SYLLABUS

MASTER OF COMPUTER APPLICATIONS

MCA

2024 SCHEME

Programme Outcomes (POs)

PO1(Foundation Knowledge): Apply knowledge of mathematics, programming logic, and coding fundamentals for solution architecture and problem-solving.

PO2 (Problem Analysis): Identify, review, formulate, and analyze problems primarily focusing on customer requirements using critical thinking frameworks.

PO3 (Development of Solutions): Design, develop, and investigate problems with an innovative approach for solutions incorporating ESG/SDG goals.

PO4 (Modern Tool Usage): Select, adapt, and apply modern computational tools, such as the development of algorithms, with an understanding of the limitations, including human biases.

PO5 (Individual and Teamwork): Function and communicate effectively as an individual or team leader in diverse and multidisciplinary groups, using methodologies such as agile.

PO6 (Project Management and Finance): Apply the principles of project management, including scheduling and work breakdown structure, and be knowledgeable about finance principles for profitable project management.

PO7 (Ethics): Commit to professional ethics in managing software projects, especially in financial aspects. Learn to use new technologies for cybersecurity and insulate customers from malware.

PO8 (Life-long Learning): Continuously enhance management skills and the ability to learn, keeping up with contemporary technologies and ways of working.

Index

ST. JOSEPH'S
COLLEGE OF ENGINEERING
AND TECHNOLOGY,
- PALAI -
AUTONOMOUS

MCA TWO-YEAR

SEMESTER 3

24SJMCA201	DATA SCIENCE & MACHINE LEARNING	CATEGORY	L	T	P	CREDIT
		CORE	3	1	0	4

Preamble:	This is an introductory course on data science and basic concepts behind various machine learning techniques. Machine learning is the study of adaptive computational systems that improve their performance with experience. At the end of the course the students should be able to design and implement machine learning solutions to classification, regression, and clustering problems and to evaluate and interpret the results of the algorithms.
Prerequisite:	Probability and Statistics, Linear Algebra, Programming in Python/R.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Discuss the fundamental concepts of data science and data visualization techniques.	Understand (K2)
CO2	Apply the basics of machine learning, lazy learning and probabilistic learning algorithms to solve data science problems.	Apply (K3)
CO3	Describe decision trees, classification rules & regression methods and how these algorithms can be applied to solve data science problems.	Apply (K3)
CO4	Solve data science problems using neural networks and support vector machines.	Apply (K3)
CO5	Discuss clustering using k-means algorithm and evaluate & improve the performance of machine learning classification models.	Understand (K2)

Mapping of course outcomes with program outcomes								
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	1	1	-	-	-	-	-	3
CO2	3	3	3	-	-	-	-	3
CO3	3	3	3	-	-	-	-	3
CO4	3	3	3	-	-	-	-	3
CO5	3	3	3	-	-	-	-	3

Mark distribution

Continuous Internal Evaluation Pattern:

Attendance : 8 marks

Continuous Assessment Test (2 numbers) : 20 marks

Assignment/Quiz/Course project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is data science and why do we need data science?
2. Explain the data science classification and illustrate data science tasks.
3. Describe the various methods to understand data.
4. Explain the typical methods to visualize data.

Course Outcome 2 (CO2)

1. Explain the differences between supervised and unsupervised machine learning algorithms.
2. Describe the key concepts that define nearest neighbour classifiers, and why they are considered "lazy" learners.
3. Explain how to apply k-NN classifier in a data science problem.
4. State Bayes' theorem in statistics. Outline the Naive Bayes algorithm to build classification models.
5. Use Naive Bayes algorithm to determine whether a red domestic SUV car is a stolen car or not using the following data:

Example	Color	Type	Origin	Stolen?
1	red	sports	domestic	yes
2	red	sports	domestic	no
3	red	sports	domestic	yes
4	yellow	sports	domestic	no
5	yellow	sports	imported	yes
6	yellow	SUV	imported	no

7	yellow	SUV	imported	yes
8	yellow	SUV	domestic	no
9	red	SUV	imported	no
10	red	sports	imported	yes

Course Outcome 3 (CO3):

1. Classify data science tasks using decision trees and classification rule learners.
2. Discuss the various feature selection measures.
3. How to simplify a decision tree by pruning.
4. Describe how to construct classification rules from decision trees.
5. Explain the concepts of regression and correlation.
6. How to estimate a linear regression model.
7. Consider the following set of training examples:

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

a) Find the entropy of this collection of training examples with respect to the target function "classification"?
b) Calculate the information gain of a2 relative to these training examples?

Course Outcome 4 (CO4):

1. Explain how artificial neural networks mimic human brain to model arbitrary functions and how these can be applied to real-world problems.
2. Describe different activation functions and network topology.
3. Discuss basic idea behind the backpropagation algorithm.
4. Explain how a support vector machine can be used for classification of linearly separable data.
5. How to compute the distance of a point from a hyperplane.
6. How the kernel trick is used to construct classifiers in nonlinearly separated data.

Course Outcome 5 (CO5):

1. Explain how the clustering tasks differ from the classification tasks.
2. How clustering defines a group, and how such groups are identified by k-means clustering algorithm.
3. Find the three clusters after one epoch for the following eight examples using the k- means algorithm and Euclidean distance: A1 = (2,10), A2 = (2,5), A3 = (8,4), A4 = (5,8), A5 = (7,5), A6 = (6,4), A7 = (1,2), A8 = (4,9). Suppose that the initial seeds (centres of each cluster) are A1, A4 and A7.
4. Explain the various matrices used to measure the performance of classification algorithms
5. Explain the concepts of bagging and boosting.
6. Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data.

Syllabus

MODULE	CONTENTS	HOURS
I	**Introduction to data science:** Data science classification, Data science process – Prior knowledge, Data preparation, Modelling, Application. **Data exploration:** Data sets, Descriptive statistics for univariate and multivariate data. **Data visualisation:** Histogram, Quartile plot, Distribution chart, Scatter plot, Bubble chart, Density chart	**9**

II	**Introduction to machine learning:** How machines learn - Data storage, Abstraction, Generalisation, Evaluation, Machine learning in practice - Types of machine learning algorithms. **Lazy learning:** Classification using K-Nearest Neighbour algorithm - Measuring similarity with distance, Choice of k, Preparing data for use with k-NN. **Probabilistic learning:** Understanding Naive Bayes - Conditional probability and Bayes theorem, Naive Bayes algorithm for classification, The Laplace estimator, Using numeric features with Naive Bayes.	**10**
III	**Decision tree learning:** Concept of decision tree, Divide and conquer approach, C5.0 Decision tree algorithm, Choosing the best split, Pruning the decision tree. **Classification rules learning:** Concept of classification rules, Separate and conquer approach, The 1R algorithm, Rules from decision trees. **Regression methods:** Concept of regression, Simple linear regression, Ordinary least squares estimation, Correlations, Multiple linear regression.	**10**
IV	**Neural Network Learning:** Artificial neurons, Activation functions, Network topology, Training neural networks with backpropagation. **Support Vector Machines:** Hyperplanes, Classification using hyperplanes, Maximum margin hyperplanes in linearly separable data, Using kernels for non-linear spaces.	**10**
V	**Clustering:** The k-means clustering algorithm, Using distance to assign and update clusters, Choosing number of clusters. **Evaluating model performance:** Confusion matrices, Precision and recall, Sensitivity and specificity, Precision and recall, F-measure, ROC curves, Cross validation - K-fold cross validation, Bootstrap sampling. **Improving model performance:** Bagging, Boosting, Random forests.	**8**

Text Books

1. Vijay Kotu, Bala Deshpande, Data Science Concepts and Practice, Morgan Kaufmann Publishers 2018 (Module 1)

2. Brett Lantz, Machine Learning with R, second edition, PackT publishing 2015 (Modules 2 to 5)

Reference Books

1.Michael Steinbach, Pang-Ning Tan, and Vipin Kumar, Introduction to Data Mining, Pearson 2016.

