **9L**

Vector mapping - elementary icons - particle traces – streaklines, streamlines – streamribbons and streamtubes - global icons - Tensor mappings - elementary icons - global icons. Continuum volume display: Volume rendering Terminology, Surface and Volume rendering techniques, Optimization.

UNIT-IV **9L**

Applications of engineering visualization: Case studies created in the laboratory. FUTURE TRENDS: Trends in Computing Hardware, Animation, Video and multi-media, software trends in Visualization.

Text books and Reference books:

1. TorstenMöller and Bernd Hamann Robert D Russell, "Mathematical Foundations of Scientific Visualization, Computer Graphics and Massive Data Exploration", Springer-Verlag Berlin Heidelberg
2. Helen Wright, "Introduction to Scientific Visualization", Springer.
3. Richard S Gallagher, "Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis", CRC Press, CRC Press LLC.

Interactive 3D Graphics

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-906	Interactive 3D Graphics	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcomes:

1. Understand the fundamentals of interactive graphics and 3D rendering pipeline stages.
2. Apply camera transformations and coordinate systems for 3D scene setup.
3. Model, texture, and light 3D objects using various techniques and file formats.
4. Apply shaders to implement real-time effects and animations.
5. Implement interactivity in 3D scenes including object manipulation and UI overlays.
6. Optimize rendering performance and explore applications in AR/VR and real-time systems.

TOPIC COVERED

UNIT-I **9L**

Introduction to interactive graphics: Real-time vs pre-rendered Coordinate systems: World, View, Clip, Screen 3D rendering pipeline overview (Vertex processing to Fragment output) Camera transformation: LookAt, Projection (Perspective & Orthographic) Scene setup: Viewports, Clipping, Field of View (FoV)

UNIT-II **9L**

Modeling, Texturing, and Lighting Mesh representation: Vertices, edges, faces, buffers Importing 3D models (OBJ, FBX, glTF) Texture mapping: UV coordinates, Mipmaps Material properties: Phong, Blinn-Phong, PBR basics Lighting models: Ambient, Directional, Point, Spot

UNIT-III **9L**

Shading, Animation, and Interactivity Introduction to shaders: GLSL / Shader Graph / HLSLVertex and fragment shader structure Object interaction: Picking, dragging, rotating Basic animations: Keyframes, interpolation, skeleton-based motion User interface overlays in 3D scenes

UNIT-IV **9L**

Advanced Interactivity and Applications Scene graph and hierarchical modeling Optimization: Culling, Level of Detail (LOD), Batching Real-time performance metrics (FPS, GPU profiling) Introduction to AR/VR and WebXR (optional extension) Capstone project planning

Reference Books:

1. "Interactive Computer Graphics: A Top-Down Approach with WebGL" by Edward Angel and Dave Shreiner – Pearson, 7th Edition.
2. "Computer Graphics: Principles and Practice" by John F. Hughes et al. – Addison-Wesley, 3rd Edition.
3. "Real-Time Rendering" by Tomas Akenine-Möller, Eric Haines, and Naty Hoffman – A K Peters/CRC Press, 4th Edition.

List of Experiments:

1. Implement a basic 3D scene with camera setup using LookAt and perspective/orthographic projections.
2. Import and render 3D models with texture mapping using UV coordinates.
3. Apply lighting models like ambient, directional, and point light on 3D objects.
4. Write simple vertex and fragment shaders using GLSL or Unity Shader Graph.
5. Create basic object interaction (drag, rotate, pick) within a 3D scene.
6. Animate a model using keyframes or skeleton-based animation techniques.
7. Optimize scene performance using culling, LOD, and view GPU performance metrics.

Sensors and actuators in AR/VR system

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-916	Sensors and actuators in AR/VR system	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcome:

1. Understand the components and operation of AR/VR systems.
2. Explain human perception factors and depth cues in immersive experiences.
3. Apply motion tracking techniques and sensor fusion for real-time positioning.

4. Demonstrate use of haptic devices in virtual environments.
5. Interface sensors and actuators with AR/VR engines for real-time data-driven experiences.
6. Develop interactive AR/VR prototypes and real-time visualization.

TOPIC COVERED

UNIT-I **9L**

Overview of AR and VR systems: HMDs, controllers, sensors, displays Human perception and immersion: Field of view, latency, resolution, depth cues Hardware components: IMUs, gyroscopes, accelerometers, magnetometers Position tracking: Optical, inside-out vs outside-in, SLAM basics

UNIT-II **9L**

Sensors for Spatial Tracking and Motion Capture 6DoF (Degrees of Freedom) tracking and pose estimation Optical tracking systems: IR cameras, markers, MoCap basics Depth sensors: Time-of-Flight, LiDAR, structured light (e.g., Kinect) Sensor fusion and Kalman filtering for accurate motion estimation

UNIT-III **9L**

Actuators and Haptic Feedback in Immersive Systems Types of actuators: Vibration motors, servo motors, linear actuators Haptic feedback systems: gloves, suits, vests Tactile vs force feedback Applications of haptics in training, gaming, and simulations

UNIT-IV **9L**

Interfacing and Prototyping for AR/VR Applications Sensor data communication: Serial, Bluetooth, I2C, USB Integrating hardware with Unity/Unreal using plugins/middleware Real-time data acquisition and visualization Prototyping: Building an interactive AR/VR experience using sensors and actuators

Textbooks

1. Understanding Virtual Reality – William R. Sherman & Alan B. Craig
2. Haptics: Science, Technology, and Applications – Springer
3. Augmented Reality: Principles and Practice – Dieter Schmalstieg& Tobias Hollerer
4. Manufacturer datasheets (MPU6050, Kinect, Leap Motion.

List of Experiments:

1. Setup and explore an HMD system and evaluate field of view, latency, and resolution.
2. Implement basic tracking using an IMU sensor
3. Perform and visualize orientation data in real time.
4. Integrate a vibration motor or haptic glove with Unity.
5. Fuse sensor data using a Kalman filter for smoother motion tracking.
6. Connect sensors Engine using serial/Bluetooth communication and visualize data.
7. Design a simple AR/VR prototype that integrates user interaction with hardware sensors and actuators.

