Shri Vile Parle Kelavani Mandal's

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Scheme and Detailed syllabus (DJS22)

Third Year B.Tech

in

Computer Science and Engineering (Data Science)

(Semester VI)

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Scheme of Semester - VI for Department of Computer Science and Engineering (Data Science)
Academic Year - 2024 - 2025

Sr	Course Code	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment (B)					Aggregate (A+B)	Credits	
			Theory (hrs.)	Practical (hrs.)	Tutorial (hrs.)	Credits	Duration	Theory	Oral	Pract	Oral & Pract	End Sem Exam Total	Term Test 1 (TT1)	Term Test 2 (TT2)	Total (TT1 & TT2)	Term Work Total	CA Total			
1	DJS22DSC601	Machine Learning - III (Reinforcement Learning)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	**4**
	DJS22DSL601	Machine Learning - III Laboratory	--	2	--	1	--	--	--	--	--	--	--	--	--	25	25	25	1	
2	DJS22DSC602	Natural Language Text Processing	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	**4**
	DJS22DSL602	Natural Language Text Processing Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	25	25	50	1	
3	DJS22DSC603	Image Processing and Computer Vision - II	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	**4**
	DJS22DSL603	Image Processing and Computer Vision - II Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	25	25	50	1	
4	DJS22DSL604	Programming Laboratory	2	2	--	3	2	--	--	--	50	50	--	--	--	50	50	100	3	**3**
5@	DJS22DSC6011	Cloud Computing	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	**4**
	DJS22DSL6011	Cloud Computing Laboratory	--	2	--	1	--	--	--	--	--	--	--	--	--	25	25	25	1	
	DJS22DSC6012	Recommender System	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22DSL6012	Recommender System Laboratory	--	2	--	1	--	--	--	--	--	--	--	--	--	25	25	25	1	
	DJS22DSC6013	Embedded System & RTOS	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22DSL6013	Embedded System & RTOS Laboratory	--	2	--	1	--	--	--	--	--	--	--	--	--	25	25	25	1	
	DJS22DSC6014	Computational Neuroscience	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22DSL6014	Computational Neuroscience Laboratory	--	2	--	1	--	--	--	--	--	--	--	--	--	25	25	25	1	
6	DJS22ILLL2	Innovative Product Development IV	--	2	--	1	2	--	--	--	25	25	--	--	--	25	25	50	1	**1**
		Total	**14**	**12**	0	**20**	**16**	**260**	**50**	0	**75**	**385**	**80**	**60**	**140**	**175**	**315**	**700**	**20**	**20**

@ Any 1 Elective Course

Prepared by Checked by Head of the Department Vice Principal Principal

Continuous Assessment (A):

Course	Assessment Tools	Marks	Time (hrs.)
Theory	a. One Term test (based on 40 % syllabus)	20	1
	b. Second Term test (next 40 % syllabus) / presentation / assignment / course project / group discussion / any other.	15	1
	Average marks of a and b	35	--
Audit course	Performance in the assignments / qui / power point presentation / poster presentation / group project / any other tool.	--	As applicable
Laboratory	Performance in the laboratory and documentation.	25	
Tutorial	Performance in each tutorial & / assignment.	25	
Laboratory &Tutorial	Performance in the laboratory and tutorial.	50	

The final certification and acceptance of term work will be subject to satisfactory performance upon fulfilling minimum passing criteria in the term work / completion of audit course.

Semester End Assessment (B):

Course	Assessment Tools	Marks	Time (hrs.)
Theory / * Computer based	Written paper based on the entire syllabus.	65	2
	* Computer based assessment in the college premises.		
Oral	Questions based on the entire syllabus.	25	as applicable
Practical	Performance of the practical assigned during the examination and the output / results obtained.	25	2
Oral & Practical	Project based courses - Performance of the practical assigned during the examination and the output / results obtained. Based on the practical performed during the examination and on the entire syllabus.	As per the scheme	2

Prepared by Checked by Head of the Department Principal

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Machine Learning – III (Reinforcement Learning) (DJS22DSC601)

Machine Learning – III Laboratory (DJS22DSL601)

Pre-requisite: Machine Learning-I, Machine Learning-II and Artificial Intelligence.

Course Objectives: To make students learn to build programs that act in a stochastic environment, based on past experience using various Reinforcement Learning methods.

Outcomes: Students will be able to

1. Explain basic and advanced Reinforcement Learning techniques.
2. Identify suitable learning tasks to which Reinforcement learning and Deep Reinforcement Learning techniques can be applied.
3. Apply appropriate Reinforcement Learning method to solve a given problem.

Machine Learning – III (Reinforcement Learning) (DJS22DSC601)		
Unit	**Description**	**Duration**
1	**Introduction:** Reinforcement Learning (RL), Elements of Reinforcement Learning, Reinforcement Learning Vs Supervised Learning, Approaches of solving Reinforcement Learning: Value based, policy based, model based, Exploration - Exploitation dilemma, Evolutionary methods, Immediate Reinforcement Learning.	**04**
2	**Immediate Reinforcement Learning:** **Bandit Problems:** Bandit problems, Value-action based methods (sample average), Greedy method, €-greedy method, Incremental Implementation, Non-stationary problem, Optimistic Initial values, UCB algorithm, Thompson Sampling. **Policy Gradient Approaches**: Linear reward Penalty Algorithm, Parameterized policy representation(Θ), Evaluation of policy(η(Θ)), REINFORCE algorithm.	**06**
3	**Full Reinforcement Learning:** Difference between Immediate and Full Reinforcement Learning, Agents and Environment, Goals, Rewards, Returns, Policy in Full Reinforcement Learning, Episodic and Continuing Tasks. **Markov Decision Process (MDP):** Markov Property, Finite Markov Decision Process, Value functions, Bellman's equations, optimal value functions, Definition of MDP in Reinforcement Learning, Solution of the Recycling Robot problem	**08**
4	**Dynamic Programing:** Policy evaluation, policy improvement, policy iteration, value iteration, Asynchronous Dynamic Programing, bootstrap, full back up. **Monte Carlo Method:** Advantages of Monte Carlo over Dynamic Programing, Monte Carlo Control, on-policy, off- policy, Incremental Monte Carlo, Issues/Assumptions in Monte Carlo Methods, Solution of BlackJack using Monte Carlo Method.	**08**