2. Jiawei Han, Micheline Kamber and Jian Pei, Data mining Concepts and techniques Morgan Kaufmann Publishers 2012

3. Peter Harrington, Machine Learning in action, Dreamtech publishers 2012

4. Dr M Gopal, Applied Machine learning, McGraw Hill Education Private Limited

5. E. Alpayidin, Introduction to Machine Learning, Prentice Hall of India (2005)

6.T. Hastie, RT Ibrashiran and J. Friedman, The Elements of Statistical Learning, Springer 2001

7.Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly, First edition, 2015

8.Introducing Data Science, Davy Cielen, Arno D. B. Meysman, Mohamed Ali, Manning Publications Co., 1st edition, 2016

Web Resources:

1. https://www.coursera.org/learn/machine-learning

2. https://www.coursera.org/learn/data-scientists-tools

Course Contents and Lecture Schedule:

No	Topic	No: of Lectures
1	**Module 1**	
1.1	Introduction to data science - What is data science? Why data science?	2hrs
1.2	Data science classification	1 hr
1.3	Data science process - Prior knowledge, Data preparation, Modelling, Application	2 hrs
1.4	Data exploration- Data sets, Descriptive statistics for univariate and multivariate data	2 hrs
1.5	Data visualization – Histogram, Quartile plot, Distribution chart, Scatter plot, Bubble chart, Density chart	2 hrs
2	**Module 2**	
2.1	How machines learn – Data storage – Abstraction Generalisation – Evaluation	1 hr
2.2	Machine learning in practice – Types of machine learning algorithms	1hr
2.3	Classification: Lazy learning - K-Nearest Neighbour algorithm	2 hrs
2.4	Measure of similarity, Choice of k	1 hr
2.5	Preparing data for use with k-NN	1 hr

2.6	Probabilistic Learning: Conditional probability and Bayes theorem.	2hrs
2.7	Naive Bayes algorithm 2	2 hrs
3	**Module 3**	
3.1	Concept of decision tree, Divide and conquer approach	1 hr
3.2	C5.0 Decision tree algorithm	1 hr
3.3	Choosing the best split, Pruning the decision tree	2 hrs
3.4	Classification rules learning: Concept of classification rules, Separate and conquer approach	2 hrs
3.5	The 1R algorithm, Rules from decision trees	1 hr
3.6	Regression methods: Concept of regression, Correlation	1 hr
3.7	Simple linear regression, Ordinary least squares estimation	1 hr
3.8	Multiple linear regression	1 hr
4	**Module 4**	
4.1	Understanding neural networks - Artificial neurons	1 hr
4.2	Activation functions, Network topology	2 hrs
4.3	Training neural networks with back propagation	1 hr
4.4	Understanding Support Vector Machines, Classification with hyperplane	2 hrs
4.5	Linearly separable data, Nonlinearly separable data	1 hr
4.6	Methods to find maximum margin hyperplanes in linearly separable data	1 hr
4.7	Using kernels for non-linear spaces	2 hrs
5	**Module 5**	
5.1	Understanding Clustering - The k-means clustering algorithm	1 hr
5.2	Using distance to assign and update clusters, Choosing the appropriate number of clusters	1 hr
5.3	Evaluating model performance: Confusion matrices, Precision and recall, Sensitivity and specificity, Precision and recall, F-measure, ROC curves.	2 hrs
5.4	Cross validation: K-fold cross validation, Bootstrap sampling	2 hrs
5.5	Improving model performance: Bagging, Boosting	1 hr
5.6	Random forests	1 hr

24SJMCA203	DESIGN & ANALYSIS OF ALGORITHMS	CATEGORY	L	T	P	CREDIT
		CORE	3	1	0	4

Preamble:	The syllabus is prepared with a view to provide a strong foundation to students in design and analysis of computer algorithms and to introduce them the advanced topics such as Network Flows, Approximation algorithms and Randomised algorithms.
Prerequisite:	Knowledge in Data Structures

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Discuss the basic concepts in computer algorithms and their analysis & design using Divide and Conquer.	K2
CO2	Explain the concepts of Greedy Strategy and Dynamic Programming to use it in solving real world problems.	K3
CO3	Explain the Branch & Bound technique, Backtracking technique and Lower bounds.	K2
CO4	Describe the fundamental concepts of Computational Complexity and Network Flows.	K2
CO5	Discuss the concepts of Approximation and Randomised Algorithms.	K2

Mapping of course outcomes with program outcomes								
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2					
CO2	3	2	2					
CO3	3	2	2					
CO4	3	2	2					
CO5	3	2	1					

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 8 marks
Continuous Assessment Test (2 numbers) : 20 marks
Assignment/Quiz/Course project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains10 compulsory short answer questions, 2 from each module. Each question carries

3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

- Define "Time Complexity" of an algorithm?
- What is the need for analysing an algorithm?
- Define Big Oh Notation.
- Define the terms Best Case, Worst Case and Average case complexities.
- Explain the Merge Sort algorithm with an example

Course Outcome 2 (CO2)

- Explain the Greedy Control abstraction.
- Write the Prim's algorithm and illustrate with an example.
- State and illustrate the Principle of Optimal Substructure.
- Explain a solution to the Travelling Salesman problem using Dynamic Programming.

Course Outcome 3(CO3):

- Explain the N-Queen's problem and its solution using Backtracking.
- Explain the 8-puzzle problem and illustrate how it can be solved using Branch and Bound.
- Bring out the notion of Decision Trees.
- What is the lower bound of the time complexity of Comparison based sorting algorithms?.

Course Outcome 4 (CO4):

- Define class P and NP.
- What is Polynomial Time Reduction?
- Show that the Clique problem is NP-Complete.
- Define the Terms - Flow Network and Network Flow.
- Explain the Ford-Fulkerson Algorithm.

Course Outcome 5 (CO5):

- What is an Approximation algorithm?
- Describe the 2-approximation algorithm for Vertex Cover problem.
- What is a Randomised algorithm?
- Explain the Schwartz-Zippel Lemma. How this Lemma can be used to test the identity of two polynomials.

Syllabus

Module	Contents	Hours
I	**Review of Algorithm Analysis**: Time and Space Complexity, Asymptotic Notations, Recurrence Equations, Solving Recurrence Equations- Substitution method and Iteration method. **Divide and Conquer**: Control Abstraction, Merge Sort, Quick Sort, Matrix Multiplication.	8
II	**Greedy Strategy**: Control Abstraction, Knapsack Problem, Minimal Spanning Tree Algorithms- Prim's and Kruskal's Algorithm, Job Scheduling with deadlines **Dynamic Programming**: Control Abstraction, Principle of Optimal Substructure, All Pairs shortest path problem, Travelling Salesman Problem, Bellman-Ford Algorithm	9
III	**Backtracking:** Control Abstraction, N-Queens problem, Sum of Subsets Problem **Branch and Bound**: Control Abstraction, 8- Puzzle problem **Lower Bounds:** The Decision Tree method, Lower Bounds for Comparison based Sort and Searching *(Analysis not required)*	7
IV	**Complexity Theory**: Class P and NP, Polynomial time reductions, Class NP Hard and NP- Complete, Example Problems- Vertex Cover problem, Clique Problem. **Network Flows**: Flow Networks and Network Flow, Max- Flow Min Cut Theorem, Ford Fulkerson method, Bipartite matching (*Analysis not required)*	11
V	**Introduction to Approximation Algorithms**: Approximation Ratio, 2-approximation algorithm for Vertex Cover problem, Vertex Cover Approximation using Linear Programming and LP Rounding Algorithm. **Introduction to Randomised Algorithms**: Review of Basic Probability, Schwartz-Zippel Lemma and Polynomial Identity Testing, Randomized Quick Sort (*Proof of Expected Worst Case Analysis not required)*	10

Text Books

1. Thomas H. Cormen, et al., "Introduction to Algorithms", Prentice Hall, 3rd Edition (2010)
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Orient Longman, Universities Press, 2nd Edition (2008)

Reference Books

1. Richard Neapolitan, Kumarss Naimipour, "Foundations of Algorithms", Jones and Bartlett Publishers, Inc, 4th Edition (2011).
2. Sara Baase, Allen Van Gelder, "Computer Algorithms: Introduction to Design and Analysis", Pearson India, 3rd Edition (2002).
3. A. Levitin, "Introduction to the Design & Analysis of Algorithms", Pearson Education, 3rd Edition (2008).

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Review of Algorithm Analysis and Divide & Conquer**	**8 Hours**
1.1	Time and Space Complexity	1
1.2	Asymptotic Notations	1
1.3	Recurrence Equations, Solving Recurrence Equations- Substitution method	1
1.4	Iteration method	1
1.5	Divide and Conquer: Control Abstraction, Merge Sort, Merge Sort Analysis	2
1.6	Quick Sort, Quicksort analysis	1
1.7	Matrix Multiplication	1
2	**Greedy Strategy and Dynamic Programming**	**9 Hours**
2.1	Greedy Strategy: Control Abstraction, Knapsack Problem	1
2.2	Minimum Cost Spanning Tree	1

2.3	Prim's algorithm	1
2.4	Kruskal's algorithm	1
2.5	Job Scheduling with deadlines	1
2.6	Dynamic Programming: Control Abstraction, Principle of Optimal substructure	1
2.7	All Pairs shortest path problem	1
2.8	Travelling Salesman Problem	1
2.9	Bellman-Ford Algorithm	1
3	**Backtracking, Branch & Bound, Lower Bounds**	**7 Hours**
3.1	Backtracking: Control Abstraction N- Queens problem	1
3.2	Sum of subsets problem	1
3.3	Branch and Bound: Control Abstraction 8- Puzzle problem	1
3.4	Lower Bounds: The Decision Tree method	2
3.5	Lower Bounds for Comparison based Sorting	1
3.6	Lower bounds for searching	1
4	**Computational complexity, Network Flows**	**11 Hours**
4.1	Class P, NP	1
4.2	Polynomial Time Reductions	1
4.3	Class NP-Hard and NP-Complete	2
4.4	Vertex Cover Problem	1
4.5	Clique problem	1