Minor Degree 5: Blockchain and Cryptocurrency

Category	Subject Code	Name of Subject	Credit			Total Credit
			L	T	P	
PE1	ECS-107	Fundamental of Blockchain Technology and Applications	3	1	0	4
PE3	ECS-307	Private and Consortium Blockchain for Enterprise	3	1	0	4
PE5	ECS-507	Smart Contracts and Decentralized Applications	3	1	0	4
PE7	ECS-707	Introduction to Solidity Language	3	0	2	4
PE9	ECS-907	Blockchain for Social Impact: Democracy & Governance	3	1	0	4/
	ECS-917	Blockchain Security and Next Generation Applications	3	1	0	4
					Total	20

Fundamental of Blockchain Technology and Applications

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-107	Fundamental of Blockchain Technology and Applications	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective: It contains a basic introduction to familiarize students with the basics of advanced programming techniques. Here are the key objectives covered in this course.

1. To understand the concepts of blockchain technology
2. To understand the consensus and hyper ledger fabric in blockchain technology
3. Analyze the architecture and consensus mechanisms of Bitcoin.
4. Evaluate smart contract languages in automating and enforcing digital agreements.
5. Explore privacy-enhancing technologies in blockchain vulnerabilities.

Course Outcome: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. State the basic concepts of block chain
2. Paraphrase the list of consensus and Demonstrate and Interpret working of Hyper ledger Fabric
3. Implement SDK composer tool and explain the Digital identity for government
4. Compare and contrast different blockchain platforms.
5. Identify appropriate blockchain platforms for specific use cases.
6. Testing and debugging blockchain applications

TOPIC COVERED

UNIT-I **9L**

History: Digital Money to Distributed Ledgers -Design Primitives: Protocols, Security, Consensus, Permissions, Privacy: Block chain Architecture and Design-Basic crypto primitives: Hash, Signature, Hash chain to Block chain-Basic consensus mechanisms.

UNIT-II **9L**

Requirements for the consensus protocols-Proof of Work (PoW)-Scalability aspects of Block chain consensus protocols: Permissioned Block chains-Design goals-Consensus protocols for Permissioned Block chains, Zcash and Zk-SNARKS for anonymity preservation, attacks on Blockchains – such as Sybil attacks, selfish mining, 51% attacks.

UNIT-III **9L**

Decomposing the consensus process-Hyper ledger fabric components-Chain code Design and Implementation: Hyper ledger Fabric II:-Beyond Chain code: fabric SDK and Front End-Hyper ledger composer tool, Using smart contracts to enforce legal contracts, comparing Bitcoin scripting vs. Ethereum Smart Contracts

UNIT-IV **9L**

Block chain in Financial Software and Systems (FSS): -Settlements, -KYC, -Capital markets-InsuranceBlock chain in trade/supply chain: Provenance of goods, visibility, trade/supply chain finance, invoice management/discounting. Blockchain for Government: Digital identity, land records and other kinds of record keeping between government entities, public distribution system / social welfare systems: Block chain Cryptography: Privacy and Security on Block chain.

Textbooks:

1. Mark Gates, “Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of money”, Wise Fox Publishing and Mark Gates 2017.
2. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O'Dowd, Venkatraman Ramakrishna, “Hands-On Block chain with Hyper ledger: Building decentralized applications with Hyperledger Fabric and Composer”, 2018.
3. Bahga, Vijay Madisetti, “Block chain Applications: A Hands-On Approach”, ArshdeepBahga, Vijay Madisetti publishers 2017.

Private and Consortium Blockchain for Enterprise

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-307	Private and Consortium Blockchain for Enterprise	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Understand the architecture of Ethereum Virtual Machine.
2. Apply state replication and consensus concepts.
3. Perform unit testing, debugging, and gas estimation.
4. Analyze security vulnerabilities in smart contracts.
5. Evaluate smart contract quality assurance practices.
6. Explore Blockchain-as-a-Service (BaaS) platforms beyond Ethereum.

TOPIC COVERED

UNIT-I 9L

History of Virtual Machines- State replication, consensus and the Ethereum Architecture - Introduction to the Ethereum Virtual Machine and EVM Byte Code interpretation Incentivisation structures, rewards schemes, and gas pricing

UNIT-II 9L

The Ethereum Name Services (ENS) Unit Testing and Debugging Contracts Estimating Gas Costs Basics of using Truffle for testing Troubleshooting and Debugging Common design patterns Smart Contract security – overview of attacks on Ethereum smart contracts Deployment Considerations and Other Smart Contract Platforms

UNIT-III 9L

Smart Contacts Quality Assurance Beyond Ethereum Blockchain-as-a-Service (BaaS) and the Dark Market- Secure smart contracts with OpenZeppelin Experimenting

UNIT-IV 9L

Introduction to formal methods for verifying correctness and security.Cross-chain Communication: Concepts of interoperability and tools for cross-chain smart contracts (e.g., Polkadot, Cosmos, LayerZero).

Reference Books:

1. "Mastering Ethereum: Building Smart Contracts and DApps" By Andreas M. Antonopoulos and Gavin Wood
2. "Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners"By Chris Dannen

Smart Contracts and Decentralized Applications

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	

ECS-507	Smart Contracts and Decentralized Applications	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Understand the ecosystem of smart contracts within Ethereum networks.
2. Develop smart contracts using Solidity.
3. Compare the technical foundations of Ethereum and Bitcoin.
4. Analyze Ethereum's and its roles in contract execution.
5. Implement and deploy smart contracts.
6. Identify common smart contract vulnerabilities and apply best practices for secure development.

TOPIC COVERED

UNIT-I **9L**

An overview to the history of smart contracts o An intro to the life-cycle of a smart contract Ethereum's smart contract languages- Interfacing with Ethereum Networks (overview of Ethereum Networks, Clients, Wallets, Transactions etc.)

UNIT-II **9L**

The Solidity Programming Language o Development Environments Blockchain technology Supporting Turing-Complete Languages- A comparison of Ethereum and Bitcoin- Overview of Ethereum's tech stack, architecture.

UNIT-III **9L**

The Ethereum reward scheme, Mist, EVM, Swarm, Whisper, Eth, Gas- A simple Solidity ontract (Contract Walk-through)- The Solidity compiler o Ethereum Contract ABI-Deployment with the Web3.js or Web3J library Virtual Machines and Beyond

UNIT-IV **9L**

Security in Smart Contracts: Common vulnerabilities and best practices. Testing and Debugging: Tools like Ganache, Remix IDE for unit testing and debugging smart contracts.

Reference Book:

1. "Mastering Ethereum: Building Smart Contracts and DApps" By Andreas M. Antonopoulos and Gavin Wood
2. "Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners"By Chris Dannen

Introduction to Solidity Language

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-707	Introduction to Solidity Language	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Outcome:

1. Understand compiler versions, pragmas, and fundamental in Solidity.
2. Develop user-defined functions with control structures.
3. Use exception handling, and event logging.
4. Apply object-oriented principles in Solidity.
5. Implement abstract contracts, interfaces, and function overloading.
6. Integrate smart contracts with front-end libraries.