5	**Temporal Difference Learning:** What is Temporal Difference learning, Advantages of Temporal Difference methods over Monte Carlo and Dynamic Programming methods, TD (0), On-policy vs off-policy, SARSA, Q learning. **Eligibility traces:** N-step Temporal Difference methods, On-line vs Off-line updation, TD(**λ**): forward view, backward view, Traces: Accumulating trace, Dutch trace, Replacing trace, Equivalence of forward and backward view, SARSA(**λ**).	**06**
6	**Deep Reinforcement Learning:** **Function Approximation:** Drawbacks of tabular implementation, Function Approximation, Gradient Descent Methods, Linear parameterization, Policy gradient with function approximation. **Deep Reinforcement Learning:** Intro of Deep Learning in Reinforcement Learning, Deep learning training workflow, Categories of Deep learning, Deep Q-Network, Ways of improving Deep Q-Network, REINFORCE in Full Reinforcement Learning, Actor-Critic Algorithm, A2C, A3C, DDPG.	**07**
	Total	**39**

Suggested List of Experiments:

Machine Learning – III Laboratory (DJS22DSL601)	
Sr. No.	**Title of the Experiments**
1	Bandit Problem: • Implement Greedy and Epsilon greedy methods. • Comparison between Greedy and Epsilon Greedy Policy • UCB: Upper Confidence Bound
2	Policy Gradient (Convergence) • Implement REINFORCE algorithm on a CartPole/ Lunar Lander.
3	Dynamic Programming and Monte Carlo Methods • Implementation of GridWorld using Dynamic Programming • Jack's Car Rental using Dynamic Programming • Gamblers Problem using Dynamic Programming • BlackJack using Monte Carlo • Race Track Problem
4	Temporal Difference • Implement Frozen Lake using SARSA • Implement Grid world using Q learning
5	Deep Reinforcement Learning • Compare the performance of Reinforcement Learning and Deep Reinforcement Learning on a Cart pole problem. • Implementation of Deep Q-Network algorithm • Actor Critic: Find the optimal policy using the Actor Critic method.

Minimum 10 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (15m) and Assignment/Quizzes (10m).

Books Recommended:

Textbooks

1. Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT Press, 2nd Edition, 2018.
2. Laura Graesser Wah Loon Keng, "Foundations of Deep Reinforcement Learning," Pearson Education, 1st Edition, 2020.

Reference Books

1. Phil Winder, "Reinforcement Learning Industrial Applications of Intelligent Agents", O'Reilly, 1st Edition, 2020.
2. Csaba Szepesvari, "Algorithms for Reinforcement Learning," Morgan & Claypool Publishers, 1st Edition, 2019.
3. Enes Bilgin, "Mastering Reinforcement Learning with Python", Packt publication, 1st Edition, 2020.
4. Brandon Brown, Alexander Zai, "Deep Reinforcement Learning in Action", Manning
5. Publications, 1st Edition, 2020.
6. Micheal Lanham, "Hands-On Reinforcement Learning for Games," Packt Publishing, 1st Edition, 2020
7. Abhishek Nandy, Manisha Biswas, "Reinforcement Learning: With Open AI, TensorFlow and Keras using Python," Apress, 1st Edition, 2018.

Weblinks:

1. NPTEL Course in Reinforcement Learning: https://onlinecourses.nptel.ac.in/noc22_cs75/preview
2. Reinforcement Learning Course (Stanford University): https://www.youtube.com/watch?v=FgzM3zpZ55o
3. AI Games with Deep Reinforcement Learning: https://towardsdatascience.com/how-to-teach-an-ai- to-play-games-deep-reinforcement-learning-28f9b920440a
4. Deep Reinforcement Learning: https://www.v7labs.com/blog/deep-reinforcement-learning-guide

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Natural Language Text Processing (DJS22DSC602)

Natural Language Text Processing Laboratory (DJS22DSL602)

Pre-requisite: Machine Learning-I, Machine Learning-II, Foundations of Data Analysis, Statistics for Data Science

Course Objectives:

To introduce basics of language computation fundamental through morphological computation, syntax, semantic and discourse analysis. Apply these concepts to develop Computational Models for Real World Applications

Outcomes: Students will be able to

1. Understand the pre-processing required for linguistic data types.
2. Apply appropriate pre-processing technique on linguistic data.
3. Analyze different Machine Learning and deep learning algorithms to develop applications based on natural language processing.

Natural Language Text Processing (DJS22DSC602)		
Unit	**Description**	**Duration**
1	**Introduction:** Generic Natural Language Processing (NLP) system, levels of NLP, Knowledge in language processing, Ambiguity in Natural language, stages in NLP, challenges of NLP, Applications of NLP Machine Translation, Sentiment Analysis etc. **Text Processing:** Word Tokenization and Segmentation, Lemmatization, Bag of words, N-gram language model, N-gram for spelling correction. Edit distance - Dynamic Programming Approach, Weighted Edit Distance, Finding Dictionary Entries with Small Edit Distances, Noisy Channel Model, Non-word errors Real-word errors. Evaluation of Language Models, Basic Smoothing, Advanced Smoothing Models. Advanced: Perplexity's Relation to Entropy.	**05**
2	**Text Classification:** Text classification definition and datasets, Generative text classifiers (naive Bayes) Discriminative text classifiers (logistic regression), Bag-of-words Generative Classifier, BOW Discriminative Model , Multi-class Classification: Softmax, Gradient Descent, Statistical significance testing, Dataset understanding and creation.	**05**
3	**Recurrent Neural Networks:** Recurrent Neural Network ,RNNs as Language, RNNs for Sequence Classification, Stacked Recurrent Neural network, Bidirectional RNNs, Managing Context in RNNs:Long Short Term Memory (LSTMs) and gated Recurrent Unit(GRUs).	**06**
4	**Computational Semantics and Semantic Parsing:** **Vector Semantics**: Words and Vectors, Term Frequency-Inverse Document Frequency (TFIDF), Word2vec, Continuous Bag of Words, ELMO, Vector Visualizing Embedding's, Semantic properties of embedding's, Bias and Embedding's Evaluating Vector Models, Cosine for measuring similarity, Pointwise Mutual Information (PMI), PPMI vector models. **Lexical Semantics:** Word Senses -Relations Between Senses, WordNet: A Database of Lexical Relations, Word Sense Disambiguation Alternate WSD algorithms and Tasks.	**07**

5	**Computational Morphology and Syntax Analysis:** **Computational Morphology:** Morphological Processes, Morphological Analysis- Inflectional morphology & Derivational morphology, Regular expression, Finite State Automata, Finite State Transducer, Morphological parsing with FST, Lexicon free FST Porter stemmer, Two - level Morphology. **Syntax Analysis:** Introduction to POS Tagging, Probabilistic Tagging, Markov Models, Hidden Markov Models (HMM) for POS Tagging, Conditional Random Fields (CRF), Named Entities and Named Entity Tagging, Context-Free Grammars-Derivation, Constituency Parsing, Dependency Parsing.	**12**
6	**Discourse Coherence:** Coherence Relation, Discourse Structure Parsing, Centring and Entity-Based Coherence, Global Coherence.	**04**
	Total	**39**

Suggested List of Experiments:

Natural Language Text Processing Laboratory (DJS22DSL602)	
Sr. No.	**List of Experiment**
1	Perform Pre-processing steps in Natural language Processing (Tokenization, Stop Word detection, Stemming and Lemmatization.
2	Implement Parts of Speech tagging using HMM
3	Implement word-embedding and TF-IDF vectors in Natural language Processing
4	Implement language model using Ngram language model
5	Generate recursive set of sentences using Context Free Grammar Identify the word senses using "synset" in NLTK
6	Implement Spelling Check, Spelling Correction and Auto complete using Language models or CFG.
7	Implement a Spam classifier in Natural Language Processing
8	Implement Fake News Classifier Using LSTM-Deep Learning in NLP
9	Implement a Sentiment Analysis in Natural Language Processing
10	Implement NLP application on Regional Language
11	Implement Question Answering in NLP
12	Implement Chatbot in NLP
13	Implement Information Retrieval for extracting Text from Webpages and Images
14	Mini Project

Minimum 10 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (15m) and Quizzes (10m).