4.6	Flow Networks and Network Flows	2
4.7	Max Flow Min Cut Theorem	1
4.8	Ford Fulkerson's method	1
4.9	Bipartite matching	1
5	**Approximation & Randomised Algorithms**	**10 Hours**
5.1	Approximation algorithms- introduction, Approximation Ratio	1
5.2	2- approximation algorithm for Vertex Cover problem	1
5.3	Vertex Cover Approximation using Linear Programming and LP Rounding Algorithm	2
5.4	Randomized Algorithms: introduction, Review of Basic Probability	1
5.5	Review of Basic probability	2
5.6	Schwartz-Zippel Lemma and Polynomial Identity Testing	2
5.7	Randomized Quick Sort	1

24SJMCA263	CYBER SECURITY & CRYPTOGRAPHY	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble:	This course is designed to provide theoretical concepts used in cryptography and to introduce the students to various cryptographic algorithms and techniques used for implementing data security and protection. This course also discusses common web application security vulnerabilities.
Prerequisite:	Student is expected to have studied mathematics courses that cover Elementary Number Theory.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Explain various types of security attacks, security mechanisms, security services and classical encryption techniques.	K2
CO2	Make use of Symmetric and Asymmetric encryption techniques to solve cryptographic problems.	K3
CO3	Describe the concepts of message authentication codes, hash functions and digital signing techniques for ensuring secure transactions.	K2
CO4	Discuss security services in Application, Transport and Network layers.	K2
CO5	Explain common web application security vulnerabilities and various prevention mechanisms.	K2

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	1				1	1
CO2	2	2	2	1			1	1
CO3	2	1	1				1	1
CO4	2	1	1				1	1
CO5	2	2	1	1			1	1

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 8 marks
Continuous Assessment Test (2 numbers) : 20 marks
Assignment/Quiz/Course project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

- Briefly explain each component of OSI security architecture.
- Compare Substitution and Transposition techniques in cryptography.
- Explain how steganography is used in cryptography.

Course Outcome 2 (CO2)

- Explain block cipher modes of operation.
- Compare DES and AES.
- Perform encryption and decryption using RSA Algorithm with parameters: P=17, q = 11, e = 7, M = 88.

Course Outcome 3(CO3):

- Compare the features of HMAC and CMAC algorithms.
- Explain important steps in DSS.
- Describe the terms (a) birthday attack (b) hash cash (c) blind signature

Course Outcome 4 (CO4):

- Explain any one protocol used in E-mail for security.
- Explain how security is provided in Network Layer using IPsec.
- Describe the process of securing electronic transactions.

Course Outcome 5 (CO5):

- Discuss any four Application Security Risks.
- Which are the different forms of XSS and how to prevent these?
- Explain the attack scenario of any four web application security vulnerabilities.

Syllabus

Module	Contents	Hours
I	**Introduction to Cryptography**, OSI security architecture: Security Services, Mechanisms and attacks- Phishing, Ransomware, DoS attack. Network security model. Classical Encryption techniques - Symmetric cipher model, substitution techniques, transposition techniques. Steganography.	7
II	**Conventional Symmetric Key Encryption**: Block ciphers and Stream Ciphers, Block Cipher Design Principles, Modes of operation, Data Encryption Standard, Advanced Encryption Standard (AES), Multiple Encryption, Triple DES. **Public key cryptography:** Principles of public key cryptosystems-The RSA algorithm-Key management – Diffie Hellman Key exchange - Elliptic curve arithmetic - Elliptic curve cryptography.	10
III	**Hash Functions and MAC:** Properties of hash functions, birthday attack, hashcash, Message Authentication Code Algorithms, MAC protocols: HMAC, CMAC. **Digital Signatures**: Classification of signature schemes: RSA signature, Digital Signature Standard, Overview of ElGamal and Schnorr schemes. One time signature schemes, Attacks on Digital Signatures, Blind Signatures.	10
IV	**Introduction to Cyber Security, Email Security**: Security Services for email, Attacks possible through email, Establishing keys privacy, authentication of the source, Message Integrity, Non-repudiation, Pretty Good Privacy, S/MIME. **IP Security:** Overview of IPSec, IPv4 and IPv6, Authentication Header, Encapsulation Security Payload (ESP), Internet Key Exchange. **Transport Level Security**: SSL/TLS Basic Protocol, computing the keys, client authentication, PKI as deployed by SSL, Attacks fixed in v3, Exportability, Encoding, Secure Electronic Transaction (SET).	10

V	**Common web application security vulnerabilities**: Injection flaws, Broken authentication, Sensitive data exposure, XML External Entities (XXE), Broken access control, Security misconfiguration, Cross-Site Scripting (XSS), Insecure deserialization, Using components with known vulnerabilities, Insufficient logging & monitoring. Example attack scenarios of each of the vulnerabilities listed; how to prevent them.	8

Text Books

1. William Stallings, "Cryptography and Network Security," 6th Edition, Pearson Education.
2. Behrouz A. Forouzan, "Introduction to Cryptography and Network Security", Tata McGraw-Hill Publishing 2nd Edition.

Reference Books

1. Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security", Prentice Hall of India.
2. Manuel Mogollon, "Cryptography and Security Services – Mechanisms and Applications", Cybertech Publishing.
3. William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, "Firewalls and Internet Security" Addison- Wesley.

Web References

1. http://www.hashcash.org/hashcash.pdf [Reference for hash cash]
2. https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf. **[Reference for Module 5]**
3. https://www.coursera.org/learn/crypto
4. https://www.coursera.org/learn/crypto2

Course Contents and Lecture Schedule

No.	Topic	No. of Lecture
1	**Introduction to Cryptography**	**7 Hours**
1.1	What is cryptography, Related Terms, Need of cryptosystems	1

1.2	OSI security architecture: Security Services, Mechanisms	1
1.3	Security attacks- Phishing, Ransomware, DoS attack.	1
1.4	Network security model	1
1.5	Classical Encryption techniques, Symmetric cipher model	1
1.6	Substitution techniques	1
1.7	Transposition techniques, Steganography	1
2	**Conventional Symmetric and Public Key Encryption**	**10 Hours**
2.1	Block ciphers and Stream Ciphers, Block Cipher Design Principles	1
2.2	Modes of operation	1
2.3	Data Encryption Standard	1
2.4	Advanced Encryption Standard (AES)	1
2.5	Multiple Encryption, Triple DES	1
2.6	Public key cryptography: Principles of public key cryptosystems	1
2.7	The RSA algorithm	1
2.8	Key management	1
2.9	Diffie Hellman Key exchange	1
2.10	Elliptic curve arithmetic - Elliptic curve cryptography.	1
3	**Hash Functions and MAC**	**10 Hours**
3.1	Properties of hash functions, birthday attack	1
3.2	Hashcash, Message Authentication Code Algorithms	1
3.3	MAC protocols: HMAC, CMAC	1

3.4	Digital Signatures: Classification of signature schemes	1
3.5	RSA signature	1
3.6	Digital Signature Standard	1
3.7	Overview of ElGamal and Schnorr schemes	1
3.8	One time signature schemes	1
3.9	Attacks on Digital Signatures	1
3.10	Blind Signatures	1
4	**Introduction to Cyber Security**	**10 Hours**
4.1	Email Security: Security Services for email, Attacks possible through email	1
4.2	Establishing keys privacy, authentication of the source, Message Integrity, Non-repudiation	1
4.3	Pretty Good Privacy, S/MIME	1
4.4	IP Security: Overview of IPSec	1
4.5	IPv4 and IPv6, Authentication Header	1
4.6	Encapsulation Security Payload (ESP), Internet Key Exchange	1
4.7	Transport Level Security: SSL/TLS Basic Protocol	1
4.8	computing the keys, client authentication, PKI as deployed by SSL	1
4.9	Attacks fixed in v3, Exportability, Encoding	1
4.10	Secure Electronic Transaction (SET)	1
5	**Common web application security vulnerabilities**	**8 Hours**
5.1	Common web application security vulnerabilities	1

5.2	Injection flaws, Broken authentication	1
5.3	Sensitive data exposure, XML External Entities (XXE)	1
5.4	Broken access control, Security misconfiguration	1
5.5	Cross-Site Scripting (XSS), Insecure deserialization	1
5.6	Using components with known vulnerabilities, Insufficient logging & monitoring.	1
5.7	Example attack scenarios of each of the vulnerabilities listed	1
5.8	How to prevent each of the vulnerabilities.	1

24SJMCA265	CLOUD COMPUTING	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble:	The syllabus is prepared with a view to equip the students to learn basic concepts in cloud computing - compute, storage, networking. They should gain basic understanding of orchestration, HA and failover.
Prerequisite:	Awareness in Virtualisation and Containers is desirable.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Understand the basic concepts in cloud computing and OpenStack logical architecture	K2
CO2	Discuss OpenStack cloud controller and common services	K3
CO3	Compare different OpenStack compute service components and storage types	K2
CO4	Describe the OpenStack Networking- Connection types and networking services	K2
CO5	Discuss orchestration, HA and failover in OpenStack	K2

Mapping of course outcomes with program outcomes								
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	2			2				2
CO2	3	2		2				2
CO3	2	2		2				2
CO4	2	2		2				2
CO5	2	2	2	2				2

Mark Distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 8 marks
Continuous Assessment Test (2 numbers) : 20 marks
Assignment/Quiz/Course Project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum 2 subdivisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. List and explain various components of Nova compute service.
2. Explain the neutron architecture?
3. Briefly describe keystone identity management.