TOPIC COVERED

UNIT-I **9L**

Understanding the different compiler versions and pragmas Authoring smart contracts o Contracefinitions- Basic data types- Local and State VariablesGlobal Variables and Functions Predefined Global Variables Structs and Enums o Mapping and Arrays Build-in Functions (e.g., addmod, keccak256)

UNIT-II **9L**

User Functions Expressions and Control Structures Valid expressions of the language Exception Handling (e.g., assert, require, revert, throw) Events and Logging Conditional logic Implementation of loops

UNIT-III **9L**

Object Oriented Constructs Contract constructor and self-destruct Function Modifiers and Fallback functions Calling other contracts o Inheritance and Multiple Inheritance Declaring Abstract Classes and Interfaces

UNIT-IV **9L**

Implementation of Abstract interfaces Function Overloading Experimenting with Front-end Libraries Intro to front-end web interfaces Decentralized Data Storage

Reference Books:

"Solidity Programming Essentials: A Beginner’s Guide to Build Smart Contracts for Ethereum and Blockchain" By: Ritesh Modi

List of Experiments:

1: Write and deploy a simple smart contract with basic data types.

2: Create a contract that uses structs and enums to store and manage user data.

3: Implement functions with control structures (if-else, loops).

4: Implement handle exceptions using require, assert, and revert.

5: Develop a smart contract with events and observe event logs in Remix or Ganache.

6: Build a contract demonstrating object-oriented concepts.

7: Create and deploy an abstract contract and an interface.

8: Overload functions in a smart contract and test calling different versions based on parameters.

Blockchain for Social Impact: Democracy & Governance

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-907	Blockchain for Social Impact: Democracy & Governance	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Understand the fundamentals of blockchain technology.
2. Analyze the role of smart contracts.
3. Decentralized applications in enabling trustless digital systems.
4. Evaluate blockchain in democratic processes such as digital identity verification.
5. Compare governance structures and decision-making processes.
6. Assess legal, ethical, and cybersecurity challenges associated with blockchain.

TOPIC COVERED

UNIT-I **9L**

Fundamentals of blockchain and distributed ledger technologies. Consensus mechanisms: Proof-of-Work, Proof-of-Stake, and others. Smart contracts and decentralized applications .

UNIT-II **9L**

Blockchain in Democratic Processes Applications in e-voting systems and digital identity verification Enhancing transparency and accountability in governance. Case studies on blockchain's role in electoral processes. Cornell - Online Education Programs

UNIT-III **9L**

Governance Models in Blockchain Decentralized Autonomous Organizations (DAOs) and their governance structures. Quadratic voting and other innovative decision-making mechanisms. Comparative analysis of governance in public vs. private blockchains.

UNIT-IV **9L**

Regulatory and Ethical Considerations Legal frameworks governing blockchain applications. Ethical implications of decentralization and data sovereignty. Privacy concerns and cybersecurity challenges.

Reference Books:

"Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications" by Imran Bashir

"Blockchain: Blueprint for a New Economy" by Melanie Swan

Blockchain Security and Next Generation Applications

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-917	Blockchain Security and Next Generation Applications	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:

1. Students should be able to understand the security and performance-related issues of blockchain.
2. Students should be able to learn techniques and tools to tackle the security related issues of blockchain.

Course Outcomes: After completion of course, students would be able to:

1. To understand the security and performance perspective of blockchain technology.
2. To learn and apply security analysis and performance-enhancing techniques related to blockchain.
3. To understand difference between different types of coins and tokens related to blockchain technology.
4. To understand the benefits of blockchain in banking sector.
5. Students should be able to understand the benefits of using blockchain in financial sector.
6. Students should understand how decentralized nature of blockchain is impacting banking and financial sector.

Unit 1 **9L**

Security Issues: Blockchain Related Issues, Higher-Level Language (Solidity) Related Issues, EVM Bytecode Related Issues, Real-Life Attacks on Blockchain Applications/ Smart Contracts, Trusted Execution Environments.

Unit 2 **9L**

Security Tools for Smart Contracts: Working, Advantages, And Disadvantages of ToolsOyente, Securify, Maian, Manticore, Mythril, SmartCheck, Verx. Secure Key Management, Quantum Resilience Keys.

Unit 3 9L

Performance Related Issues: Transaction Speed, Transaction Fees, Network Size, Complexity, Interoperability Problems, Lack of Standardization. Lack of Supportive Regulations Related to Blockchain Applications. Decentralized Finance (DeFi): Concept, Benefits and Risks Associated with DeFi, Centralized vs Decentralized finance, DeFi Projects, DeFi future trends.

Unit 4 9L

Cryptocurrencies: Concept, Cryptocurrency Mining, Uses of Cryptocurrencies, Tokens, Token vs Crypto Coin, Concept of ICOs (Initial Coin Offerings), Benefits of Using ICOs, STOs (Security token offerings), ICO vs STO, Cryptocurrency wallets. Decentralized Markets: Concept of Decentralized markets, impact of decentralization on financial market, Decentralized Exchanges (DEX), Security, control and privacy concerns related to DEX, Liquidity and Usability of DEX, best DEXs for trading,

TextBooks/References:

1. Mastering Ethereum: Building Smart Contracts and Dapps Book by Andreas Antonopoulos and Gavin Wood, Shroff Publisher/O′Reilly Publisher.
2. Melanie Swan, Blockchain: Blueprint for a new economy, Shroff Publisher/O’Reilly Publisher.
3. Ron Quaranta, Blockchain in Financial Markets and Beyond: Challenges and Applications, Risk Books Publisher.
4. Richard Hayen, Blockchain & FinTech: A Comprehensive Blueprint to Understanding Blockchain & Financial Technology. - Bitcoin, FinTech, Smart Contracts, Cryptocurrency,

Minor Degree 6: Software Engineering

Category	**Subject Code**	**Name of Subject**	**Credit**			**Total Credit**
			L	**T**	**P**	
PE1	ECS-108	Introduction to C Programming	3	0	2	4
PE3	ECS-308	Data Structures and Algorithms using C	3	0	2	4
PE5	ECS-508	Introduction to Software Engineering	3	1	0	4
PE7	ECS-708	Software Product Management	3	1	0	4
PE9	ECS-908	Software Testing and Quality Assurance	3	1	0	4/
	ECS-918	Agile Software Processes	3	1	0	4
Total						20

Introduction to C Programming

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-108	Introduction to C Programming	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Objective: The course covers the basics of programming and demonstrates fundamental programming techniques, customs and terms including the most common library functions and the usage of the pre-processor.