Books Recommended:

Textbooks:

1. Jurafsky and Martin, "Speech and Language Processing", Prentice Hall, 3rd Edition, 2020.
2. Uday Kamath, "Deep Learning for NLP and Speech Recognition", 1st Edition, 2019.

Reference Books:

1. Jelinek, F., "Statistical Methods for Speech Recognition", The MIT Press, 2022.
2. Yuli Vasiliev "Natural Language Processing with Python and spaCy - A Practical Introduction", No Starch Press, 2022.
3. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana, "Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems", O'Reilly, 1st Edition, 2020.

Web Links

1. Virtual Lab: -https://nlp-iiith.vlabs.ac.in/
2. Virtual Lab:-http://vlabs.iitb.ac.in/vlabs dev/vlab_bootcamp/bootcamp/The_Big_Bang_Nerds/index.html
3. Nptel Course: - https://nptel.ac.in/courses/106105158

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Image Processing and Computer Vision – II (DJS22DSC603)

Image Processing and Computer Vision - II Laboratory (DJS22DSL603)

Pre-requisite:

1. Mathematics for Intelligent System
2. Machine Learning -I and II
3. IPCV -I

Objectives:

To introduce theory and computation related to imaging geometry, and scene understanding. Also, to provide exposure to clustering, classification and deep learning techniques applied in computer vision.

Outcomes: On completion of the course, the learner will be able to:

1. Understand various data capturing methods.
2. Apply appropriate object detection and object segmentation methods.
3. Apply suitable method to analyze complex vision data.
4. Develop suitable vision model for prediction.

Image Processing and Computer Vision – II (DJS22DSC603)		
Unit	**Description**	**Duration**
1	**Camera Geometry Model:** Basics of Real Aperture Camera, Lens as LSI System, Geometric Projective, 2D Transformations, 3D Transformations, Homography Computation, planar homography, Camera geometry, Stereo geometry, Linear Filtering, Correlation, Convolution, Hierarchy of Transformations, Rotational Representation, Weak perspective projection and orthographic projection, coordinate system, camera parameters and camera calibration Algorithm.	**06**
2	**Object Detection:** **Two Stage/Proposal**: Convolutional Neural Networks for Detection: R-CNN, Fast R-CNN, Faster R-CNN, RFCN and Mask RCN; Architecture and Issues in each algorithm. visualization of Kernels; Backprop-to-image/Deconvolution Methods; **One Stage/Proposal Free:** YOLO, SSD, evaluation metrics (IoU, AP), Non-max suppression YOLO Loss function, Variants of YOLO.	**08**
3	**Face Recognition and Verification:** Zero-shot, One-shot, Few-shot Learning; Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss; Attention Models in Vision, Spatio-temporal Models, Action/Activity Recognition, Region-based convolutional neural network, Semantic segmentation	**06**
4	**Generative Models:** Types of generative models: Implicit and Explicit density; Generative Adversarial Network; Vanilla GAN, Mode Collapse in GAN, Conditional GAN, DC GAN, GAN objective functions, JSD Divergence, EM Distance Least Squares.	**06**
5	**Object Segmentation:** Semantic segmentation, Scene Parsing, semantic flow, Bilinear Interpolation, Symmetry in Segmentation, Featured image pyramid, pixel-wise softmax, PSPNet, FPN, UNet, clustering method for segmentation, Distance metrics(Euclidean, Cosine, Hamming, Manhattan, Minkowski, Chebyshev, Jaccard, Haversine, Sorensen-Dice), Linkage Types (Single, Average, Complete, Centroid).	**07**

6	**Motion Analysis and action recognition:** Introduction to motion analysis, Horn and Shunck method, Lucas-Kanade algorithm for optical flow, Deep learning in optical flow estimation. Motion models. Introduction to action recognition, Action classification, Action localization. Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation. Visual object tracking methods and its examples, multiple objects tracking methods, Tomasi and Kanade Motion factorization algorithm, Applications of feature point tracking: mosaicing, video stabilization, structure from motion.	**06**
	Total	**39**

Suggested List of Experiments

Image Processing and Computer Vision - II Laboratory (DJS22DSL603)	
Exp.	**Suggested experiments**
1	Object Detection (CNN): Cancer Cells Detection using Medical Image Processing.
2	Object Detection (CNN): Comparative analysis of different CNN models on Image Dataset.
3	Object Detection (YOLO): Identifying vehicle from a Road Traffic CCTV video Footage.
4	Implement transfer learning using the models (CNN, YOLO, etc.).
5	Similarity-Based Image Matching Using Siamese Networks.
6	Fine-Grained Image Recognition for Subcategory Classification.
7	Face Recognition: Facial Key Point Detection, Face verification, Hybrid image formation for identification of facial expression classification and detection.
8	GAN: Converting Black and white image into Colored image.
9	GAN: Deep fake Detection.
10	Image Segmentation: Image Categorization for a given Vision Dataset.
11	Motion Analysis: Spatio-Temporal Analysis for Body Postures.
12	Mini Project

Minimum of 6 experiments from the given list or add new experiments. At least one experiment on each: CNN, YOLO, GAN, U-Net should be implemented.

*The Term Work will be calculated based on Laboratory Performance (15m) and Computer Based Assessment/Mini Project (10m).

Books Recommended:

Text books:

1. Richard Szeliski, "Computer Vision: Algorithms and Applications (2nd Edition)", Springer, 2022.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2019.
3. Benjamin Planche, Eliot Andres, "Hands-On Computer Vision with TensorFlow 2", Packt Publishing, 2019.
4. Gonzalez, Rafael C., and Woods, Richard E., "Digital Image Processing (4th Edition)", Pearson, 2018.
5. Mark Nixon, "Feature Extraction and Image Processing for Computer Vision (3rd Edition)", Academic Press, 2019.
6. Scott Krig, "Computer Vision Metrics: Survey, Taxonomy, and Analysis (2nd Edition)", Apress, 2019.