Course Outcome 2 (CO 2):

1. Explain the telemetry services in OpenStack.
2. Explain the steps involved in bringing up a working OpenStack Ansible on the deployment host.
3. Explain the steps in network configuration

Course Outcome 3 (CO 3):

1. Explain briefly swift architecture
2. Briefly explain how data is handled in the cluster by swift
3. What is meant by CPU over commitment?

Course Outcome 4 (CO 4):

1. Explain steps in associating a floating IP to a virtual machine.
2. Briefly explain the steps in creating a virtual network with two subnets
3. Briefly explain Linux bridge-based connectivity?

Course Outcome 5 (CO 5):

1. Briefly explain the major components in heat?
2. Explain the different metrics that can be measured in a highly available infrastructure?
3. Explain the need for Service level agreement.

Syllabus

Module 1: Overview of OpenStack (7 Hours)
Introduction to cloud computing, private cloud, public cloud, hybrid cloud architecture. Cloud Services - Infrastructure as a Service, Platform as a Service, Storage as a Service. Designing OpenStack Cloud Architectural Consideration - OpenStack - The new data centre paradigm - OpenStack logical architecture - Nova - Compute Service-Neutron - Networking services - Gathering the pieces and building a picture - A sample architecture setup.
Module 2: OpenStack cluster - Controller and common services (6 Hours)
OpenStack Cluster – The Cloud Controller and Common Services- Asymmetric clustering, Symmetric clustering, The cloud controller - The keystone service. The nova-conductor service, The nova-scheduler service, The API services, Image management, The network service, The horizon dashboard, The telemetry services.
Module 3: OpenStack compute and Storage (12 Hours)
OpenStack Compute -The compute service components - Deciding on the hypervisor - OpenStack Magnum Project - Segregating the compute cloud - Overcommitment considerations - Storing instances' alternatives - Understanding instance booting - Planning for service recovery. OpenStack Storage - Block, Object, and File Share - Understanding the storage types - Ephemeral Storage - Persistent storage - A spotlight on Swift - Deploying Swift service - Using block storage service: Cinder.
Module 4: OpenStack Networking (10 Hours)
The architecture of Neutron - Implementing virtual networks - Connecting virtual networks with routers - Implementing network security in OpenStack. OpenStack Networking - The architecture of Neutron - Implementing virtual networks - VLAN, Tunnel based, Virtual Switches, The ML2 Plugin. Neutron Subnets - Connecting virtual networks with routers - Configuring the routing service - connecting networks using a virtual router, connecting to the external world, connectivity from the external world, associating a floating IP - Implementing network security in OpenStack
Module 5: OpenStack Orchestration, HA and failover (10 Hours)
Orchestration in OpenStack - Heat and its Components, stacking in OpenStack, OpenStack Orchestration with Terraform. OpenStack HA and failover: Scope of HA in OpenStack, HA in the database, HA in the Queue, Implementing HA on RabbitMQ.

Text Book

1. Omar Khedher, Chandan Datta Chowdhury, Mastering OpenStack, 2nd Edition, Packt Publishing, 2017

Reference Books

1. Tom Fifield, Diane Fleming, Anne Gentle,Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, OpenStack Operations Guide, O'REILY,1/e, 2014.
2. Uchit Vyas, Applied OpenStack Design Patterns, Apress, 1/e, 2016.
3. V. K. Cody Bumgardner, OpenStack in action, Manning, 2016.
4. Amar Kapadia, Sreedhar Varma, Kris Rajana, Implementing Cloud Storage with OpenStack Swift, Packt Publishing, 2014.
5. https://docs.openstack.org/wallaby/?_ga=2.231002015.1428061357.1620834394-1139122985.1620834394

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Overview of OpenStack**	**7 Hours**
1.1	Introduction to cloud computing, private cloud, public cloud, hybrid cloud architecture.	1
1.2	Cloud Services - Infrastructure as a Service, Platform as a Service, Storage as a Service	1
1.3	Designing OpenStack Cloud Architectural Consideration - OpenStack - The new data center paradigm -OpenStack logical architecture	1
1.4	Nova - Compute service	1
1.5	Neutron - Networking services	1
1.6	Gathering the pieces and building a picture	1
1.7	A sample architecture setup	1
2	**OpenStack cluster - Controller and common services**	**6 Hours**
2.1	OpenStack Cluster – The Cloud Controller and Common Services- Asymmetric clustering, Symmetric clustering	1
2.2	The cloud controller - The keystone service	2
2.3	The nova-conductor service, The nova-scheduler service, The API services, Image management.	1
2.4	The network service, The horizon dashboard, The telemetry services	2

3	**OpenStack compute and Storage**	**12 Hours**
3.1	The compute service components-Deciding on the hypervisor-OpenStack Magnum project	1
3.2	Segregating the compute cloud	1
3.3	Overcommitment considerations	1
3.4	Storing instances' alternatives	1
3.5	Understanding instance booting	1
3.6	Planning for service recovery	1
3.7	OpenStack Storage - Block, Object, and File Share-Understanding the storage types	1
3.8	A spotlight on swift	2
3.9	Deploying swift service	1
3.10	Using Block Storage Service Cinder	2
4	**OpenStack Networking**	**10 Hours**
4.1	The architecture of Neutron	1
4.2	Implementing virtual networks - VLAN, Tunnel based	1
4.3	Virtual Switches, The ML2 Plugin	1
4.4	Neutron Subnets	2
4.5	Connecting virtual networks with routers - Configuring the routing service	1
4.6	Connecting networks using a virtual router, Connecting to the externa world	1
4.7	Connectivity from the external world, Associating a floating IP	1
4.8	Implementing network security in OpenStack	2
5	**OpenStack Orchestration, HA and Failover**	**10 Hours**
5.1	Orchestration in OpenStack, Heat and its Components	1
5.2	Stacking in OpenStack	2
5.3	OpenStack Orchestration with Terraform	2

5.4	Scope of HA in OpenStack	2
5.5	HA in the database	1
5.6	HA in the Queue, Implementing HA on RabbitMQ	2

Suggested Assignments

1) Create VMs in your physical machine using OpenStack to set up the following services: Moodle, MySQL Server, Samba. Design the desired configuration of the physical machine to handle the requirements of the entire college.
2) Set up storage services for storing external files for Moodle.
3) Set up firewall rules for samba, MySQL server, allow the connection to MySQL server only to Moodle VM.
4) Set up recovery plans for the above services
5) Convert the MySQL server to HA MySQL server.
6) Setup a load balancer for the Moodle server.

		CATEGORY	L	T	P	CREDIT
24SJMCA281	**INTERNET OF THINGS**	**ELECTIVE**	3	1	0	4

Preamble:	This course intends to provide insight into new innovations that will build novel type of interactions among things and humans, and enables the realization of smart cities, infrastructures, and services for enhancing the quality of life and utilization of resources. An overview of IoT and its related concepts, different IoT architectures and their components, emerging paradigms such as Fog computing, Platforms and solutions supporting development and deployment of IoT applications, message passing mechanisms such as RPC, REST, and CoAP, data and knowledge management, data confidentiality, data integrity, and operation control issues faced by IoT are included in the course.
Prerequisite:	Basic concepts of Information Technology and Internet.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Describe the main concepts and features of the IoT paradigm.	Level 2: Understand
CO2	Discuss Fog computing, TinyOS - nesC and programming frameworks for IoT	Level 2: Understand
CO3	Describe the data management techniques applied to the IoT environment.	Level 2: Understand
CO4	Explain security, and privacy in IOT environments	Level 2: Understand
CO5	Discuss key enablers and solutions to enable practical IoT systems	Level 2: Understand

Mapping of course outcomes with program outcomes								
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	0	2	2	0	0	0	2
CO2	3	0	2	3	0	0	0	2
CO3	3	0	3	3	1	0	0	2
CO4	2	0	3	3	1	0	0	2
CO5	3	1	3	3	2	2	0	3

3/2/1: High/Medium/Low

Mark Distribution

Total Marks	CIA	ESE	ESE Duration
100	40	60	3 Hours

Continuous Internal Evaluation Pattern:

Attendance : 8 marks
Continuous Assessment Test (2 numbers) : 20 marks
Assignment/Quiz/Course project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum of 2 subdivisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1)

1. Compare SOA-based architecture and API-oriented architecture.
2. Neatly sketch the open IoT architecture for IoT/CLOUD convergence.
3. List and explain the applications of device/cloud collaboration.

