Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

NEP Compliant Structure & Syllabus of

Department of Computer Science and Engineering – Data Science Pattern

'A-24'

S. Y. B. Tech.

Effective from Academic Year 2025-26

Prepared by: - Board of Studies in Computer Science and Engineering – Data Science

Approved by: - Academic Board, Vishwakarma Institute of Technology,

Pune

Chairman-BoS

Dean - Academics

Chairman – Academic Board

Contents

Program Outcomes

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, social and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Academic Information – Please visit www.vit.edu

FF No.: 653

Structure of Second Year CSE-DS for Academic Year 2025-26

Course Code	Course Name	Type	Teaching Learning Scheme				
			Th	Tut	Lab	Hrs. / Week	Credits
Module 3 - Semester 1							
DS2003	FUNDAMENTALS OF DATA STRUCTURE AND ALGORITHMS	**DC**	3	0	2	5	4
DS2004	FUNDAMENTALS OF DATA SIENCE	**DC**	2	0	2	4	3
DS2005	OBJECT ORIENTED PROGRAMMING	**DC**	2	0	2	4	3
DS2006	COMPUTER NETWORK	**DC**	1	0	2	3	2
DSM001	DATA VISUALIZATION	**IEL**	2	1	0	3	3
HS2002	FROM CAMPUS TO CORPORATE - 1	**IEL**	2	0	0	2	2
HS2001	REASONING AND APTITUDE DEVELOPMENT - 3	**DC**	1	0	0	1	1
DS2001	DESIGN THINKING - 3	**SM**	0	1	0	1	1
DS2002	ENGINEERING DESIGN AND INNOVATION - III	**SM**	0	0	2	2	2

Assessment scheme

Course Code	Course Name	ESE TH (W)	ESE TH (O)	CVV	CP	LAB	GD/ PPT/ HA	MSE (O)	T1 (O)	T2 (O)	ESE	PRACT + CVV	Total
DS2003	FUNDAMENTALS OF DATA STRUCTURE AND ALGORITHMS				30	10						40 + 20	100
DS2004	FUNDAMENTALS OF DATA SIENCE	40		20	30	10							100
DS2005	OBJECT ORIENTED PROGRAMMING		40		30	10	GD/ PPT 20						100
DS2006	COMPUTER NETWORK	40		30			GD/ PPT 30						100
DSM001	DATA VISUALIZATION						HA 30		35	35			100
HS2002	FROM CAMPUS TO CORPORATE - 1		50					50					100
HS2001	REASONING AND APTITUDE DEVELOPMENT - 3										100		100
DS2001	DESIGN THINKING - 3										100		100
DS2002	ENGINEERING DESIGN AND INNOVATION - III							30			70		100

Chairman-BoS

Dean - Academics

FF No. : 654

DS2003: FUNDAMENTALS OF DATA STRUCTURE AND ALGORITHMS

Credits: 4 **Teaching Scheme Theory:** 3 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites:

Introduction to Programming, Discrete Mathematics, Basic Problem-Solving Skills

Course Objectives:

1. Understand basic data structures and algorithmic complexity.
2. Implement searching, sorting, and recursion algorithms.
3. Apply advanced structures like trees, graphs, and hashing.
4. Design solutions using stacks, queues, and linked lists.
5. Compare and optimize algorithm performance.
6. Use tools to develop and test algorithmic solutions

Course Relevance:

These subject builds essential skills in organizing data and designing efficient algorithms, enabling effective problem-solving and optimized software development.

SECTION-1 Introduction to Data Structures and Complexity
Abstract Data Types, need of data structures, classification of data structures, complexity analysis of algorithms using Big-O, Big-Ω and Big-Θ notations, time-space trade-off, recursion basics and applications.
SECTION-2 Arrays and Searching & Sorting Techniques
1D and 2D arrays, sparse matrix representation, Simple and Fast Transpose, linear search, binary search, bubble sort, selection sort, insertion sort, merge sort, quick sort, time complexity of all techniques.
SECTION-3 Stacks and Queues
Stack operations and applications like expression evaluation, infix to postfix conversion, prefix conversion, implicit and explicit stack, recursion using stack; queue operations, circular queue, priority queue, deque, applications of queues in real-world problems.
SECTION-4 Linked Lists
Singly linked list, doubly linked list, circular linked list, operations (insert, delete, traverse, search), applications such as polynomial operations, memory management using dynamic allocation.
SECTION-5 Trees
Tree terminologies, binary trees, binary search trees (BST), tree traversals (inorder, preorder, postorder), AVL trees, expression trees, heap trees, applications in decision making and file systems.
SECTION-6: Graphs and Hashing
Graph representation using adjacency list and matrix, BFS, DFS, applications of graphs in networking and social media, hash tables, collision resolution techniques: chaining and open addressing.