Reference Books:

1. Adrian Kaehler, Gary Bradski, "Learning OpenCV 4: Computer Vision with Python", O'Reilly Media, 2019.

2. Richard Hartley, Andrew Zisserman, "Multiple View Geometry in Computer Vision (2nd Edition)", Cambridge University Press, 2020.
3. E. R. Davies, "Computer Vision: Principles, Algorithms, Applications, Learning (5th Edition)", Academic Press (Elsevier), 2017.
4. Mohamed Elgendy, "Deep Learning for Vision Systems", Manning Publications, 2020.
5. Yeshwanth Reddy, Kishore Ayyadevara, "Modern Computer Vision with PyTorch", Packt Publishing, 2020.

Web Links:

1. Virtual Lab on Vision and deep learning Lab, https://www.ee.iitb.ac.in/~viplab/
2. Virtual Lab on Computer Vision Laboratory https://www.iitk.ac.in/ee/computer-vision-lab
3. Course on Modern Computer Vision https://www.youtube.com/playlist?list=PLzWRmD0Vi2KVsrCqA4VnztE4t71KnTnP5
4. Coursera course on Advanced Computer Vision with TensorFlow https://www.coursera.org/learn/advanced-computer-vision-with-tensorflow
5. Udemy course on Deep Learning and Computer Vision A-Z™: OpenCV, SSD & GANs | Udemy
6. Vision Lab: Computeer Vision http://cse.iitm.ac.in/lab_details.php?arg=NQ
7. Funded Projects on Computer Vision at NAVER LABS Europe https://europe.naverlabs.com/research/computer-vision/

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Programming Laboratory (DJS22DSL604)

Prerequisites: Core Java and OOP concepts.

Objective: The objective of this course is:

1. To familiarize students with advanced object-oriented concepts and design patterns in Java for creating scalable applications.
2. To enable students to optimize data handling through the Java Collections Framework, generics, and the Streams API.
3. To equip students with skills to design, build, and secure web applications using Spring and Spring Boot frameworks, with a focus on database connectivity and microservices architecture.

Outcomes: On completion of the course, the learner will be able to:

1. Apply advanced object-oriented concepts and design patterns in Java to develop scalable and maintainable solutions for real-world problems.
2. Optimize data processing and performance using the Java Collections Framework, Streams API, and generics.
3. Build secure, database-driven web applications using Spring and Spring Boot, with an understanding of microservices and RESTful web services.

Programming Laboratory (DJS22DSL604)		
Unit	**Description**	**Duration**
1	**Advanced Object-Oriented Concepts** **Design Patterns:** Introduction to design patterns: Singleton, Factory, Observer, Strategy, Implementing design patterns in Java. **SOLID Principles:** Understanding and applying SOLID principles for better design, Examples and case studies. **Interfaces and Abstract Classes:** Advanced uses of interfaces and abstract classes, Default methods in interfaces (Java 8 and above).	**05**
2	**Java Collections Framework and Advanced Streams** **Collections:** List, Set, Map, and Queue interfaces, ArrayList, LinkedList, HashSet, Tree Set, HashMap, LinkedHashMap. **Generics:** Introduction to generics in Java, Creating generic classes and methods Bounded type parameters. **Java Streams**: Introduction to Streams API (Java 8 and above), Creating streams from collections, arrays, and I/O, Stream operations: map, filter, reduce, collect, Parallel streams for performance optimization.	**08**
3	**Java Reflection API** **Introduction to Reflection:** Understanding the Java Reflection API Accessing and manipulating class properties at runtime. **Use Cases of Reflection:** Creating instances of classes dynamically Inspecting methods, fields, and annotations Use of Reflection in frameworks like Spring.	**04**

4	**Java Database Connectivity (JDBC)** **JDBC Overview:** Connecting to databases using JDBC, Executing SQL queries and managing results, ORM Frameworks. **Introduction to Object-Relational Mapping (ORM):** Overview of Hibernate and JPA,Creating a simple application using Hibernate.	**04**
5	**Spring Framework** **Introduction to Spring:** Overview of Spring Framework features, Inversion of Control (IoC) and Dependency Injection (DI). **Spring Core:** Understanding Beans, Application Context, and Bean Lifecycle Configuring Spring with XML and Java annotations. **Spring AOP (Aspect-Oriented Programming):** Introduction to AOP concepts, Creating and using aspects in Spring.	**06**
6	**Spring Boot Framework** **Introduction to Spring Boot**: Understanding its purpose and advantages over traditional Spring. **Setting Up Spring Boot Applications:** Project structure and configuration. **Building RESTful Web Services:** Creating REST APIs using Spring Boot. **Spring Data JPA:** Introduction to database interactions and repository pattern. **Securing Spring Boot Applications:** Basics of security in Spring Boot using Spring Security. **Profiles and Configuration:** Managing different environments and configurations in Spring. **Spring Boot Actuator**: Monitoring and managing Spring Boot applications. **Testing Spring Applications:** Unit and integration testing with Spring Test framework. **Micro services Architecture:** Basics of microservices, using Spring Cloud.	**12**
	Total	**39**

***Suggested List of Experiments*:**

Programming Laboratory (DJS22DSL604)	
Sr. No.	**List of Experiment**
1	Implementation of Functional Interfaces, Comparable and Comparator
2	Implementation of Optional Class, Date/Time API.
3	Implementation of Annotations.
4	Implementation of Singleton Design Patterns.
5	Implementation of Structural Design Patterns & Behavioural Design Patterns.
6	Creating JDBC application
7	Implementation of different collection types (stacks, queues, vectors etc)
8	Creation of generic classes, methods
9	Use reflection API to examine or modify the behaviour of methods, classes, and interfaces at runtime.
10	Using streams API to implement program logic by composing functions and executing them in a data flow.
11	Create a Spring application configured with both XML and Java annotations.
12	Implement logging functionality using Spring AOP.
13	Create a Spring Boot application with Spring Data JPA for database interaction.

Minimum 10 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (25m) and Computer Based Assessment (25m).

Books Recommended:

Textbooks:

1. Anita Seth, B.L. Juneja, "JAVA: ONE STEP AHEAD", Oxford University Press, 1st Edition, May 2017.
2. Patrick Niemeyer, Daniel Leuck, "Learning Java", O'Reilly Media, Inc, 4th Edition, June 2013.
3. Mert Caliskan, Kenan Sevindik, Rod Johnson, Jürgen Höller "Beginning Spring", Wrox February 2015.

Reference Books:

1. Herbert Schildt, "Java: The Complete Reference", 9th edition, McGraw Hill.
2. Uttam K. Roy, "Advanced Java Programming, Oxford University Press, 2015.