Course Outcome 2 (CO2)

1. What are the advantages associated with Fog computing?
2. Comment on the four broad requirements that motivate the design of TinyOS.
3. Summarize the communication paradigms and technologies used in resource-constrained environments.

Course Outcome 3(CO3)

1. Explain stream and stream processing in IoT.
2. Write and explain the algorithm for distributed anomaly detection by clustering ellipsoids.
3. Discuss the general architecture of a stream-processing system in IoT.

Course Outcome 4 (CO4)

1. Give an overview on the security requirements of IoT.
2. How can you nullify the impact of fault in high-availability cluster?
3. Explain the BCK with pre-shared keys for TinyTO.

Course Outcome 5 (CO5)

1. Give an overview on the Wired Gateway Interfaces.
2. List the features to select the gateway hardware.
3. List the steps to prepare Raspberry Pi for the execution.

Syllabus

Module 1 (9 Hours)
Overview of Internet of Things: Open-source semantic web infrastructure for managing IoT resources in the Cloud - Device/Cloud Collaboration framework for intelligence applications.
Module 2 (11 Hours)
Introduction to Fog Computing: principles, architectures, and applications. TinyOS - NesC, Programming frameworks for Internet of Things
Module 3 (8 Hours)

Stream processing in IoT: foundations, state-of-the-art, and future directions - A framework for distributed data analysis for IoT
Module 4 (9 Hours)
Security and privacy in the Internet of Things- Internet of Things - robustness and reliability. TinyTO: two-way authentication for constrained devices in the Internet of Things - Obfuscation and diversification for securing the Internet of Things
Module 5 (8 Hours)
Creating a simple IoT project - Preparing Raspberry Pi Interfacing the hardware – Internal representation of sensor values- Persisting data - Creating the actuator project - Creating a controller.

More detailed knowledge may be acquired through seminars, assignments and talks by eminent external experts and also by implementing a micro project.

Any one of the following or similar micro projects may be given as part of the course.
1. Smart Gas Leakage Detector
2. Night Patrol at home

Text Books
1. RajkumarBuyya; Amir VahidDastjerdi ,

Reference Books
S. SitharamaIyengar; Nandan Parameswaran; Vir V. Phoha; N. Balakrishnan; Chuka Okoye, December 14, 2010

Web Resources
1. https://www.coursera.org/specializations/internet-of-things
2. http://web.mit.edu/professional/digital-programs/courses/IoT

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction**	**9 Hrs**
1.1	Overview of Internet of Things	3
1.2	Open-source semantic web infrastructure for managing IoT resources in the Cloud	3
	Device/Cloud Collaboration framework for intelligence applications.	3
2	**Programming frameworks**	**(11 Hours)**
2.1	Introduction to Fog Computing	2
2.2	principles, architectures, and applications	3
2.3	TinyOS - NesC	3
2.4	Programming frameworks for Internet of Things	3
3	**Data management techniques**	**(8 Hours)**
3.1	Stream processing in IoT	2
3.2	foundations, state-of-the-art, and future directions	3

3.3	A framework for distributed data analysis for IoT	3
4	**Security and privacy**	**(9 Hours)**
4.1	Security and privacy in the Internet of Things	2
4.2	Internet of Things - robustness and reliability	2
4.3	TinyTO: two-way authentication for constrained devices in the Internet of Things	2
4.4	Obfuscation and diversification for securing the Internet of Things	3
5	**IoT Implementation**	**(8 Hours)**
5.1	Creating a simple sensor project	2
5.2	Preparing Raspberry Pi	1
5.3	Interfacing the hardware –---	1
5.4	Internal representation of sensor values	1
5.5	Persisting data	1
5.6	Creating the actuator project	1
5.7	Creating a controller	1

24SJMCA283	DEEP LEARNING	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble:	This course intends to provide insight into deep learning. This topic is currently a much sought-after skill and is under active research. Students have to refer appropriate research papers and multiple books to get in-depth knowledge about the topics**.** Instructors may give suitable programming assignments to augment the material covered in the classroom.
Prerequisite:	Basic concepts of linear algebra, probability and optimization.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Explain the basic concepts of deep learning.	K2
CO2	Design neural networks using TensorFlow	K3
CO3	Solve real world problems with CNN.	K3
CO4	Solve real world problems with RNN.	K3
CO5	Describe the concepts of GAN.	K2

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2		2				2
CO2	3	2	3	3				2
CO3	3	3	3	3				2
CO4	3	3	3	3				2
CO5	3	2	3	2				2

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 8 marks
Continuous Assessment Test (2 numbers) : 20 marks
Assignment/Quiz/Course project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the model of a biological neuron.
2. Explain Perceptron learning algorithm.
3. Explain the role of batch normalization in training a neural network.

Course Outcome 2 (CO2)

1. Draw and demonstrate the VGG-16 architecture.
2. Sketch the AlexNet architecture and explain its functionalities.

Course Outcome 3(CO3):

1. Design a convolutional neural network which can classify MNIST handwritten data.
2. An input image has been converted into a matrix of size 12 X 12 along with a filter of size 3 X 3 with a Stride of 1. Determine the size of the convoluted matrix.
3. Why do we prefer Convolutional Neural networks (CNN) over Artificial Neural networks (ANN) for image data as input?

Course Outcome 4 (CO4):

1. You are given an image data set with 10 classes. Describe how you will use deep learning to build a classifier.
2. Design a system to generate deep fakes from an image.

Course Outcome 5 (CO5):

1. Describe auto encoders and how they help in dimensionality reduction.
2. Explain how GANS work.

Syllabus

Module I (8 Hours)
Review of Neural Networks: Model of a biological neuron, McCulloch Pitts Neuron, Activation Functions, Perceptron, Perceptron Learning Algorithm and Convergence, Multilayer Perceptron, Back propagation, Learning XOR, Sigmoid Neurons, Gradient Descent, Feed forward Neural Networks.
Module II (10 Hours)
Training Neural Networks: Initialization, dropout, batch normalization and dropout, overfitting, underfitting, training and validation curves. Data Visualization: Feature and weight visualization, tSNE. Introduction to TensorFlow: graphs, nodes, Tensor data structures - rank, shape, type, Building neural networks with TensorFlow, Introduction to Keras.
Module III (10 Hours)
Convolutional Neural Networks: Convolution operation, Convolutional layers in neural network, pooling, fully connected layers. Case study: Architecture of Lenet, Alexnet and VGG 16
Module IV (8 Hours)
Recurrent Neural Networks: Back propagation, vanishing gradients, exploding gradients, truncated backpropagation through time, Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) cells, solving the vanishing gradient problem with LSTMs.
Module V (9 Hours)
Autoencoders, variational autoencoders. Generative Adversarial Networks (GAN): Discriminative and generative models, GAN discriminator, GAN generator, upsampling, GAN Training, GAN challenges, loss functions, cross entropy, minimax loss, Wasserstein loss.

Programming assignments using TensorFlow maybe given at the end of each module to get hands on experience.

Textbooks.

1. Generative Deep Learning: David Foster, OReily, (2019)
2. Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT press

(2016)
3. Hands on Machine Learning with Scikit Learn and TensorFlow, Aurélien Géron (2019)
4. Deep Learning Illustrated, Jon Krohn, Grant Beyleveld, Aglae Bassens, Pearson, 1st Edn., (2020)
5. Online book Dive Deep into Machine Learning at https://d2l.ai/

References

Module 1

a. https://www.cse.iitm.ac.in/~miteshk/CS6910/Slides/Lecture2.pdf
b. https://www.cse.iitm.ac.in/~miteshk/CS6910/Slides/Lecture3.pdf

Module 2

a. http://neuralnetworksanddeeplearning.com
b. Hands on Machine Learning with Scikit Learn and TensorFlow, Aurélien Géron
c. Probabilistic Machine Learning: An Introduction, Kevin Murphy
d. https://www.researchgate.net/publication/228339739_Viualizing_data_using_t-SNE

Module 3

a. https://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture11.pdf
b. Convolutional neural networks for visual computing (Chapter 4), Ragav Venkatesan and Baoxin Li CRC press

Module 4

a. On the difficulty of training RNNs: https://arxiv.org/pdf/1211.5063.pdf
b. LSTM: A Search Space Odyssey: https://arxiv.org/abs/1503.04069
c. Understanding Deriving and Extending the LSTM: https://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html
d. Understanding LSTM Networks: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
e. https://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture14.pdf
f. https://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture15.pdf

Module 5

a. GANs in Action: Deep Learning with Generative Adversarial Network Jakub Langgr, Vladimir Bok
b. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play David Foster
c. https://developers.google.com/machine-learning/gan