List of Practicals: (Any Ten)

1. Design a recursive program to compute the factorial of a given number, illustrating how function calls build up and unwind like a stack.
2. Simulate bed allocation in a hospital ward using a sparse matrix where only a few beds are occupied, and display the occupied vs. available status efficiently.
3. Develop a phonebook application where users can search for contacts using both linear and binary search methods. Compare the search performance for small vs. large contact lists.
4. Build a shopping cart system where products can be sorted by price or ratings using various sorting algorithms. Measure and compare the time efficiency of each method for large orders.
5. Create a calculator that validates and evaluates arithmetic expressions using stacks. Implement infix to postfix conversion and postfix evaluation to simulate real-time computation.
6. Design a queue-based ticketing system (like in banks or hospitals) where customers are issued tokens and served in their order of arrival. Use a circular queue to manage overflow efficiently.
7. Simulate how train coaches (bogies) are connected using different linked lists. Each coach is a node; use singly, doubly, and circular lists to represent and manipulate various train configurations such as adding, removing, and navigating between coaches.
8. Create a contact management system where names are stored in a Binary Search Tree (BST). Implement insertion, deletion, and search functionalities to maintain a well-organized directory.
9. Design a city navigation simulator using graphs where roads between landmarks are represented as edges. Use BFS and DFS to find paths from one place (e.g., home) to another (e.g., hospital or restaurant).
10. Implement a phone directory system using hash tables where each contact is stored by name. Handle collisions using separate chaining and provide options to add, delete, and search for contacts efficiently.
11. Develop a system to manage a library shelf where book titles are stored and sorted alphabetically. Use sorting techniques like insertion and quick sort and compare their performance.
12. Simulate an undo-redo feature of a text editor using two stacks. Enable users to type, undo recent changes, and redo actions as seen in common writing tools.
13. Create a task scheduling system where each task has a priority level. Use a priority queue to ensure higher priority tasks (e.g., emergency alerts) are served before others.
14. Simulate a memory management module where memory blocks are allocated and freed using linked lists. Implement dynamic memory allocation.
15. Build a laundry basket simulator where clothes are added one by one and removed in reverse order using a stack. Display the order in which clothes are removed for washing.
16. Develop an inventory management system for a small retail store where products are stored in a hash table using their IDs as keys. Implement features to add, search, update, and remove products efficiently while resolving collisions using open addressing.

List of Projects:

1. Smart Hospital Management System (Description: Manage patient queues, emergency prioritization, bed allocation using sparse matrix, and patient search with hashing.)

2. E-Commerce Cart and Inventory System (Description: Handle a shopping cart where products are sorted by price/rating, inventory tracked, and undo-redo implemented for cart updates.)
3. City Route Planner and Navigation Assistant (Description: Simulate city paths with graphs and implement BFS/DFS to find routes between locations.)
4. Digital Library Book Manager (Description: Store and manage book records, search by title, sort alphabetically, and track borrow/return actions.)
5. Student Academic Dashboard (Description: Maintain student marks, attendance, course enrollments, and performance summaries.)
6. Login and Credential Manager (Description: Manage usernames and passwords with hashing. Include features like login attempts tracking and user lookup.)
7. Supermarket Billing System (Description: Simulate billing with product scanning, subtotal calculation, discounts, and inventory updates.)
8. To-Do List Manager (Description: Let users add, remove, and track tasks. Include priority tasks and optional reminders or undo functionality.)
9. Expression Solver and Visualizer (Description: Create a calculator that converts infix expressions to postfix and evaluates them step-by-step.)
10. Railway Coach Management Simulator (Description: Model how coaches are connected in a train using linked lists. Allow coach addition, removal, and navigation.)