Weblinks:

1. Nptel Course: https://onlinecourses.nptel.ac.in/noc20_cs58/preview
2. Oracle links: https://docs.oracle.com/javase/tutorial/collections/; https://docs.oracle.com/javase/tutorial/jdbc/
3. Spring documentation: https://docs.spring.io/spring-boot/index.html

DEPARTMENT ELECTIVES

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Cloud Computing (DJS22DSC6011)

Cloud Computing Laboratory (DJS22DSL6011)

Pre-requisite: System Fundamentals and Basic Networking

Objectives:

1. Understand the core principles of cloud computing, including parallel and distributed computing concepts, and virtualization techniques.
2. Analyze the architecture of cloud computing, covering cloud service models, types of clouds, and key migration strategies.
3. Explore Virtual Private Cloud (VPC) concepts, Elastic Compute Cloud (EC2) services, and their role in cloud infrastructure design and management.
4. Learn cloud-based storage solutions, Database as a Service (DBaaS) offerings, and cloud security measures for data protection.

Outcomes: On completion of the course, the students will able to:

1. Demonstrate the ability to differentiate between parallel and distributed computing and understand the role of virtualization in cloud environments.
2. Apply knowledge of cloud architecture to select appropriate cloud service models and types, and perform effective cloud migrations.
3. Configure and manage VPCs, EC2 instances, and understand best practices for cloud networking and instance management.
4. Implement cloud storage solutions, leverage DBaaS, and ensure robust cloud security using industry-standard practices and AWS security services.

Cloud Computing (DJS22DSC6011)		
Unit	**Description**	**Duration**
1	**Introduction to Cloud Computing:** Principles of Parallel and Distributed Computing: Parallel vs. distributed computing, Elements of parallel computing and Distributed Computing. **Virtualization:** Characteristics of virtualized environments, Taxonomy of virtualization techniques: hosted, bare-metal, Hypervisor and Xen Architecture, Para virtualization with Compiler Support, CPU Virtualization, Other Virtualizations: Storage, Network, Desktop and Application Server Virtualization, Virtualization and cloud computing	**06**
2	**Cloud Computing Architecture:** The cloud reference model: SAAS, IAAS, PAAS, Types of clouds: Public, Private Hybrid, Community, Economics of the cloud, Open challenges. **Migrating Applications to the Cloud:** Key aspects, cloud migration techniques, phases during migration, cloud emulators.	**06**
3	**Virtual Private Cloud (VPC):** Introduction to VPC and its benefits, Networking concepts within a VPC (subnets, route tables, security groups) VPC peering and connectivity options, VPC design best practices and considerations. **Elastic Compute Cloud (EC2) Service:** Overview of EC2 and its role in cloud computing, EC2 instance types and families, Provisioning and launching EC2 instances, configuring security groups and key pairs, Managing EC2 instances (start, stop, terminate), Elastic IP addresses and Elastic Network Interfaces (ENIs).	**08**

4	**Cloud-Based Storage:** Provisioning Cloud Storage, Exploring Cloud Backup Solutions, Cloud Storage Interoperability **Database as a Service** Key advantages of Database as a service offering, Amazon S3, Elastic Block Store (ESB), Amazon SimpleDB,	**06**
5	**Understanding Cloud Security:** Securing the Cloud: The security boundary, Security service boundary, Security mapping, Securing Data: Brokered cloud storage access, Storage location and tenancy, Encryption, Auditing and compliance, Establishing Identity and Presence, Identity protocol standards: Windows Azure identity standards **Data Protection:** protect data at rest and in transit, Identify Amazon Simple Storage Service (Amazon S3) protection features, Encrypt data in Amazon S3, Differentiate between client-side encryption (CSE) and server-side encryption (SSE), Identify Amazon Web Services (AWS) services that help protect your data.	**08**
6	**Administration for Clouds:** The AAA model, single sign-on flor clouds, industry implementation for AAA, authentication management standards for controlling access, SAML, authorization management, accounting for resource utilization.	**05**
	Total	**39**

Suggested List of Laboratory Experiments:

Cloud Computing Laboratory (DJS22DSL6011)	
S No	**Name of Experiment**
1	Virtualisation: • Hosted Virtualisation • Bare Metal Virtualisation
2	Host a Static Website on cloud.
3	Create and migrate relational database on cloud.
4	Create a Virtual Private Clouds and establish connections between each other.
5	Implement user level authentication on your cloud applications.
6	Implement Load balancing on your created cloud application.
7	Automate Infrastructure Development.
8	Implement serverless architecture and configure notification services.
9	Implement Hybrid storage and Data Migration.
10	Mini Project

Minimum 10 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (15m) and Assignment/Quizzes (10m).

Books Recommended:

Text Books

1. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah, "Cloud computing Black Book" Dreamtech Publication, 2014.
2. Rajkumar Buyya,"Mastering Cloud Computing", McGraw Hill Education, 2017
3. Ray Rafaels, "Cloud Computing: From Beginning to End," CreateSpace Independent Publishing, 2015.

Reference Books

1. Temitayo Fagbola, Kamal Kant Hiran, "Cloud Computing: Master The Concepts, Architecture and Applications with Real-World Examples And Case Studies", BPB Publications, 2019.
2. Dr. Sunilkumar, S. Manvi, "Cloud Computing: Concepts and Technologies", CRC Press, 2021
3. Ricardo Puttini, Thomas Erl, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture," Pearson Publication, 2013
4. Michael J Kavis,"Architecting the Cloud," Wiley, 2014.
5. Thomas Erl, Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", Pearson Education, 2014

Web Links:

1. A course on Cloud Computing: https://onlinecourses.nptel.ac.in/noc22_cs20/preview
2. A comprehensive guide to Social Network Analysis: https://www.analyticsvidhya.com/blog/2021/04/what-is-cloud-computing/
3. AWS Cloud Services: https://docs.aws.amazon.com//?nc2=h_ql_doc_do

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Recommender System (DJS22DSC6012)

Recommender System Laboratory (DJS22DSL6012)

Pre-requisite: Statistics for Data Science, and Machine Learning – I.

Course Objectives: To provide students with the basic concepts of Recommender Systems, design space, trade-offs and its application in various domain.

Outcomes: Students will be able to

1. Compare different types of Recommender Systems.
2. Understand various issues related to recommender system development.
3. Design a recommender system for a given problem.
4. Relate data collected from a recommender system to understand user preferences and/or behavior.