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Module 1**	**8 Hours**
1.1	Review of Neural Networks: Model of a biological neuron	1
1.2	McCulloch Pitts Neuron, Activation functions	1
1.3	Perceptron, Perceptron Learning Algorithm	1
1.4	Convergence, Multilayer Perceptron	1
1.5	Back propagation	1
1.6	Learning XOR, Sigmoid Neurons	1
1.7	Gradient Descent, Feed forward Neural Networks	2
2	**Module 2**	**10 Hours**
2.1	Training Neural Networks	1
2.2	Initialization, Dropout	1
2.3	Batch normalization and drop out	1
2.4	Over fitting, under fitting, training and validation curves, data visualization, feature and weight visualization, tSNE	2
2.5	Introduction to TensorFlow, graphs, nodes, Tensor Data Structures - rank, shape, type	2
2.6	Building neural networks with tensor flow	2
2.7	Introduction to Keras	1
3	**Module 3**	**10 Hours**
3.1	Convolutional neural networks	1
3.2	Convolution operation	2
3.3	Back propagation in multilayer neural networks	1
3.4	Convolutional layers in neural network, pooling	2
3.5	Fully connected layers	2
3.6	Case study: Architecture of Lenet, Alexnet and VGG 16	2
4	**Module 4**	**8 Hours**
4.1	Recurrent neural networks	1

4.2	Back propagation: vanishing gradients, exploding gradients	1
4.3	Truncated Backpropagation Through Time	1
4.4	LSTM	1
4.5	Gated Recurrent Units (GRUs)	1
4.6	Long Short-Term Memory (LSTM) Cells	1
4.7	Solving the vanishing gradient problem with LSTMs	2
5	**Module 5**	**9 Hours**
5.1	Autoencoders, Variational autoencoders	2
5.2	Generative Adversarial Networks (GAN)	1
5.3	Discriminative and generative models	2
5.4	GAN Discriminator, GAN Generator, upsampling,	1
5.5	GAN Training	1
5.6	GAN challenges, Loss functions, cross entropy, minimax loss, Wasserstein loss	2

24SJMCA241	DATA SCIENCE LAB	CATEGORY	L	T	P	CREDIT
		LAB	0	1	3	2

Preamble:	This is an introductory practical course on Data Science and student will learn how to use various scientific libraries in python to implement data mining techniques and machine learning algorithms.
Prerequisite:	Fundamentals of programming, python programming fundamentals, Machine learning, fundamentals of web programming

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Use different python packages to perform numerical calculations, statistical computations and data visualization	Apply (K3)
CO2	Use different packages and frameworks to implement regression and classification algorithms.	Apply (K3)
CO3	Use different packages and frameworks to implement text classification using SVM and clustering using k-means	Apply (K3)
CO4	Use popular deep learning frameworks to develop and train neural network models and evaluate the performance of different models	Apply (K3)
CO5	Implement programs for web data mining and natural language processing using NLTK	Apply (K3)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	-	-	-	3
CO2	3	3	2	3	-	-	-	3
CO3	3	3	2	3	-	-	-	3
CO4	3	3	2	3	-	-	-	3
CO5	3	3	2	3	-	-	-	3

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	7.5
Maintenance of daily lab record and GitHub management	10
Regular class viva voce	7.5
Timely completion of day-to-day tasks	10
Tests/Evaluation	15

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output	20%	

Course Level Assessment Questions

Course Outcome 1 (CO1)

- Review of python programming – Programs review the fundamentals of python (simple python programs ice breaker) – (at most one lab session)
- Matrix operations (using vectorization) and transformation using python and SVD using Python.
- Programs using matplotlib / plotly / bokeh / seaborn for data visualisation.
- Programs to handle data using pandas

Course Outcome 2 (CO2)

- Program to implement k-NN classification using any standard dataset available in the
  public domain and find the accuracy of the algorithm.

- Program to implement Naïve Bayes Algorithm using any standard dataset available in
  the public domain and find the accuracy of the algorithm.

- Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.

Course Outcome 3 (CO3):

- Program to implement text classification using Support vector machine.

- Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm.

- Program to implement k-means clustering technique using any standard dataset available in the public domain

Course Outcome 4 (CO4):

- Program to implement multilevel perceptron for classification (Iris Dataset).

- Program to implement Feedforward Neural Network for Binary Classification using activation functions like Sigmoid and ReLU.

Course Outcome 5 (CO5)

Web Data Mining:

- Implement a simple web crawler (ensure ethical conduct).
- Implement a program to scrap the web page of any popular website – suggested python package is scrappy (ensure ethical conduct).

Natural Language Processing:

Problems may be designed for the following topics so that students can get hands on experience in using python for natural language processing:

- Part of Speech tagging
- N-gram and smoothening
- Chunking

Syllabus
Review of python programming, Matrix operations, Data Visualisation using matplotlib /plotly / bokeh / seaborn, Data handling using pandas, Classification k-NN algorithm, Naïve Bayes algorithm, Implementation of linear and multiple regression techniques, Text classification using Support vector machine, Implementation of Decision Trees, Clustering using k-means algorithm, Convolutional Neural Network to classify images using Keras framework, Web Crawler and Scrapping web pages, Implementation of NLP - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK.

Reference Books

1. Christopher M Bishop, “Pattern Learning and Machine Learning”, Springer, 2006
2. E. Alpayidin, “Introduction to Machine Learning”, Prentice Hall of India (2005)
3. T. Hastie, RT Ibrashiran and J. Friedman, “The Elements of Statistical Learning”, Springer 2001
4. Toby Segaran, “Programming Collective Intelligence: Building Smart Web 2.0 Applications”, O' Reilly Media; 1 edition (16 August 2007).
5. Drew Conway, John Myles White, “Machine Learning for Hackers: Case Studies and
6. Algorithms to Get You Started”, O' Reilly Media; 1 edition (13 February 2012)
7. Simon Rogers, Mark Girolami, “A First course in Machine Learning”, CRC Press, First Indian reprint, 2015.
8. Tom Mitchell, “Machine Learning”, McGraw Hill, 1997.
9. Bing Liu, Web Data Mining - Exploring Hyperlinks, Contents and Usage Data Second edition, Springer 2011

Course Contents and Lab Schedule:

Sl. No	Topic	No: of Hours
1	Review of python programming, Matrix operations, Programs using matplotlib / plotly / bokeh / seaborn for data visualisation and programs to handle data using pandas.	8 hrs
2	Program to implement k-NN classification using any standard dataset available in the public domain and find the accuracy of the algorithm	2 hrs
3	Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm	4 hrs
4	Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.	4 hrs
5	Program to implement text classification using Support vector machine.	4 hrs
6	Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm	4 hrs
7	Program to implement k-means clustering technique using any standard dataset available in the public domain	4 hrs
8	Program on multilayer and feed forward neural network to classify data using any data set available in public domain	6 hrs
9	Program to implement a simple web crawler and scrapping web pages.	6 hrs
10	Implement problems on natural language processing - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK	4 hrs

24SJMCA243	MOBILE APPLICATION DEVELOPMENT LAB	CATEGORY	L	T	P	CREDIT
		LAB	0	1	3	2

Preamble:	This is a practical course on Mobile Application Development and student will learn how to program in Android Platform and develop applications using SQLite that run on Android Operating System.
Prerequisite:	Basic knowledge on programming and database concepts.

Course Outcomes: After the completion of the course the student will be able to		**K Level**
CO1	Design and develop user interfaces for mobile apps using basic building blocks, UI components and application structure using Emulator	K3
CO2	Write simple programs and develop small applications using the concepts of UI design, layouts and preferences	K3
CO3	Develop applications with multiple activities using intents, array adapter, exceptions and options menu.	K3
CO4	Implement activities with dialogs, spinner, fragments and navigation drawer by applying themes	K3
CO5	Develop mobile applications using SQLite.	K3

Mapping of course outcomes with program outcomes								
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	3	3	2	0	0	2
CO2	3	3	3	3	2	0	0	2
CO3	3	3	3	3	2	0	0	2
CO4	3	3	3	3	2	0	0	2
CO5	3	3	3	3	2	2	0	2

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	7½
Maintenance of daily lab record and GitHub management	10
Regular class viva voce	7½
Timely completion of day-to-day tasks	10
Tests/Evaluation	15

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output	20%	

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Design a Login Form with username and password using Linear Layout and toast valid credentials
2. Write a program that demonstrates Activity Lifecycle.
3. Implementing basic arithmetic operations of a simple calculator
4. Implement validations on various UI controls

Course Outcome 2 (CO2)

1. Design a registration activity and store registration details in local memory of phone using Intents and SharedPreferences
2. Design a simple Calculator using GridLayout and Cascaded LinearLayout
3. Create a Facebook page using RelativeLayout; set properties using .xml file
4. Develop an application that toggles image using FrameLayout

Course Outcome 3(CO3):

1. Implement Adapters and perform exception handling
2. Implement Intent to navigate between multiple activities

3. Develop application that works with explicit intents
4. Implement Options Menu to navigate to activities
5. Develop an application that uses ArrayAdapter with ListView.