1. To develop C Programs using basic programming constructs
2. To develop C programs using arrays and strings
3. To develop applications in C using functions and structures

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Basic terminology used in computer programming.
2. Programs development in C Language by writing, compiling and debugging.
3. Design of programs involving simple statements, conditional statements, iterative statements, array, strings, functions, recursion, structure and union.
4. Difference between call by value and call by reference and dynamic memory allocations and use of pointers.
5. Basic operations on a file.
6. Basics of dynamic memory.

TOPIC COVERED

UNIT-I **9L**

Fundamentals of C Programming: Structure of C Program, Writing and Executing the First C Program, Components of C Language, Standard I/O, Formatted I/O. Conditional Program Execution: Applying if and switch Statements, Nesting if and else. Program Loops and

Iterations: Use of while, do while and for Loops, Multiple Loop Variables, Use of break and continue Statements, goto Statement.

UNIT-II **9L**

Arrays: One Dimensional, Multidimensional Array and Their Applications, Declaration and Manipulation of Arrays. Strings: String Variable, String Handling Functions, Array of Strings. Functions: Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions. Storage Classes revisited.

UNIT-III **9L**

Pointers: Pointer Variable and its Importance, Pointer Arithmetic Pointers and Arrays, Pointer and Character Strings, Pointers and Functions, Array of Pointers, Pointers to Pointers. Structure: Declaration and Initialization of Structures, Structure as Function Parameters, Structure Pointers. Union: Declaration and Initialization of Unions, Union as Function Parameters, Union Pointers.

UNIT-IV **9L**

Dynamic Memory Allocation: malloc, calloc, realloc, free functions. File Management: Defining and Opening a File, Closing a File, Input/ Output Operations in Files. The Pre-processor Directives, Macros. Command Line Arguments. Introduction to Graphics Programming.

Textbooks

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 7th Edition,
   Pearson.
2. Schildt, Herbert, Complete Reference with C, Tata McGraw Hill.
3. Kerninghan and Ritchie, The C programming Language, 2nd Edition, Prentice Hall.
4. Richard Bird, Introduction to Functional Programming using Haskell, 2nd Edition, Prentice- Hall International, 1998.

Reference Books

1. Greg Michaelson, An Introduction to Functional Programming Through Lambda Calculus, Dover Edition, Addition Wesley Publication.
2. Samuel P. Harbison, and Guy L. Steele Jr., C-A Reference Manual, Fifth Edition, Prentice Hall, 2002.

List of Experiments

Write a program to print “Hello, World!” and accept user input using formatted I/O.

1. Implement a program to find the largest of three numbers using nested if-else statements.
2. Create a program that prints all prime numbers between 1 and N using for, while, and do-while loops.
3. Write a program to reverse a string using arrays and string handling functions.
4. Develop a program to implement a recursive function for calculating factorial of a number.
5. Write a program to add two matrices using multidimensional arrays.

6. Create a program to swap two numbers using pointers and demonstrate pointer arithmetic.
7. Design a program to store student information using structures and print the details.
8. Write a program to dynamically allocate memory for an array and perform insertion and deletion of elements.
9. Implement file operations: write data to a file and read it back to display on the screen.
10. Use command line arguments to accept input parameters and print them.
11. Create simple macros using preprocessor directives to find the maximum and minimum of two numbers.:

Data Structures and Algorithms using C

Subject Code	Name of Subject	Credit				Total Credit
		L	T	P		
ECS-308	Data Structures and Algorithms using C	3	0	2		4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Objective:
This course helps the students in gaining the knowledge of basic principles of Data Structures. The principal objectives of this course are:
1. To provide the knowledge of basic data structures and their implementations.
2. To understand importance of data structures in context of writing efficient programs.
3. To develop skills to apply appropriate data structures in problem solving.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.
1. Learn the basic types for data structure, implementation, and application.
2. Know the strength and weakness of different data structures.
3. Use the appropriate data structure in context of solution of given problem.
4. Develop programming skills which require to solve given problem.
5. Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data
6. Solve problem involving graphs, trees and heaps.

TOPIC COVERED

UNIT-I **9L**
Introduction: Basic Terminology, Elementary Data Organization, Structure Operations, Complexity and Time-Space Trade-off. Arrays: Definition, Representation and Analysis, Single and Multi-Dimension Array, Address Calculation, Application of Arrays, Character, String in C, Character String Operation, Arrays Parameters, Ordered List, Sparse Matrices and Vectors.Stacks: Array Representation and Implementation of Stack, Operations on Stacks: Push &Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of Stack, Conversion of Infix to Prefix and Postfix Expressions,

UNIT-II **9L**

Queues: Array and Linked Representation and Implementation of Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular Queues, D-Queues and Priority Queues. Linked List: Representation and Implementation of Singly Linked Lists, Two-Way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and Deletion to / from Linked Lists, Insertion and Deletion Algorithms, Doubly Linked List, Linked List in Array, Polynomial Representation and Addition, Generalized Linked List, Garbage Collection and Compaction.

UNIT-III **9L**

Trees: Basic Terminology, Binary Trees, Binary Tree Representation, Algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary Trees, Traversing Binary Trees, Threaded Binary Trees, Traversing Threaded Binary Trees, Huffman Algorithm. Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-Trees.

UNIT-IV **9L**

Searching and Hashing: Sequential Search, Binary Search, Comparison and Analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical Consideration for Internal Sorting. Graphs: Terminology &Representations, Graphs &Multi-Graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

TEXTBOOKS

1. Horowitz and Sahani, Fundamentals of Data Structures, Galgotia Publication, New Delhi.

2. R. Kruse et al, Data Structure and Pragram Design in C, Pearson Education Asia Delhi

3. A. M. Tenenbaum, Data Structures using C & C++, PHI, India

4. K Loudon, Mastering Algorithms with C, Shroff Publication and Distributor Pvt. Ltd.

5. Bruno R Preiss, Data Structure and Algorithms with Object Oriented Design Pattern in C++, John Wiley & Sons

6. Adam Drozdek, “Data Structures and Algorithms in C++”, Thomson Asia Pvt. Ltd. Singapore

REFERENCE BOOKS

1. Lewis, H.R., Denenberg, L., Data Structures and their Algorithms. Published by AddisonWesley, UK, 1991

2. Oluwadare, S.A., Agbonifo, O.C., Fundamentals of Data structures and Algorithms. Lecture Notes, 2013

List of Experiments:

1. Implement a program to create, traverse, and perform basic operations on single and multi-dimensional arrays.
2. Write a program to implement stack operations (push, pop) using arrays and test infix to postfix expression conversion.