Recommender System (DJS22DSC6012)		
Unit	**Description**	**Duration**
1	**Introduction to Recommender Systems:** Recommender Systems Function, Techniques, Application and Evaluation, Recommender Systems and Human Computer Interaction, Trust, Explanations and Persuasiveness, Conversational Systems, Visualization, Biases in Recommender Systems: Statistical, cultural and cognitive, data and algorithm bias and self-selection biases, Issues working with RSs data sets: The cold-start problem. **Recommendation System Properties:** User Preference, Prediction Accuracy, Coverage, Confidence, Trust, Novelty, Serendipity, Diversity, Utility, Risk, Robustness, Privacy, Adaptivity. **Performance evaluation of RSs Experimental settings:** Evaluation metrics: Rating prediction and accuracy, Ranking Measures: NDPM, Spearman's ρ, R-Score, MAP, NDCG, MRR, implicit/explicit. Other metrics: fairness, coverage, diversity, novelty, serendipity.	**09**
2	**Content-based Recommender System:** High level Architecture of Content-based Systems, Advantages and Drawbacks of Content-based Filtering, Item profiles, discovering features of documents, obtaining item features from tags, representing item profiles, Methods for Learning User Profiles, Similarity based retrieval, Classification algorithms, Knowledge based recommendation: Knowledge representation and reasoning, Case based recommenders.	**05**
3	**Neighborhood-based Recommendation Methods:** Advantages of Neighborhood Approaches, Neighborhood-based Recommendation, User-based Rating Prediction, User-based Classification Regression Vs Classification, Item-based Recommendation, User-based Vs Item-based Recommendation, Rating Normalization, Similarity Weight Computation, Neighborhood Selection, Advanced Techniques: Dimensionality Reduction Methods, Graph-based Methods, Feature selection. Item representation, Methods for learning user profiles. Model based and preprocessing based approaches, Attacks on collaborative recommender systems	**06**

4	**Collaborative filtering-based Recommender System:** Baseline predictors through least squares, Implicit feedback, Matrix factorization models: SVD, SVD++, Time-aware factor model, Comparison, echo chambers, data drift and concept drift. **Neighborhood models**: Similarity measures, Similarity-based interpolation, jointly derived interpolation weights. Global neighborhood model, Factorized neighborhood model, Temporal models. Step-by-step solution of the RS problem. Temporal dynamics at neighborhood models and Between neighborhood and factorization.	**06**
5	**Constraint-based Recommenders**: Development of Recommender Knowledge Bases, User Guidance in Recommendation Processes, Calculating Recommendations. **Context-Aware Recommender Systems Trust:** Context in Recommender Systems, Modeling Contextual Information in Recommender Systems. Paradigms for Incorporating Context in Recommender Systems: Contextual Pre- Filtering, Contextual Post-Filtering, Contextual Modeling, Combining Multiple Approaches, Additional Issues in Context-Aware Recommender Systems.	**07**
6	**Hybrid approaches:** Deep Recommender systems, Multimodal Recommenders, Monolithic hybridization design: Feature combination, Feature augmentation, Parallelized hybridization design: Weighted, Switching, Mixed, Pipelined hybridization design: Cascade Meta-level, Limitations of hybridization strategies, deployment of recommender systems for given timeframe/users/items, Testing and Explainability in recommenders.	**06**

Suggested List of Experiments:

Recommender System Laboratory (DJS22DSL6012)	
Sr. No.	**Title of the Experiment**
1	Processing and analysis of public recommender systems datasets, and performance evaluation and comparison / Master spreadsheet-based tools.
2	Compare and analyze performance of Content-based recommendation engine on different datasets for Book, Movie, Song, product Recommendation.
3	Implement Recommendation System using K-Nearest Neighbors and evaluate its performance on different dataset.
4	Build project-association recommenders using association rule mining.
5	Build a Recommendation Engine with Item-Based Collaborative Filtering.
6	Implement Context-Aware Recommender Systems Trust.
7	Build Constraint-based Recommenders to provide valuable support for users searching for products and services in e-commerce environments.
8	Implement Hacker News algorithm /Subreddit User Recommendation System based on Netflix's Algorithm.

9	Implement Bayesian personalized ranking using matrix factorization algorithm.
10	Implement Google PageRank algorithm for recommendation.
11	Implement unsupervised learning - Autoencoders and Restricted Boltzmann Machines.
12	Implement recommender systems in 5G wireless networks for optimizing wireless network performance and deploy designed recommender System as Hosted Interactive Web Service on AWS.
13	Mini Project

Minimum 10 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (15m) and Assignment/Quizzes (10m).

Books Recommended:

Textbooks:

1. Jannach D., Zanker M. and FelFering A., "Recommender Systems: An Introduction", Cambridge University Press, 1st Edition, 2011.
2. Kim Falk, "Practical Recommender Systems", Manning, 1st Edition, 2019
3. Manouselis N., Drachsler H., Verbert K., Duval E., "Recommender Systems for Learning", Springer, 1st Edition, 2013.
4. C.C. Aggarwal, "Recommender Systems: The Textbook", Springer, 1st Edition, 2016.

Reference Books:

1. M.D. Ekstrand, J.T. Riedl, J.A. Konstan, "Collaborative filtering recommender systems", Now publishers, 1st Edition, 2011.
2. J. Leskovec, A. Rajaraman and J. Ullman, "Mining of massive datasets", Cambridge, 2nd Edition, 2012.
3. Rounak Banik, "Hands-On Recommendation Systems with Python: Start building", Ingram short title, 2018
4. P. Pavan Kumar, S. Vairachilai, Sirisha Potluri, "Recommender Systems: Algorithms and Applications", CRC Press, 1st edition, 2021.

Web Links:

1. Udemy course on Recommender Systems and Deep Learning in Python:
2. https://realpython.com/build-recommendation-engine-collaborative-filtering
3. Coursera course on Recommender Systems Specialization: https://www.coursera.org/specializations/recommender-systems

Prepared by Checked by Head of the Department Principal

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Embedded Systems & RTOS (DJS22DSC6013)

Embedded Systems & RTOS Laboratory (DJS22DSL6013)

Pre-requisite courses: Microprocessors and Microcontrollers

Objectives: To study concepts involved in embedded hardware and software for system realization.

Outcomes: On completion of the course, learner will be able to:

1. Identify and describe various characteristic features and applications of embedded systems.
2. Analyze and identify hardware for embedded system implementations.
3. Analyze and identify various software issues involved in embedded systems for real timerequirements.
4. Analyze and explain the design life-cycle for embedded system implementation.