Course Outcome 4 (CO4):

1. Develop an application that use GridView with images and display Alert box on selection
2. Develop an application that implements Spinner component and perform event handling
3. Apply themes via code and manifest file
4. Develop application using Fragments
5. Implement Navigation drawer

Course Outcome 5 (CO5):

1. Create database using SQLite and perform INSERT and SELECT
2. Perform UPDATE and DELETE on SQLite database
3. Develop an application as a micro project which uses SQLite database as an assignment

SYLLABUS

Fundamentals: Basic Building blocks – Activities, Services, Broadcast Receivers and Content providers, UI Components – Views and notifications Components for communication -Intents and Intent Filters
Application Structure: AndroidManifest.xml, user-permission – sdk, Resources and R.java, Assets, Layouts and Drawable Resources, Activities and Activity lifecycle.
Emulator-Android Virtual Device: Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS
Basic UI design: Form widgets, Text Fields, Validation of EditText, Layouts, [dip, dp, sip, sp] versus px
Preferences: Shared Preferences, Preferences from xml
Menu: Option menu, Context menu, menu from xml, menu via code
Intents: Explicit Intents, Implicit intents
UI design: Time and Date, Images and media, Android Adapter and ListView, Composite, Alert Dialogs and Toast, Popup, Fragments, Navigation drawer
Tabs, Tab Activity Styles & Themes: styles.xml, drawable resources for shapes, gradients (selectors), style attribute in layout file, Applying themes via code and manifest file
Content Providers: SQLite Programming, SQLite Open Helper, SQLite Database, Cursor, Reading and updating Contacts, Reading bookmarks

Reference Books

1. Joseph Annuzzi Jr, Lauren Darcey, Shane Condor, "Advanced Android Application Development, Developers Library", Pearson Education, 4th Edition (2015)
2. Lauren Darcey, Shane Condor, "Android, Wireless Application Development", Pearson Education, 3rd Edition.
3. Paul Deitel, Harvey Deitel, Alexander Wald, "Android 6 for programmers, An AppDriven Approach", Pearson Education
4. Rap Payne, "Beginning App Development with Flutter: Create Cross-Platform Mobile Apps", Apress (2019)

Course Contents and Lecture Schedule

Sl No	Topic	No. of hours
1	Fundamentals – Basic building blocks	3
2	Application structure, layout and resources	3
3	Android Virtual Device, Activity Lifecycle	3
4	Basic UI Design and EditText Validation	4
5	Shared Preferences, RelativeLayout, FrameLayout, GridLayout and Preferences from xml	9
6	ArrayAdapter, ListView and Exception handling	3
7	Various Menu options	3
8	Explicit and Implicit Intents	3
9	Images and media, Dialogs, Spinner component, Popups, Fragments, Navigation drawer	6
10	Applying themes and styles .xml	3
11	SQLite Programming	6

24SJMCA245	MINI PROJECT	CATEGORY	L	T	P	CREDIT
		PROJECT	0	0	4	2

Preamble:	This project work aims to enable the students to apply the software engineering principles on a real software project, to make the students familiar with the stages of a deployment pipeline and to develop a software product using the latest software development methodology.
Prerequisite:	Knowledge in software engineering principles and programming skills.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Identify a real-life project which is useful to society / industry	K2
CO2	Interact with people to identify the project requirements	K2
CO3	Apply suitable development methodology for the development of the product / project	K3
CO4	Analyse and design a software product / project	K3
CO5	Test the modules at various stages of project development	K3
CO6	Build and integrate different software modules	K3
CO7	Document and deploy the product / project	K3

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3					2
CO2		3	2		2			2
CO3	3	2	3	2	2	2		
CO4	3	2	3	2				
CO5	2		3	2	2			
CO6	3		3	2	2	2		
CO7	2		3	2	2	2	2	3

Mark distribution

Total Marks	CIE	ESE
100	100	-

Marks Division:

Continuous evaluation by Supervisor, Scrum Master and Project Guide	50 Marks
Interim evaluation by the Project Assessment Board	25 Marks
Final evaluation by the Project Assessment Board	25 Marks
Total	**100 Marks**

Guidelines:

- Students shall identify Real-Life Projects which are relevant and useful to the society or industry.
- The project shall be an individual project and must be done in-house. The student has to spend time in the lab for the project work.
- Attendance as per MCA regulations is applicable for submitting the project for final evaluation.
- Students shall submit project synopsis and get prior approval from the Project (Faculty) Supervisor before the project work begins.
- If there is a customer for the project then he/she will be the Product Owner (External Guide) and a faculty from the department will be the Internal Guide. If there is no such customer then the Internal Guide himself/herself shall act as the Product Owner.
- A faculty / technical staff shall act as the Scrum Master to continuously monitor the project development. Periodic meetings, of less than 15 minutes, at the convenience of the Scrum Master are to be highly encouraged. Ensure such meetings occur once in three days.
- Set a sprint as two weeks, ensure biweekly reviews. A review shall not exceed 30 minutes. A demo to the Product Owner (Project Guide) is mandatory in every review.
- The student shall maintain a Scrum Book (Rough Record) which has to be divided into 4 parts – (i) Product Backlog (ii) Database & UI Design (iii) Testing & Validation and (iv) Details of Versions. Make dated entries in the corresponding part at regular intervals. The corrections and comments from Product Owner and Scrum Master should be clearly indicated with the Date.
- Test Driven Development methodology may be practiced for the project development. BugZilla, BackLog or any such tool may be used for Bug Tracking
- Git shall be used for Version Control and Git commit history may be verified as part of project evaluation
- LaTeX or an equivalent tool shall be used for preparing Presentations and Project Report.

- Interim evaluations of project's progress shall be conducted as part of Internal Assessment. Project Evaluation Board may consist of Project Supervisor, Product Owner, Scrum Master and one other Faculty Member from the department. Scrum reviews shall not be sacrificed for such presentations.
- At the end of the semester entire project development activities shall be evaluated internally by the Project Evaluation Board.

Week	Schedule
1	Familiarisation with build tools (editor/IDE, compiler such as gcc with commonly used options/switches, debugger like dgb). Familiarisation with an IDE (Eclipse, NetBeans…), that supports build tools and common version control operations using Git . Familiarisation with Docker Selection of Topic, Formation of Development Team, Feasibility analysis.
2	Topic Approval, Meeting of Development Team including Scrum Master with Product Owner. Informal, preliminary discussions of requirements. Creating user stories in the rough record.
	Commencement of the Project.
3	Identifying modules, Initial Design of Database & UI. Creating a Docker container for the environment Creating an empty git repository by Scrum Master / one member of the Development team and setting permission to other members. Pushing the first version of the Project along with a Readme file containing contact details of team members. Creating pull requests for sample update of Readme by each member and merging the pull requests of one by another.
4-5	Setting up systems for development, testing and production. Design of the basic model of a simple deployment pipeline Creating a suitable folder structure (Maven's folder structure is desirable). Creating Unit tests using an XUnit framework, Writing the build and code analysis script, Writing acceptance test scripts and test cases, Setting up a Continuous Integration System like Jenkins. Automating acceptance tests with Selenium, Karate or an equivalent tool, writing a simple deployment script that uses scp/rsync or Ansible for copying the Dockerfile and running Docker with

	ssh. First Scrum Review. (Here onwards, the Scrum reviews are conducted on every other week)
7	Project Presentation - Interim Evaluation to be based on Git History
14	Submission of Project Report, with Scrum Book Project Presentation – Final Evaluation to be based on Git History, Scrum Book, Project Report and Presentation

References

1. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) 1st Edition
2. Alistair Cockburn, Agile Software Development: The Cooperative Game, Addison Wesley, 2nd Edition (2006).
3. Andrew Hunt, David Thomas, The Pragmatic Programmer: From Journeyman to Master, Pearson India, 1st Edition (2008).
4. Ken Schwaber, Mike Beedle, Agile Software Development with Scrum, Pearson (2008).
5. Lisa Crispin, Janet Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison Wesley Professional, 1st Edition (2008).
6. Mike Cohn, User Stories Applied: For Agile Software Development, Addison Wesley, 1st Edition, (2004).
7. Pressman, R.S., Software Engineering: A Practitioner's Approach, McGraw Hill SE, 7th Edition, (2010).
8. Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, Prentice Hall Imprint, Pearson Education, 2nd Edition (2002).\
9. Rod Stephens, Beginning Software Engineering, Wrox Series, Wiley India Pvt Ltd (2015).

Web Reference

1. Introduction to DevOps (https://www.edx.org/course/introduction-devops-microsoft- dev212

ST. JOSEPH'S
COLLEGE OF ENGINEERING
AND TECHNOLOGY,
- PALAI -
AUTONOMOUS

MCA TWO-YEAR

SEMESTER 4

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
24SJMCA242	COMPREHENSIVE VIVA	VIVA	-	-	-	6

Preamble: Comprehensive Viva intends to assess the knowledge gained by a student in the core courses of this programme and to make the student aware of his/her knowledge level and where he/she stands after completing this programme. This course will help the student in preparing for comprehensive examinations and improve the confidence in answering questions in objective mode.