Suggest an assessment Scheme:
Suggest an Assessment scheme that is best suited for the course. Ensure 360-degree assessment and check if it covers all aspects of Blooms Taxonomy.

CP	LAB	PRACT	CVV
30	10	40	20

CP - Course Project
LAB - Continuous Assessment
PRACT – External Practical Exam
CVV – Comprehensive Viva

Text Books: (As per IEEE format)

1. E. Horowitz, S. Sahni, D. Mehta; Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2017.
2. Y. Langsam, M. Augenstein, A. Tannenbaum; Data Structures Using C and C++, 2nd Edition, Pearson Education, 2006.

Reference Books: (As per IEEE format)

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein; Introduction to Algorithms, 4th Edition, MIT Press, 2022. https://mitpress.mit.edu/9780262033848/
2. Brad Miller, David Ranum; Problem Solving with Algorithms and Data Structures using Python; Open Book Project; 3rd Edition, 2023. https://runestone.academy/ns/books/published/pythonds/index.html
3. Narasimha Karumanch, Data Structures and Algorithms Made Easy, Publisher CareerMonk Publications, 2024

Moocs Links and additional reading material:

1. Prof. Naveen Garg; Data Structures and Algorithms; NPTEL – IIT Delhi; https://nptel.ac.in/courses/106/102/106102064
2. UC San Diego, National Research University Higher School of Economics; Data Structures; Coursera; https://www.coursera.org/learn/data-structures

Course Outcomes:

Course Outcomes:
Student should be able to

1. Analyze and compute time and space complexity of algorithms
2. Apply searching and sorting algorithms to solve computational problems
3. Apply stack and queue operations in applications
4. Develop and use various types of linked lists to manage dynamic data
5. Apply tree data structures for hierarchical data representation and operations
6. Demonstrate understanding of graph and hashing techniques and their applications.

CO PO Map

CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO:1	3	3		2	2							2	1	1
CO:2	3	3	2		2							1	2	1
CO:3	3	2	3		3								1	2
CO:4	3	2	3		3								2	1
CO:5	3	2	3		3							1	2	2
CO:6	3	3	3		3							1	2	3

CO attainment levels

CO	Attainment level
CO .1	
CO .2	
CO .3	

CO .4	
CO .5	
CO .6	

Future Courses Mapping:
Advanced Algorithms, Database Systems, Operating Systems, Software Engineering, Artificial Intelligence, Machine Learning, Computer Networks.

Job Mapping:
Software Developer, Data Scientist, System Analyst, Backend Engineer, AI/ML Engineer, Database Administrator, Network Engineer, Tech Consultant.

FF No. : 654

DS2004: Fundamentals of Data Science

Credits: 3

Teaching Scheme Theory: 2 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites:

Python programming, fundamental concepts in statistics, Basic Problem-Solving Skills

Course Objectives:

1. Understand the evolution, scope, lifecycle, and real-world applications of data science.
2. Learn data collection methods, preprocessing techniques, and feature extraction from various data sources.
3. Perform exploratory data analysis and create effective data visualizations using Python and visualization tools.
4. Develop basic machine learning models for regression and classification problems.

Course Relevance:

This course builds foundational skills in data science, enabling students to analyze real-world data, derive insights, and develop predictive models—preparing them for advanced courses and careers in analytics, machine learning, and AI-driven decision-making.

SECTION-1 Introduction to Data Science
Overview, Evolution and Scope, Components: Data, Algorithms, Insights, Data Science Life Cycle: Data acquisition, Analysis, Action, Applications of Data Science: Healthcare, Finance, Retail & E-commerce, Manufacturing, Smart Cities & IoT & Social Media & Marketing Case Study: Predictive Maintenance in Manufacturing – to identify the type of data requirement and to sketch a basic DS workflow.
SECTION-2 Data Collection, Cleaning, and Feature Extraction
Data Sources and Types: Types of data: structured, unstructured, semi-structured, Data formats: numerical, categorical, time-series, text. Sources: databases, APIs, IoT sensors, web, surveys. Data Collection Techniques: Manual vs automated collection methods, Basics of web scraping using tools like requests, BeautifulSoup. Data Cleaning and Preprocessing: Handling missing values: Removing duplicates and irrelevant data, Standardizing and converting data types, Detecting and treating outliers (Z-score, IQR – theory only), Encoding categorical variables: label encoding, one-hot encoding, Feature Extraction: Understanding features, Feature selection, Feature engineering, Converting non-numeric to numeric data for models Case Study: Social Media Sentiment Analysis- to process and extract features from user-generated text data to support sentiment classification
SECTION-3 Exploratory Data Analysis using Python
Exploratory Data Analysis (EDA) Concepts: Importance of EDA in Data Science projects, Basic statistics: Mean, Median, Mode, Standard Deviation, understanding distributions: Normal, Skewed, Correlation analysis (Pearson, heatmaps).Visualization Tools and Techniques:

matplotlib ,tableau, excel: Line plots, bar charts, histograms, scatter plots, funnel chart, heatmap,waterfall chart.
Case Study: Indian rainfall data (2010 to 2020)-to analyze trends across different regions.

SECTION-4 Model Building and Evaluation

Model Building: Introduction to model training workflow methods. Building simple models: Linear Regression, K-Nearest Neighbors (KNN).
Model Evaluation Techniques Metrics for Regression: RMSE, MAE, R^2 Score. Metrics for Classification: Accuracy, Precision, Recall, F1-score, Confusion Matrix, Cross-validation.
Case Study – Student Performance Prediction- machine learning model that predicts whether a student is likely to pass or fail based on personal, academic, and family-related data.

List of Practicals: (Any Ten)

1. Explore a sample dataset and summarize its structure, including column names, data types, and basic statistics.
2. Select a real-world dataset and describe its contents, variable types, and potential use cases.
3. Clean a dataset by identifying and handling missing values, duplicates, and inconsistent data formats.
4. Analyse the distribution and central tendencies of variables and detect any skewness or irregularities.
5. Create meaningful visual representations to understand trends, comparisons, and relationships in the data.
6. Divide the dataset into subsets for training and testing, and prepare it for analysis by standardizing the values.

Suggest an assessment Scheme:
Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

CP	LAB	ESE(W)	CVV
30	10	40	20

CP - Course Project
LAB - Continuous Assessment
ESE (W)– End Sem Exam (Written)
CVV – Comprehensive Viva

Text Books: (As per IEEE format)

1. C. O'Neil, R. Schutt; Doing Data Science; 1st Edition; O'Reilly Media; 2013
2. J. Grus; Data Science from Scratch: First Principles with Python; 2nd Edition; O'Reilly Media; 2019
3. W. McKinney; Python for Data Analysis; 2nd Edition; O'Reilly Media; 2018

Reference Books: (As per IEEE format)

1. C. O'Neil, R. Schutt; Doing Data Science; 1st Edition; O'Reilly Media; 2013. Accessed: Jun. 17, 2025. [Online]. Available: https://www.oreilly.com/library/view/doing-data-science/9781449363871/
2. J. Grus; Data Science from Scratch: First Principles with Python; 2nd Edition; O'Reilly Media; 2019. Accessed: Jun. 17, 2025. [Online]. Available: https://www.oreilly.com/library/view/data-science-from/9781492041122/
3. W. McKinney; Python for Data Analysis; 2nd Edition; O'Reilly Media; 2018. Accessed: Jun. 17, 2025. [Online]. Available: https://www.oreilly.com/library/view/python-for-data/9781491957653/
4. 4. C. Müller, S. Guido; Introduction to Machine Learning with Python; 1st Edition; O'Reilly Media; 2016. Accessed: Jun. 17, 2025. [Online]. Available: https://www.oreilly.com/library/view/introduction-to-machine/9781449369880/

Moocs Links and additional reading material:

1. J. Grus; 'Data Science from Scratch'; O'Reilly Learning; https://learning.oreilly.com/library/view/data-science-from/9781492041122/;
2. C. Brooks; 'Python Data Structures'; Coursera; https://www.coursera.org/learn/python-data;
3. R. Tibshirani, T. Hastie; 'Statistical Learning'; Stanford Online; https://online.stanford.edu/courses/sohs-ystatslearning-statistical-learning
4. A. Ng; 'Machine Learning'; Coursera; https://www.coursera.org/learn/machine learning

Course Outcomes:

Student should be able to

1. Explore the fundamentals, applications, lifecycle, and ethical aspects of data science in various real-world domains.
2. Collect and preprocess structured and unstructured data, and apply feature extraction techniques to prepare data for analysis.
3. Analyse data using statistical summaries and visualize relationships and patterns through suitable graphical representations.
4. Build and evaluate predictive models using appropriate techniques for classification and regression.