Embedded Systems & RTOS (DJS22DSC6013)		
Unit	**Description**	**Duration**
1	**Introduction to embedded systems** Characteristics and Design metrics of Embedded system, Real time systems: Need for Real-time systems, Hard-Soft Real-time systems.**,** Challenges in Embedded System Design: Power, Speed and Code density, Power supply considerations in Embedded systems: Low power features-Idle & Power, down mode, Sleep mode, and Brown-out detection	**04**
2	**Embedded Hardware** Introduction to Embedded Architecture: Embedded cores, Types of memories, Sensor Interface, Communication Interfaces: Comparative study of serial communication interfaces (RS-232,RS-485), SPI, I2C, CAN, USB, Wired LAN (Ethernet) (IEEE 802.3), Wireless LANs & Long Distance Comm. Wireless Fidelity – LoRA Mesh. Selection criteria of above interfaces.	**08**
3	**ARM Architecture:** Comparative study of A, R & M series of processors with introduction to different families and their capabilities- use cases. Understanding the Cortex M0/0+, M3, M4, M33, M55 and M7 in terms of scalability from low performance applications to base server applications and moving towards 64-bit architecture. Introducing Pipelining Concepts & basic instruction features such as ARM Mode, Thumb and Thumb 2 mode, Instruction and Data Caches (Cortex-M7 and Cortex-A); FPU & MPU Coprocessors. Introducing the STM 32 F446 RE Nucleo Board and its capabilities with sensor interfacing	**08**
4	**Introduction to RTOS** **Real-time Operating system:** Need of RTOS in Embedded system software and comparison with GPOS, Foreground/Background processes, Interrupt latency, Task, Task- states, Multi-tasking, Context switching, Task scheduling, Scheduling algorithms - Rate Monotonic Scheduling, Earliest Deadline First, Inter-process communication, Semaphore, Mailbox, Message queues, Event timers, Task synchronization- Shared data, Priority inversion,Deadlock. Memory Management, Shared Devices and Mutex (Priority Inversion within it)Critical Code Sections (Disable Scheduler temporarily).	**08**

5	**Practical Implementation of RTOS Concepts with FreeRTOS:** Introduction to FreeRTOS: Overview of FreeRTOS and its features, Benefits of using FreeRTOS in embedded systems. Setting up the STM32 F446 Nucleo Board for FreeRTOS, Implementation of task scheduling. Context switching in FreeRTOS. Synchronization Mechanisms in FreeRTOS. Creating and using semaphores. Managing memory in FreeRTOS. Demonstrating task synchronization techniques (e.g., shared data management, priority inversion handling).	**06**
6	**System Integration, Testing and Debugging Methodology** Embedded Product Design Life-Cycle (EDLC), Hardware-Software Co-design **Testing & Debugging:** Boundary-scan/JTAG interface concepts, Black-Box testing,White-Box testing, hardware emulation, logic analyzer.	**05**
	Total	**39**

Suggested List of Experiments:

Embedded Systems & RTOS Laboratory (DJS22DSL6013)	
Sr. No	**Title of the Experiments**
1	Introduction to STM 32 446 Nucleo Board & Getting started with Mbed
2	Introduction to the FRDM 64F Platform & Getting Started with Mbed
3	Porting, Compiling, Downloading & Running your first program – Blinky LED
4	Interfacing LCD, Speaker, Temperature Sensor & Accelerometer with Nucleo Board
5	Introduction to FreeRTOS and FreeRTOS Task Creation – Understanding the System Core Clock
6	FreeRTOS Hello World App, Semi hosting & UART Setup
7	FreeRTOS App Debugging using Segger System View Tools
8	FreeRTOS Scheduler, Kernel Interrupts, RTOS Tick and SysTick Timer
9	FreeRTOS Context Switching & Task Notification and Task Deletions
10	FreeRTOS Queue Management, Semaphore for Synchronizations, Mutual Exclusion and Memory Management

Minimum 6 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (15m) and Assignment/Quizzes (10m).

Books Recommended:

Textbooks:

1. Dr. K. V. K. K. Prasad, "Embedded Real Time System: Concepts, Design and Programming", Dreamtech, New Delhi, 2014.
2. Perry Xiao, "Designing Embedded Systems & Internet of Things with ARM mbed", Wiley, 1st Edition, 2018.
3. Sriram Iyer, Pankaj Gupta, "Embedded Real Time Systems Programming", Tata McGraw Hill Publishing Company ltd., 1st Edition, 2017.

Reference Books:

1. David Simon, "An Embedded Software Primer", Pearson, 1st Edition, 2009.

2. Jonathan W. Valvano, "Embedded Microcomputer Systems–Real Time Interfacing", Publisher-Cengage Learning, 3rd Edition, 2012.
3. Andrew Sloss, Domnic Symes, Chris Wright, "ARM System Developers Guide Designing and Optimising System Software", Elsevier, 1st Edition, 2004
4. Frank Vahid, Tony Givargis, "Embedded System Design–A Unified Hardware/Software Introduction", John Wiley & Sons Inc., 1st Edition, 2002.
5. Shibu K. V., "Introduction to Embedded Systems", Tata McGraw Hill Education Private Limited,New Delhi, 1st Edition, 2009.

Web Links:

1. A course on Embedded System Design with ARM: https://archive.nptel.ac.in/courses/106/105/106105193/
2. A course on Real Time Operating System: https://onlinecourses.nptel.ac.in/noc20_cs16/preview
3. A course on Design of Internet of Things: https://onlinecourses.nptel.ac.in/noc21_ee85/preview

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Computational Neuroscience (DJS22DSC6014)

Computational Neuroscience Laboratory (DJS22DSL6014)

Pre-requisite: Artificial Intelligence, Machine Learning, Statistics and Programming Skills.

Course Objectives:

This course aims to provide students with a strong foundation in the field of Cognitive Neuroscience, a field that studies the intricate links between the mind, the brain, and behaviour. Students will learn methods to replicate human behaviour of how to sense and perceive the world, act in it, learn and think about it, and remember it.

Outcomes: Students will be able to:

1. Understand the computational neural structure
2. Illustrate coding and decoding models
3. Describe neural circuit and network models
4. Apply information theory concepts on Brain mapping

Computational Neuroscience (DJS22**DSC6014)**		
Unit	**Description**	**Duration**
1	**Introduction:** Descriptive model, Mechanistic model and Interpretive model, Properties of Neurons, recording neuronal responses, Spike trains and firing rates: Measuring Firing Rates, Tuning curves; Stimulus: The spike Triggered Average, White noise Stimuli, Multiple Spike Triggered Averages and Spike Triggered correlations, Spike-Train Statistics – Homogenous and Inhomogeneous Poisson Process, Auto correction Function, The Poisson Spike Generator, The Neural Code	**06**
2	**Neural Encoding:** Introduction to retina, LGN, V1, area 17, Techniques of neural data collection, Reverse-Correlation Methods: Simple Cells, Spatial Receptive Fields, Temporal Receptive Fields, Response of a simple cell to counterphase grating, Space-time Receptive Fields; Linear Filtering, Non-linear input-output function, Basic encoding model, Impact of PCA on encoding model, Non-linear representation with Gaussian, Binomial and Poisson representation, The generalized linear model.	**08**
3	**Neural Decoding**: Encoding decoding, Discrimination: ROC curves, The likelihood ratio test; Population Decoding: optimal decoding method, Fisher Information; Spike-Train Decoding	**07**
4	**Information Theory:** Entropy and Mutual Information, Information and entropy maximization: for single Neuron, populations of neurons; Utilization in Retinal Ganglion cell Receptive Fields-Application, Entropy and information for spike trains	**06**
5	**Model Neurons (Neuroelectronic):** Significance of Neuroelectronic in neuroscience, Neuronal Biophysics- Membrane potentials, Single compartment models, Integrated and fire models, Voltage dependent conductance, The Hodgkin- Huxley Model, Channel Simulation, Synaptic Transmission Models, Synaptic Inputs to Integrate-and-Fire Models	**06**