3. Implement stack using linked list and perform stack operations.
4. Write a program to implement queue operations using arrays (including circular queue).
5. Implement a singly linked list with insert, delete, and search functionalities.
6. Implement a doubly linked list and perform insertion and deletion at various positions.
7. Create a program to represent and traverse binary trees (inorder, preorder, postorder) using linked representation.
8. Implement Binary Search Tree (BST) with insertion, deletion, and searching operations.
9. Write a program to implement AVL tree insertion and rotations to maintain balance.
10. Implement sequential search and binary search on arrays.
11. Write programs to implement different sorting algorithms: bubble sort, quick sort, and heap sort.
12. Implement hash table with collision resolution (e.g., chaining or linear probing).
13. Write a program to represent a graph using adjacency matrix and perform Depth-First Search (DFS) and Breadth-First Search (BFS).

Introduction to Software Engineering

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-508	Introduction to Software Engineering	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcome (CO):

1. Explain various software characteristics and analyse different software Development Models

2. Demonstrate the contents of a SRS and apply basic software quality assurance practices

to ensure that design, development meet or exceed applicable standards

3. Compare and contrast various methods for software design

4. Formulate testing strategy for software systems, employ techniques such as unit testing,

Test driven development and functional testing

5. Manage software development process independently as well as in teams and make use

of Various software management tools for development, maintenance, and analysis

TOPIC COVERED

UNIT-I 9L

Software Requirement Specification: Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, requirements validationDesign Principles: Problem partitioning, abstraction. Top down and bottom up – design, structured approach. Functional versus 6object oriented approach of design, design specification, Cohesiveness and Coupling. Overview of SA/SD Methodology, structured analysis, data flow diagrams, extending DFD to structure chart.

UNIT-II **9L**

Software project Management: Project planning and Project scheduling. Software Metrics: Size Metrics like LOC, Token Count, Function Count. Cost estimation using models like COCOMO. Risk management activities. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for software industry, SEI capability maturity model.

UNIT-III **9L**

Testing: Verification and validation, code inspection, test plan, test case specification. Level of testing: Unit, Integration Testing, Top down and bottom-up integration testing, Alpha and Beta testing, System testing and debugging. functional testing, structural testing, Software testing strategies.

UNIT-IV **9L**

Software Maintenance: Structured Vs 5 unstructured maintenance, Maintenance Models,Configuration Management, Reverse Engineering, Software Re-engineering

Reference Books

1 Pressman, “Software Engineering – A practitioner’s approach”, 3rd ed.,McGraw Hill Int. Ed..

2 K. K. Aggarwal & Yogesh Singh, “Software Engineering”, 2ndEd., New Age International.

3 SomR. S.merville, “Software Engineering”, Addison Wesley.

Software Product Management (SPM)

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-708	Software Product Management	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Explain the role of Software Product Management within business contexts.
2. Develop effective product strategies aligned with market.
3. Analyze customer segments and value propositions to achieve product-market fit.
4. Apply financial management principles to software products.
5. Understand legal frameworks, data privacy, and plan product releases.
6. Evaluate product life cycle phases and integrate development methodologies.

TOPIC COVERED

UNIT-I **9L**

Introduction and Foundations, and Software Product Management (SPM) Essentials: Introduction to the course, Software as a Business, SPM Role in organization structure and objectives Software Product Management Framework: Overview, Product Management Definitions, Product Family, Product Line, Product Platform, Structure of the SPM Framework and Startup SPM Framework

UNIT-II **9L**

Product Strategy, and Product Strategy Essentials: Role and Elements of Product Strategy and Their Interdependencies, Positioning and Product Definition, Delivery Model, Service Strategy and Sourcing Business Models: Business Models and Business Case, Business Model Canvas, Business Archetypes Customer Segments and Value Propositions : Product Market Fit, Whole Product, Market Development Model

UNIT-III **9L**

Financial Management Software Products: Costing and Pricing Performance and Risk Management AND Ecosystem Management: Performance and Risk Management, Role and Elements of Software Business Ecosystem Legal Aspects : Awareness of relevant legal frameworks, data privacy norms, Product Release Planning and Product Roadmapping: Release Planning Process and its conflicts, Structure of Release Plan Product Roadmap and its elements, Sources of input, Usage of Roadmaps

UNIT-IV **9L**

Product Life Cycle Management: Phases of the Life Cycle, Performance Management Impact From Development Methodologies: Agile, Iterative, and Waterfall Development Methodologies Strategic Management: Strategic Management Essentials, Corporate Strategy and Portfolio Management, Innovation and Resource Management, Market and Product Analysis

Text Books

1. "Software Product Management" by HB Kittlaus and SA Fricker, ISBN: 978-3-642-55140-6

Software Testing and Quality Assurance

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-908	Software Testing and Quality Assurance	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Explain fundamental concepts of software testing.
2. Apply various software testing strategies and techniques.
3. Conduct special testing types such as configuration, compatibility.

4. Develop and document comprehensive test plans, test cases, and manage the bug life cycle.
5. Utilize software testing tools and automation techniques.
6. Understand software quality assurance principles like CMM and ISO 9000.

TOPIC COVERED

UNIT-I **9L**

Introduction to Software Testing: Introduction, Definition of a Bug, Role of a Software Tester, Software Development Model, Software Testing Axioms, Software Testing Terms and Definitions.

UNIT-II **9L**

Fundamentals of Software Testing: Testing Strategies and Techniques, Structural and Functional testing, Static Black Box and Dynamic Black Box Testing Techniques. White Box Testing: Static White Box Testing, Dynamic White Box Testing. Special Types of Testing: Configuration Testing, Compatibility Testing, Graphical User Interface Testing.

UNIT-III **9L**

Documentation and Security Testing: Documentation Testing, Security Testing. Web site Testing: Web Page Fundamentals, Black Box Testing, White Box Testing and Gray Box Testing, Configuration and Compatibility Testing. Testing Tools: Benefits of Automation Testing, Random Testing, Bug Bashes and Beta Testing. Test Planning: Test Planning, Test Cases, Bug life cycle. Software Quality Assurance:

UNIT-IV **9L**

Definition of Quality, Testing and Quality Assurance at Workplace, Test Management and Organizational Structure, Software Quality Assurance Metrics, Quality Management in IT. Organizational Structure: CMM (Capability Maturity Model), ISO 9000, Software Engineering Standards.