CO PO Map

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO12	PSO 1	PSO 2
CO:1	3	3		2	2							2	1	1
CO:2	3	3	2		2							1	2	1
CO:3	3	2	3		3								1	2
CO:4	3	2	3		3								2	1

CO attainment levels

CO	Attainment level
CO .1	
CO .2	

CO .3	
CO .4	

Future Courses Mapping:
This course serves as a foundation for advanced courses such as Machine Learning, Deep Learning, Big Data Analytics, Artificial Intelligence, Business Intelligence, and Data Visualization.

Job Mapping:

Data Analyst, Junior Data Scientist, Business Analyst, Machine Learning Engineer (entry-level), and Data Visualization Specialist.

FF No. : 654

DS2005: Object Oriented Programming

Credits: 3 **Teaching Scheme Theory:** 2 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites:

Introduction to Programming in C language, Basic Problem-Solving Skills

Course Objectives:

1. Demonstrate understanding of object-oriented programming fundamentals and implement modular programs using classes and objects in C++.
2. Apply object-oriented concepts like constructors, destructors, and operator overloading to develop reusable components.
3. Implement inheritance, polymorphism, and virtual functions for designing extensible and maintainable code.
4. Use exception handling and file I/O mechanisms effectively to build robust and persistent C++ applications.

Course Relevance:

This subject develops core competencies in object-oriented programming, equipping students with the ability to design modular, reusable, and efficient software solutions. Through mastery of C++ concepts like classes, inheritance, polymorphism, and templates, learners build a strong foundation for system-level programming, competitive coding, and real-world application development.

SECTION-1 Introduction to Object Oriented Programming
Fundamentals of Object-Oriented Programming (OOP) in C++: Variables, Data Types, and Operators, Control Structures, Loops and Iteration, Functions and Modular Programming, Basics of Console Input and Output Class, Dynamic Memory Allocation. Overview of OOPs Principles, Classes & Objects, Creation & destruction of objects, Data Members, Member Functions, Access Specifier, this Pointer, Constructor &Destructor, Static class member, Friend class and functions, Function Overloading, Operator Overloading Namespace.
SECTION-2 Object Oriented Programming Principles
Inheritance, Types of Inheritance, Base and Derived class Constructors , Down casting and upcasting, Function overriding, Virtual functions, Polymorphism, Pure virtual functions, Virtual Base Class, C++ Class Hierarchy, File Stream, Text File Handling , Binary File Handling , Error handling during file operations , Overloading << and >> operators.
SECTION-3 Templates, Operator Overloading and Standard Template Library
Function Templates, Class Templates, Overloading << and >> operators, Operator Overloading Concepts, Namespace, Standard Template Library (STL) components: vector, list, map, set, stack, queue
SECTION-4 Exception and File Handling in C++
Exception Handling: Benefits, Try and catch block, throw statement, pre-defined exceptions in C++, Writing custom Exception class, Stack Unwinding, Function Templates, Class Templates. File Handling: Streams, File Input/Output, Binary File Handling, Error Handling, Use Cases
List of Practical's: 1. Fundamentals of C++ a) Write a C++ program to calculate the area of a rectangle given its length and width. b) Write a C++ program that converts temperature from Celsius to Fahrenheit using the formula: Fahrenheit = (Celsius * 9/5) + 32.

c) Write a program that takes a user's age as input and determines if they are eligible to vote or not.

d) Implement a C++ program that generates the Fibonacci sequence up to a given number 'n' using loops.

2. Functions and Modular Programming

a) Write a program that calculates the factorial of a given positive integer using a recursive function.

b) Write a modular program that checks whether a given number is prime or not, utilizing a function for prime number testing.

3. Object-Oriented Programming (OOP)

a) Write a C++ class named Rectangle that has attributes for length and width. Implement methods to calculate the area and perimeter of the rectangle.

b) Write a program using OOP concepts to simulate a basic banking system. Implement classes for customers and accounts, allowing for deposits and withdrawals.

c) Write a program to use static data members and member functions.

d) Write a program to use this pointer for resolving naming conflicts.

e) Write a program to overload arithmetic operators using the friend function.

f) Implement overloading of comparison operators (==, <, etc.).