Prerequisite: Thorough knowledge in all the courses he/she learned during this programme.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	K Level
CO 1	Articulate the concepts in the core courses learned through this programme.	K2
CO 2	Attend technical interviews with confidence.	K2
CO 3	Interpret questions and answer them with clarity.	K2
CO 4	Make use of the concepts learned through this programme in future.	K3

Mapping of course outcomes with program outcomes

Mapping of course outcomes with program outcomes								
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2		2	2	2	2	3
CO2	3	2	1	2	3		3	3
CO3	3	2			2		2	2
CO4	3	2	2	2		2	3	3

Mark distribution

Total Marks	CIE	ESE
100	-	100

Guidelines:

- Viva shall be conducted by a panel of examiners consisting of:
  - 1. Head of the department
  - 2. A senior faculty in the department.
  - 3. External examiner appointed by the competent authority.
- Viva shall be conducted for each student for a minimum of 20 minutes.
- Knowledge level of the student shall be assessed on the following topics.
  - 24SJMCA105 - Advanced Data Structures
  - 24SJMCA107 - Advanced Software Engineering
  - 24SJMCA102 - Advanced Database Management Systems
  - 24SJMCA201 - Data Science & Machine Learning
  - 24SJMCA203 - Design & Analysis of Algorithms
  - Code snippets in Java or Python to solve simple problems.
  - Technologies used in the project work.
  - Recent developments in the field of computer science.

24SJMCA244	SEMINAR	CATEGORY	L	T	P	CREDIT
		SEMINAR	-	-	2	2

Preamble:	This course intends to enable the students to gain knowledge in any of the technically relevant current topics on Computer Science or Information Technology, and to acquire confidence in presenting the topic and preparing a report
Prerequisite:	Nil

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Annotate the ideas presented in technical papers	Level 2: Understand
CO2	Comprehend a concept by referring different technical documents	Level 2: Understand
CO3	Prepare technical documents	Level 3: Apply
CO4	Present a topic before an audience	Level 3: Apply
CO5	Interact with the audience	Level 2: Understand

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	1	1		2
CO2	3	3	2	2	1	1		2
CO3	3	3	2	2	2	2	1	3
CO4	2	2	2	3	3	2	1	3
CO5	1	2	1	2	3	2	1	3

Mark Distribution

Total Mark	: 50 Marks
Continuous Evaluation Marks (*By Guide*)	: 20 Marks
Evaluation by Faculty Committee	: 30 Marks

Assessment Criteria

Scope and relevance of topic	: 20%
Quality of Presentation slides	: 10%
Presentation Skills	: 30%
Knowledge in the topic	: 20%
Report	: 20%

Guidelines

- Students shall conduct detailed study on a technically relevant current topic in Computer Science / Information Technology under the supervision of a Faculty Guide and present it as a seminar at the end of the study.
- The study may be conducted on
  - ▪ articles published in reputed journals/conference proceedings
  - ▪ recent development in Computer Science / Information Technology
  - ▪ recent research and development activity in a research lab
  - ▪ latest software tool or framework
- Students shall submit an abstract on an identified topic and get prior approval from the Faculty Guide before the study begins.
- The student shall submit a seminar report, based on the study and their findings. The report shall not be a reproduction of the original paper or manual.
- The study and its findings shall be presented in the class taking a duration of 15-20 minutes.
- LaTeX or an equivalent tool shall be used for preparing Presentations and Seminar Report.

- Students shall be encouraged to publish their study in journals and due credit shall be given to such students.
- A committee of three senior faculty members shall be constituted by the head of the department and the seminar presentation shall be evaluated by that committee.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
24SJMCA246	MAIN PROJECT	PROJECT	-	-	27	12

Preamble: This project work aims to enable the students to apply the software engineering principles on a real software project, to make the students familiar with the stages of a deployment pipeline and to develop a software product using the latest software development methodology.

Prerequisite: Knowledge in software engineering principles and programming skills.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	K Level
CO 1	Identify a real-life project which is useful to society / industry	K2
CO 2	Interact with people to identify the project requirements	K3
CO 3	Apply suitable development methodology for the development of the product / project	K3
CO 4	Analyse and design a software product / project	K4
CO 5	Test the modules at various stages of project development	K5
CO 6	Build and integrate different software modules	K5
CO 7	Document and deploy the product / project	K3

Mapping of course outcomes with program outcomes

Mapping of course outcomes with program outcomes								
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	1	2	2	2	2
CO2	3	2	2	3	2	2	2	2
CO3	3	2	2	3	2	1	1	2
CO4	3	2	2	3	2	1	2	3
CO5	3	2	1	3	1	2	1	3
CO6	3	2	2	3	2	2	1	2
CO7	3	2	2	3	1	1	1	3

Mark distribution

Total Marks	CIE	ESE
100	70	30

Marks Division

Continuous evaluation by Supervisor, Guide(s) and Scrum Master	30 Marks (Internal)
Evaluation by the Project Assessment Board	40 Marks (Internal)
Evaluation by the External expert	30 Marks (External)
Total	**100 Marks**

Guidelines:

- Students shall identify Real-Life Projects which are relevant and useful to the society or industry.
- The project shall be an individual project and must be done in-house. The student has to spend time in the lab for the project work. Attendance as per MCA regulations is applicable for submitting the project for final evaluation.
- However, in exceptional cases students shall be given permission to work on the project outside the campus and at the industry premises if the organization offering the project belongs to anyone of the following categories.
    - CMM Level 5 Certified Company
    - Publicly listed company in India
    - National Research Institute
    - Central / State Government Department
    - Project funded by the Central / State Government Agency
    - Companies approved by the Department Internship Scrutiny Committee, constituted by the Head of the Department.
- In such cases, the student is required to produce a letter from the organization before starting the project and an internship scrutiny committee constituted by the head of the department shall make the decision on permission. Industries and training institutes that offer project work for a fee shall not be permitted.
- Students shall submit project synopsis and get prior approval from the Project (Faculty) Supervisor before the project work begins.

- If there is a customer for the project then he/she will be the Product Owner (External Guide) and a faculty from the department will be the Internal Guide. If there is no such customer then the Internal Guide himself/herself shall act as the Product Owner.

- A faculty / technical staff shall act as the Scrum Master to continuously monitor the project development. Periodic meetings, of less than 15 minutes, at the convenience of the Scrum Master are to be highly encouraged. Ensure such meetings occur once in three days.

- The student shall maintain a Scrum Book (Rough Record) which has to be divided into 4 parts – (i) Product Backlog (ii) Database & UI Design (iii) Testing & Validation and (iv) Details of Versions. Make dated entries in the corresponding part at regular intervals. The corrections and comments from Product Owner and Scrum Master should be clearly indicated with the Date.

- Test Driven Development methodology may be practiced for the project development. BugZilla, BackLog or any such tool may be used for Bug Tracking.

- Git shall be used for Version Control and Git commit history may be verified as part of project evaluation .

- LaTeX or an equivalent tool shall be used for preparing Presentations and Project Report.

- Students shall be encouraged to publish their work in journals and due credit shall be given to such students.

- For the externally done projects, periodic confidential progress report and attendance statement shall be collected from the External Guide and be reviewed by the Project Supervisor.

- Set a sprint as two weeks, ensure biweekly reviews. A review shall not exceed 30 minutes. A demo to the Product Owner (Project Guide) is mandatory in every review.

- Interim evaluations of the project's progress shall be conducted by a Project Assessment Board as part of internal assessment. Two such evaluations are desirable. Scrum reviews shall not be sacrificed for such presentations.

- The Project Assessment Board shall be constituted by the Head of the Department with the following five members.
  Chairman:

  1. Head of the Department

  Members:

  2. Project supervisor/s of the student

3. One faculty member from the Department
4. One faculty member from a sister Department
5. An external expert, either from an academic/research institute or Industry. (For the externally done projects, the external guide shall be invited as external expert.)

- At the end of the semester, two evaluations shall be there on the entire project development activities. First an internal evaluation by the Project Assessment Board and second an external evaluation by an External Examiner.

- An External Examiner either from an academic institute or industry shall be appointed by the Authority concerned for the External Evaluation.

Week	Schedule
(May be scheduled inline with the College academic calendar)	
1	Selection of Topic, Submission of project synopsis and getting approval Meeting of Development Team including Scrum Master with Product Owner (Project Guide)
2	Commencement of the Project.
4	First Sprint release and Scrum Review by the Product Owner (Project Guide)
6	Second Sprint release and Scrum Review by the Project Guide First interim evaluation by the Project Assessment Board
8	Third Sprint release and Scrum Review by the Project Guide
10	Fourth Sprint release and Scrum Review by the Project Guide
11	Second interim evaluation by the Project Assessment Board
12	Fifth Sprint release and Scrum Review by the Project Guide
13	Submission of project report, with Scrum Book Final project presentation Evaluation by the Project Assessment Board
14	Final evaluation by the External Examiner.