Reference Books:

1. Software Testing: Principles and Practices"By: Srinivasan Desikan, Gopalaswamy Ramesh
2. "Foundations of Software Testing: ISTQB Certification"By: Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black

Agile Software Processes

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-918	Agile Software Processes	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

1. Interpret the concept of agile software engineering and its advantages in software development.
2. Analyze the core practices behind several specific agile methodologies.
3. Identify the roles and responsibilities in agile projects.
4. Apply agile concepts in real life projects
5. Access implications of functional testing, unit testing, and continuous integration.
6. Determine the role of design principles in agile software design

TOPIC COVERED

UNIT-I **9L**

Need of Agile software development, agile context– Manifesto, Principles, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits ofsoftware agility. Project Planning: Recognizing the structure of an agile team– Programmers, Managers, Customers. User stories– Definition, Characteristics and content.

UNIT-II **9L**

Estimation– Planning poker, Prioritizing, and selecting user stories with the customer, projecting team velocity for releases and iterations. Project Design: Fundamentals, Design principles–Single responsibility, Open-closed, Liskov substitution, Dependency-inversion, Interface-segregation Design Methodologies: Need of scrum, Scrum practices –Working of scrum, Project velocity, Burn down chart, Sprint backlog, Sprint planning and retrospective,

UNIT-III **9L**

Daily scrum, Scrum roles– Product Owner, Scrum Master, Scrum Team. Extreme Programming- Core principles, values and practices. Kanban, Feature-driven development, Lean software development. The Agile lifecycle and its impact on testing,

UNIT-IV **9L**

Test driven development– Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation.

Text Books

1. Ken Schawber, Mike Beedle, "Agile Software Development with Scrum", International Edition, Pearson.

2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First InternationalEdition, Prentice Hall.

3. Pedro M. Santos, Marco Consolaro, and Alessandro Di Gioia, "Agile Technical Practices Distilled: Alearning journey in technical practices and principles of software design", First edition, Packt Publisher.

Reference Books

3. Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams", Internationaledition, Addison Wesley.
4. Alistair Cockburn, "Agile Software Development: The Cooperative Game", 2nd Edition

Minor Degree 7: Spatial Information Technology

Category	Subject Code	Name of Subject	L	T	P	Total Credit
PE1	ECS-109	Global Positioning System	3	1	0	4
PE3	ECS-309	Principle of Remote Sensing	3	1	0	4
PE5	ECS-509	Spatial Information Technology	3	0	2	4
PE7	ECS-709	Web Technology for Spatial Information Mapping	3	0	2	4
PE9	ECS-909	Satellite Image Processing	3	1	0	4
	ECS-919	Spatial Data Quality Assessment	3	1	0	4
					Total	20

Global Positioning System

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-109	Global Positioning System	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes:

1. Explain the fundamental concepts of geodesy, Earth's shape, gravity field, and atmospheric influence on geodetic measurements.
2. Describe the architecture of GPS, including satellite constellation, signal structure, coordinate systems (WGS-84), and timing systems.
3. Identify and analyze GPS signal errors and biases affecting positioning accuracy.
4. Apply GPS orbital and navigational principles to determine precise positions using field observations and GPS receivers.
5. Process GPS data and integrate it with GIS platforms for spatial analysis and mapping applications.

Topics Covered

Unit I 9L

Introduction to geodesy and its development , Earth and its size and shape, earth and its motions- annual, spin, precession, nutation, polar motion, Earth and its gravity field – anomaly, gravity potential, geoid and deflection to vertical,. Earth and its atmosphere – physical properties, wave propagation through atmosphere, temporal variations, gravitational field of the atmosphere,

Unit II 9L

Introduction of Global Positioning System, Satellite constellation, GPS signals and data, pseudo range and carrier phase measurements; Signal structure; GPS coordinate systems: WGS-84, GPS time; GPS Errors and biases.

Unit III **9L**

GPS orbital Geometry and Navigational solution; GPS receiver, GPS antenna. Radio and its types, Radio Antenna, Surveying with GPS; Planning and field observations;

Unit IV **9L**

Data post-processing; GIS and GPS integration; Other satellite based navigational systems: GLONASS, GALILEO, modernization plans of navigational satellites. GPS Applications

Text &Reference Books:

1. Understanding GPS/GNSS: Principles and Applications (Authors: Elliott D. Kaplan and Christopher Hegarty), 3rd Edition, 2011, Publisher: Artech House
2. Gps Satellite Surveying, by Alfred Leick, Repoport, Tatarikov, 4th Edition, Wiley Publishing House.
3. Global Positioning System: Theory and Practice (Authors: B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins), 5th Edition, 2001, Publisher: Springer.
4. Global Positioning System: Theory and Applications, by Branford W. Parkinson.

Principle of Remote Sensing

Subject Code	**Name of Subject**	**Credit**			**Total Credit**
		L	**T**	**P**	
ECS-309	Principle of Remote Sensing	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes:

1. Explain the principles of remote sensing, electromagnetic spectrum, and interaction of EM energy with atmospheric and surface features.
2. Identify and differentiate between various remote sensing platforms and sensors, along with their specifications and applications.
3. Understand and perform basic digital image processing techniques including enhancement, correction, and transformation of satellite imagery.
4. Apply thematic classification techniques and assess the accuracy of classified images using standard algorithms.
5. Demonstrate the ability to utilize remote sensing data for practical applications in environmental and resource studies.

Topics Covered

UNIT-I 9L

Remote sensing system and its components, Electromagnetic spectrum, definition of emissivity, reflectance, absorbance and transmittance. Spectral signature, atmospheric window, active and passive remote sensing systems, Interaction of electromagnetic energy with atmosphere and earth features, factors affecting the reflectance, Thermal and Microwave Remote Sensing.

UNIT-II 9L

Airborne and space platforms, Advantages and disadvantages of each, principle and functioning of multi-spectral, thermal & line scanners, Multi concept of remote sensing, Different satellite and sensor combinations: LANDSAT, SPOT, IRS series of satellites and sensors. Their important characteristics: such as flight altitude, IFOV, spatial resolution, swath, spectral bands, and receptivity.

UNIT-III 9L

Introduction to Digital Image Processing, digital image representation, and characterization, Concept of color, Color composites, histograms and scatter plot, image enhancement, contrast stretching, radiometric processing including correction of atmospheric corrections; geometric corrections, Image Transformations such as subtraction, ratioing, NDVI and PCA.