4. Inheritance and Polymorphism

a) Build a hierarchy of classes representing different shapes (e.g., Circle, Triangle, Rectangle) with a common base class. Implement a virtual function for calculating the area of each shape.

b) Extend the banking system to include different types of accounts (Savings, Checking) that inherit from a common Account class. Implement polymorphic behaviour for interest calculations.

5. Write a program using try, catch, and throw for arithmetic and input errors.
6. Write a function template for finding the maximum of two values.

List of Projects:

1. Multi-Player Game Engine Framework (Using OOP + Polymorphism + STL)
   Description:
   Design a modular game engine that supports multiple games like Tic Tac Toe, Snake, and Chess. Use inheritance and polymorphism for game rules, STL containers to manage player profiles and scores, and templates for generic game logic.
2. University ERP Simulator (Object-Oriented + File Handling + Exception Handling)
   Description:
   Simulate a mini ERP with modules for student info, staff, timetable, fee tracking, and attendance. Use classes and virtual functions, file storage for persistent records, and exception handling for login errors, data corruption, etc.
3. Compiler Intermediate Code Generator (Advanced OOP + Stack + STL Map)
   Description:
   Parse simplified mathematical expressions and generate three-address code (TAC) or postfix code. Use stack, maps for symbol tables, and template functions for evaluation logic.
4. Airline Reservation & Flight Management System (Linked Lists + Operator Overloading + File Streams)
   Description:
   Allow seat reservation, cancellation, and route planning. Overload operators for seat comparisons, use linked lists for dynamic seat mapping, and manage historical booking data via files.
5. Blockchain-based Voting System (File Handling + Hashing + STL)
   Description:
   Simulate a simple blockchain to securely record votes. Use file handling to persist blocks, hash functions for verification, and vectors/queues/maps from STL to manage users, blocks, and votes.
6. Dynamic Code Snippet Organizer (Templates + File Parsing + OOP)
   Description:
   A tool to manage C++/Python snippets. Use template classes to generalize storage, file handling to import/export snippets, and OOP to manage languages, tags, and categories.

7. AI-Powered Chatbot Framework (Polymorphism + STL + File Handling)
   Description:
   Design a rules-based chatbot with polymorphic response models (default, emotion-based, logic-based), maps and vectors for pattern matching, and text file storage for chat history and knowledge base.
8. Social Media Post Scheduler (Priority Queue + Templates + Exception Handling)
   Description:
   Simulate scheduling of posts with time-based priority. Use STL priority queues, template-based scheduling logic, and exception handling for invalid scheduling formats.
9. Memory Management Visualizer (Pointer Arithmetic + Custom Allocators)
   Description:
   Create a tool that visualizes memory allocation in arrays, objects, and pointers. Implement a custom memory allocator, simulate new/delete, and show fragmentation via console visualization.
10. Trading Bot Simulator for Stock Market (Templates + STL + File I/O + Inheritance)
    Description:
    Design a simulated trading bot that makes buy/sell decisions. Use template-based data structures for financial instruments, inheritance for bot strategies, and files to import/export market data.

Suggest an assessment Scheme:
Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

CP	LAB	GD/PPT	ESE(O)
30	10	20	40

CP - Course Project
LAB - Continuous Assessment
GD/PPT – Group Discussion/PowerPoint Presentation
TH(O) – Online Examination

Reference Books: (As per IEEE format)

1. Behrouz A. Forouzan, Richard F. Gilberg, “COMPUTER SCIENCE – A Structured Programming approach using C”, Indian Edition, Thomson, 3rd edition, 2007, ISBN: 9788131517888.
2. Bjarne Stroustrup, ― The C++ Programming language‖, Third edition, Pearson Education.1997, ISBN 9780201889543. 3. Kernighan, Ritchie, “The C Programming Language”, Prentice Hall of India, 1988. ISBN 0-13-110362-8.
3. Robert Lafore, ―Object-Oriented Programming in C++, fourth edition, Sams Publishing, 2002, ISBN:0672323087 (ISBN 13: 9780672323089)
4. Herbert Schildt, ―C++ The complete reference, Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00-72226805
5. E. Balagurusamy- Object-oriented programming with C++, fourth edition, Mc Hill Professional,2008, ISBN 978-0-07-066907-9