6	Network Models- Introduction - Importance of connectivity and dynamics, Firing-Rate Models - Dynamics of average firing rates, Feedforward - Neural mapping, Recurrent Networks: Linear and Nonlinear Recurrent Networks, Stability and oscillatory dynamics.	**06**
	Total	**39**

Suggested List of Experiments:

Computational Neuroscience Laboratory (DJS22DSL6014)	
Sr. No.	**Title of the Experiments**
1.	Introduction to EEG recordings. Theory, physiology, practical aspects of recording and analysing scalp- recorded brain potentials.
2.	Designing experiments: Control, manipulation, repeated trials, and balanced conditions. Application to studies with brain recordings.
3.	Experimental approach to studying the working human brain and body. How to use Brain Voyager Brain Tutor. How to use the BESA dipole simulator.
4.	Research design and the traditional statistical foundations of experimental research: T-test. Analysis of variance. Evaluate sample data and data from a standard experiment.
5.	Recording dense-array EEG: Practical introduction.
6.	EEG analysis: How to get from the raw recording to brain waves. An example analysis.
7.	Mini Project

Minimum 6 experiments from the above-suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

*The Term Work will be calculated based on Laboratory Performance (15m) and Assignment/Quizzes (10m).

Books Recommended:

TextBooks

1. José Luis Bermúdez, "Cognitive Science: An Introduction to the Science of the Mind", Cambridge University Press, New York, 2nd Edition, 2014.
2. Jay Friedenberg, Gordon Silverman and Michael J. Spivey, "Cognitive Science: An Introduction to the Study of Mind", SAGE Publication, 4th Edition, 2021.

Reference Books

1. Michael Gazzaniga, Richard B Ivry, George R Mangun, "Cognitive Neuroscience the Biology of the Mind," W. W. Norton & Company Publication, 5th Edition, 2019.
2. Daniel Kolak, William Hirstein, Peter Mandik, Jonathan Waskan," Cognitive Science: An Introduction to Mind and Brain, Taylor and Francis, 1st Edition, 2006.

Weblinks:

1. Cognitive Science: https://plato.stanford.edu/entries/cognitive-science/
2. Cognitive Neuroscience: https://plato.stanford.edu/entries/cognitive-science/

Program: Computer Science and Engineering (Data Science) **Semester: VI**

Course: Innovative Product Development-IV (DJS22ILLL2)

Course Objectives:

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuringthat a solution is found out to address the same by designing and developing an innovative product.
2. To familiarize the students with the process of designing and developing a product, while they work as part of ateam.
3. To acquaint the students with the process of applying basic engineering fundamentals, so as to attempt at thedesign and development of a successful value-added product.
4. To inculcate the basic concepts of entrepreneurship and the process of self-learning and research required toconceptualise and create a successful product.

Course Outcome: On completion of the course, student should be able to:

1. Identify the requirement for a product based on societal/research needs.
2. Apply knowledge and skills required to solve a societal need by conceptualising a product, especially whileworking in a team.
3. Use standard norms of engineering concepts/practices in the design and development of an innovative product.
4. Draw proper inferences through theoretical/ experimental/simulations and analyse the impact of the proposedmethod of design and development of the product.
5. Develop interpersonal skills, while working as a member of the team or as the leader.
6. Demonstrate capabilities of self-learning as part of the team, leading to life-long learning, which could eventuallyprepare themselves to be successful entrepreneurs.
7. Demonstrate product/project management principles during the design and development work and also excel inwritten (Technical paper preparation) as well as oral communication.

Guidelines for the proposed product design and development:

- Students shall convert the solution designed in semester 3 and 4 into a working model, using various components drawn from their domain as well as related interdisciplinary areas.
- The working model is to be validated with proper justification and the report is to be compiled in a standard format and submitted to the department. Efforts are to be made by the students to try and publish the extended technical paper, either in the institute journal, "Techno Focus: Journal for Budding Engineers" or at a suitable publication, approved by the department research committee/ Head of the department.
- Faculty supervisor may provide inputs to students during the entire span of the activity, spread over 2 semesters, wherein the main focus shall be on self-learning.
- A record in the form of an activity log-book is to be prepared by each team, wherein the team can record weekly progress of work. The guide/supervisor should verify the recorded notes/comments and approve the same on a weekly basis.
- The focus should be on self-learning, capability to design and innovate new products as well as on developing the ability to address societal problems. Advancement of entrepreneurial capabilities and quality development of the students through the year long course should ensure that the design and development of a product of appropriate level and quality is carried out, spread over two semesters, i.e. during the semesters V and VI.

Guidelines for Assessment of the work:

- The review/ progress monitoring committee shall be constituted by the Head of the Department. The progress of design and development of the product is to be evaluated on a continuous basis, holding a minimum of two reviews in each semester.
- In the continuous assessment, focus shall also be on each individual student's contribution to the team activity, their understanding and involvement as well as responses to the questions being raised at all points in time.
- Oral examination should be conducted by Internal and External examiners. Students have to give presentation and demonstration on their working model
- The distribution of marks for term work shall be as follows:

1. Marks awarded by the supervisor based on log-book: 10
2. Marks awarded by review committee: 10
3. Quality of the write-up : 05

The overall work done by the team shall be assessed based on the following criteria:

1. Quality of survey/ need identification of the product.
2. Clarity of Problem definition (design and development) based on need.
3. Innovativeness in the proposed design.
4. Feasibility of the proposed design and selection of the best solution.
5. Cost effectiveness of the product.
6. Societal impact of the product.
7. Functioning of the working model as per stated requirements.
8. Effective use of standard engineering norms.
9. Contribution of each individual as a member or the team leader.
10. Clarity on the write-up and the technical paper prepared.
11. The semester reviews (V and VI) may be based on relevant points listed above, as applicable.

Guidelines for Assessment of Semester Reviews:

- The write-up should be prepared as per the guidelines given by the department.
- The design and the development of the product shall be assessed through a presentation and demonstration of the working model by the student team to a panel of Internal and External Examiners, preferably from industry or any research organisations having an experience of more than five years, approved by the Head of the Institution. The presence of the external examiner is desirable only for the 2nd presentation in semester VI. Students are compulsorily required to present the outline of the extended technical paper prepared by them during the final review in semester VI.

Prepared by Checked by Head of the Department Principal