UNIT-IV 9L

Ground truth: Geographic and Radiometric, Thematic classification and clustering to include unsupervised and supervised classification based on parallelepiped, minimum distance and maximum likelihood classification, accuracy assessment of classification. Applications of remote sensing, Image transformations.

Text & Reference Books:

1. Thomas Lillesand, Ralph W. Kiefer, Jonathan Chipman., Remote Sensing and Image Interpretation. Wiley
2. Reddy, M. Anji, Remote sensing and Geographic Information System BS Publications
3. B. Bhatta, Remote Sensing and GIS, Oxford University Press

Spatial Information Technology

Subject Code	**Name of Subject**	**Credit**			**Total Credit**
		L	**T**	**P**	
ECS-509	Spatial Information Technology	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes:

1. Define and explain the core concepts of Spatial Information Technology and distinguish between spatial and attribute data in various spatial models.

2. Understand and apply different representations of spatial information including regular and irregular tessellations, vector formats, and topological relationships.
3. Perform spatial data input and transformation techniques such as digitization, raster-vector conversion, and analytical transformations including rubber-sheeting.
4. Analyze and evaluate the accuracy and precision of spatial data, including positional, attribute, temporal accuracy, and data completeness.
5. Design and manage spatial databases using different DBMS models and retrieve data using spatial and attribute queries.
6. Conduct spatial analyses such as overlays, buffering, and network analysis, and apply spatial analysis tools in real-world planning and engineering applications like environmental management, transportation, and water resources.

Topics Covered

UNIT-I 9L

Definition of Spatial Information Technology, Cartography and Spatial Information, Spatial database: spatial and attribute data; Spatial models: Semantics, spatial information, temporal information, conceptual models of spatial information, Computer representation of geographic information: Regular tessellations, irregular tessellations, Vector representations, Topology and Spatial relationships, Scale and Resolution, Representation of Geographic fields, Representation of spatial objects.

UNIT-II 9L

Raster and vector data input, raster to vector data conversion, map projection, analytical transformation, rubber sheet transformation, manual digitizing and semi-automatic line following digitizer; Remote sensing data as an input to Spatial data, Direct and indirect spatial data capture Accuracy and Precision, Positional accuracy, Attribute accuracy, temporal accuracy, Lineage, Completeness, Logical consistency.

UNIT-III 9L

Spatial database Concepts and management systems, Types of Database management Systems, hierarchical, network, relational models, Object oriented DBMS, Spatial data functionality; data storage and data retrieval through query, generalization, classification, containment search within a spatial region.

UNIT-IV 9L

Overlay: arithmetical, logical and conditional overlay, buffers, inter visibility, aggregation; Network analysis; Applications of Spatial Information Technology in planning and management of utility lines and in the field of Urban Planning, Disaster Management, Smart City Projects, transportation planning and water body restoration & Climate Change.

Textbooks

1. Stan Arnoff Geographic Information Systems: A Management Perspective, WDL Publications.
2. C.P. Lo and Albert K. W. Yeung, Concepts and Techniques of Geographical Information Systems, Prentice- Hall India

3. Reddy, M. Anji, Remote sensing and Geographic Information System BS Publications Hyderabad
4. B. Bhatta, Remote Sensing and GIS, Oxford University Press

Reference books

1. Robert Laurini and Derek Thompson Fundamentals of Spatial Information Systems, Academic Press.
2. Tor Bernhardsen Geographic Information Systems: An Introduction, Wiley
3. Burrough P.A. and Rachel A. McDonell, Principles of Geographical Information Systems, Oxford Publication
4. Michael N. DeMers, Fundamentals of Geographic Information Systems, Wiley

List of Experiments

1. Familiarize with GIS software environment and basic functions on tools: QGIS/ArcGIS
2. Understand and implement raster vs. vector models and Link spatial data with tabular attribute data and visualize.
3. Digitize point, line, and polygon features from scanned maps.
4. Establish topological relationships and perform basic topology checks.
5. Apply different projections to spatial data and study distortion.
6. Digitizing and apply rubber sheet transformations to correct spatial distortions.
7. Perform sub-setting and mosaicking of adjacent location images.
8. Performing classification for land-use and land-cover on satellite imagery. Supervised Classification (Max. Likelihood, Min. Distance). Unsupervised Classification (K-means, ISODATA).
9. Compare classified image with ground truth using confusion matrix. Overall accuracy, Kappa coefficient.
10. Understand multispectral concepts using LANDSAT/IRS data. Band-wise image comparison and band combination outputs.

Web Technology for Spatial Information Mapping

Subject Code	**Name of Subject**	**Credit**			**Total Credit**
		L	**T**	**P**	
ECS-709	Web Technology for Spatial Information Mapping	3	0	2	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes (COs):

1. Explain the evolution and significance of web mapping in comparison to traditional desktop GIS systems.

2. Understand the architecture of web mapping applications including data formats, protocols, APIs, and the client-server model.
3. Identify and compare different Internet GIS software platforms such as ArcGIS Server, GeoServer, MapServer, and QGIS Server.
4. Design and develop basic web applications using HTML, XML, Java, and integrate spatial data with databases.
5. Apply knowledge of web-based spatial tools to visualize and interpret spatio-temporal data effectively.

Topics Covered

Unit I 9L

Basics of Web Mapping and Computer networks: Evolution from desktop spatial information to web spatial information, applications and importance, web mapping architecture (client-server model), key concepts: data formats, protocols, and APIs, Principles of computer networks, hardware/software, client/server computing, protocols, topologies.

Unit II 9L

History and concepts of Internet: Overview of Internet concepts & features: Internet protocol, Domain Name System, Internet services, World Wide Web, Web servers, Web clients, Internet spatial information: Application of Internet services to GIS, Internet GIS software, interoperability issues &OpenGIS. Geo-server: ArcGIS Server, GeoServer, MapServer, and QGIS server.

Unit III 9L

Web application development: Web page design principles, HTML, XML, data formats, helper applications, Java, databases and the Web.

Unit IV 9L

Application of Spatial Mapping in Real-world: Commercial utility apps such as public transports (Ola, Uber, Rapido, Chaloapp), Home delivery and (Zomato, Swiggy, Depto, Blinkit, bigBasket), public utility &spatio-temporal data visualization like WHO website, worldo-meter.

Text &Reference Books:

1. Getting to Know WEB GIS by Pindefu, Fifth Edition, 2022, Esri press.
2. GeoServer Beginner's Guide,by Stefano Iacovella, Brian Youngblood, Shroff Publishers & Distributors Pvt Ltd.
3. Beginning MapServer: Open Source GIS Development, by by Bill Kropla, Apress.
4. An Introduction to R for Spatial Analysis and Mapping, by Chris Brunsdon, Lex Comber
5. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML and Ajax, Black Book: HTML, Javascript, PHP, Java, Jsp, XML and Ajax, Black Book, by Kogent Learning Solutions Inc.
6. Advances in Web-based GIS Mapping Services and Applications, by Songnian Li,

SuzanaDragicevic, Bert Veenendaal, CRC Press.

List of Experiments

1. Analyze a simple client-server web mapping setup.Use tools like Postman to send/receive map data requests (WMS, WFS) and map tile fetching & rendering on a web client.
2. Load and convert between various spatial data formats (GeoJSON, KML, Shapefile).
3. Experiment with OGC protocols like WMS, WFS using available public servers.
4. Access and analyze publicly available Internet GIS services (Google Maps API, OpenStreetMap).Test interoperability by integrating different services.
5. Install and configure one of these servers.Publish a spatial dataset as WMS/WFS.Access the published map services using a web client or QGIS desktop.
6. Design a basic HTML page to display a map.Use XML for storing and retrieving spatial data (e.g., GML format).
7. Use Leaflet.js or Google Maps API to create an interactive web map.Add markers, popups, and layers dynamically.
8. Set up a spatial database (PostGIS or SQLite with spatial extensions).Create a web app to query spatial data and display results on a map.
9. Use helper applications like GeoJSONviewers.Create a simple Java applet or servlet that processes spatial data (optional as Java applets are deprecated).

Satellite Image Processing

Subject Code	**Name of Subject**	**L**	**T**	**P**	**Total Credit**
ECS-909	Satellite Image Processing	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes (COs):

1. Explain the fundamental concepts, system components, and steps involved in digital image processing, including image formation, sampling, quantization, and pixel relationships.
2. Apply spatial domain image enhancement techniques such as gray-level transformations, histogram processing, and spatial filtering for image improvement.
3. Analyze and implement image restoration techniques using noise models, spatial filtering, Wiener filtering, and geometric transformations.
4. Understand and apply Fourier transform concepts and frequency domain methods for image analysis and degradation estimation.

5. Perform image segmentation and object recognition using edge detection, thresholding, region-based techniques, statistical classifiers, and neural networks.
6. Utilize satellite imagery by interpreting spectral signatures and metadata; apply Python-based tools for visualization and identify objects using high-resolution remote sensing data.

Topics Covered

UNIT I **9L**

Introduction: Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system, Digital Image Fundamentals: A simple image formation model, image sampling and quantization, basic relationships between pixels .Image enhancement in the spatial domain: Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

UNIT II **9L**

Image restoration: A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function. Color Image Processing

UNIT III **9L**

Image Segmentation: Detection of discontinuous, edge linking and boundary detection, thresholding, region–based segmentation. Object Recognition : Patterns and patterns classes, recognition based on decision–theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods – matching shape numbers, string matching.

UNIT IV **9L**

Visualization of satellite images, spectral signatures, band, width, sub-setting, mosaicking, advance tool in python for data visualization. Applicability of satellite images base on metadata, spatial resolution, spectral, temporal and radiometric resolution. Object identification techniques in high-resolution images.

Text & Reference Books:

1. Computer Processing of Remotely–Sensed Images: An Introduction, by PM Mather, Fourth Edition, Wiley Publishing House.
2. Remote Sensing Digital Image Analysis, by John A. Richard, Sixth Edition, Springer Nature.
3. Digital Image Processing using MATLAB, Gonzales/ Woods/ Eddins, 3rd edition, Gatesmark Publishing, ISBN 9780982085400.
4. Fundamentals of Digital Image Processing, A K Jain, Prentice Hall, 1989, ISBN 0-13-336165-9.

5. Digital Image Processing Rafael C. González, Richard Richard Eugene Woods, Steven L.

Spatial Data Quality Assessment

Subject Code	Name of Subject	Credit			Total Credit
		L	T	P	
ECS-919	Spatial Data Quality Assessment	3	1	0	4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes

1. Define spatial data quality and explain its key dimensions and their impacts on spatial analysis.
2. Differentiate between types of positional and attribute accuracy and implement methods to measure and assess them.
3. Evaluate the completeness and consistency of spatial data, including temporal accuracy and timeliness aspects.
4. Understand and apply metadata standards to document and assess spatial data quality.
5. Identify different types of spatial data errors and model uncertainty and error propagation in spatial analysis.

Topics Covered

Unit I **9L**

Introduction to Spatial Data Quality: definition of spatial data quality, dimensions of spatial data quality (accuracy, precision, completeness, consistency, timeliness, lineage), Importance and impact of quality in spatial analysis,

Unit II **9L**

Positional and Attribute Accuracy: Types of accuracy (absolute, relative), Measuring positional accuracy, Assessing attribute accuracy, **Completeness and Consistency:** Types of completeness (logical, semantic, coverage), Consistency issues and detection methods.**Temporal Accuracy and Timeliness:** Time-related aspects of spatial data, Assessing temporal accuracy and relevance.

Unit III **9L**

Metadata and Quality Documentation: Role of metadata in spatial data quality, Standards (ISO 19157, FGDC), Creating and evaluating metadata for quality assessment. **Error Propagation and Uncertainty Modeling:** Types of errors (systematic, random), Error propagation in spatial analysis. Uncertainty quantification techniques.

Unit IV **9L**

Statistical Methods for Quality Assessment:Sampling strategies for quality assessment, Statistical testing for accuracy and precision.***Quality Assessment in Remote Sensing Data:***Radiometric and geometric accuracy, Validation of remote sensing products. ***Case Studies in Spatial Data Quality:***GIS data quality challenges in urban planning, environmental monitoring, disaster management, and transportation, ***Quality Control and Data Validation Techniques:***Automated vs. manual validation methods, Tools and software for spatial data validation.

Text & Reference Books:

1. *Fundamentals of Spatial Data Quality,*by Devillers, R., &Jeansoulin, R. (Eds.),Springer.
2. *Assuring the quality of volunteered geographic information.by* Goodchild, M. F., & Li, L. (2012).
3. Fundamentals of Spatial Data Quality, RobertJeansoulin , Rodolphe Devillers, Michael F. Goodchild, ISTE.