Moocs Links and additional reading material:

1. Prof. Partha Pratim Das Programming in C++, IIT Kharagpur https://onlinecourses.nptel.ac.in/noc21_cs02/preview
2. Programming in C++: A Hands-on Introduction Specialization https://www.coursera.org/specializations/hands-on-cpp

Course Outcomes:

Student should be able to

1. Demonstrate understanding of object-oriented programming fundamentals and implement modular programs using classes and objects in C++.
2. Apply object-oriented concepts like constructors, destructors, and operator overloading to develop reusable components.
3. Implement inheritance, polymorphism, and virtual functions for designing extensible and maintainable code.
4. Use exception handling and file I/O mechanisms effectively to build robust and persistent C++ applications.

CO PO Map

CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	3	3	2		3				2	2		2	3	3
CO2	3	3	2		3							2	3	3
CO3	3	3	3		3							3	3	3
CO4	3	3	2	2	3					2		2	3	3

CO attainment levels

CO	Attainment level
CO .1	
CO .2	
CO .3	
CO .4	
CO .5	
CO .6	

Future Courses Mapping:

Algorithm Analysis, Database Systems, Operating Systems, Data structures Artificial Intelligence, Machine Learning, Computer Networks.

Job Mapping:

Software Developer / Software Engineer, Systems Programmer, Embedded Systems Engineer Game Developer, Backend Developer, Machine Learning Engineer, AI Developer, Database Developer, Cybersecurity Analyst

FF No. : 654

DS2006: Computer Network

Credits: 2 **Teaching Scheme Theory:** 1 Hour/Week
Lab: 2 Hours/Week

Course Prerequisites:

Basic networking concepts and skills

Course Objectives:

1. To introduce the fundamental concepts of switching, network topologies, and network architectures.
2. To provide understanding of the principles and differences between circuit-switched and packet-switched networks.
3. To familiarize students with DSL technologies and various physical-layer connecting devices used in networking.
4. To develop the ability to understand and evaluate the OSI and TCP/IP reference models along with different addressing types in computer networks.

Course Relevance:

This subject develops core competencies in object-oriented programming, equipping students with the ability to design modular, reusable, and efficient software solutions. Through mastery of C++ concepts like classes, inheritance, polymorphism, and templates, learners build a strong foundation for system-level programming, competitive coding, and real-world application development.

SECTION-1 Networking Fundamentals and Physical Layer (4 Hours)

Basics of Computer Networks: Switching (Circuit & Packet), Network Topologies architecture and components, Network types, OSI/TCP-IP models, addressing, Basic Protocol stack. Transmission media: Guided and Unguided Standard terminologies. Introduction to the Physical Layer and its role in the OSI model, Protocols.

SECTION-2 Data Link and Network Layer Protocols (6 Hours)

Data Link Layer protocols, MAC addressing, Error detection and correction techniques, Hop-to-hop delivery and framing, Routing (Static and Dynamic), Router configuration basics, IP Addressing, Network Layer Protocols: IPv4 and IPv6. Subnetting (Classful/Classless)

SECTION-3 Transport and Application Layers (4 Hours)

Transport Layer responsibilities, TCP vs UDP, Socket Programming basics using TCP/IP, Application Layer protocols and services, Session and Presentation layer concepts, Overview of well-known application protocols (HTTP, FTP, DNS, SMTP), DHCP and DNS server configuration using simulation tools.

List of Practical's: (Any 5)

1. **Design and Configure a Small Computer Lab using Switches and PCs**
   Case Study: A newly established college needs a basic computer lab setup for 20 systems. Design the lab using switches and PCs in Cisco Packet Tracer to ensure proper connectivity and communication among all systems.
2. **Design and Compare Star, Bus, Ring, and Mesh Topologies**
   Case Study: An IT consultant is helping three different organizations choose the best network topology. Design each topology using Cisco Packet Tracer and compare them based on scalability, cost, and fault tolerance. Recommend topologies for specific real-life scenarios such as schools, banks, and military networks.

3. **Enable Inter-LAN Communication using Routers and Static Routing**
   Case Study: A corporate office has three departments with isolated LANs. Configure routers and apply static routing to enable communication among these departments, ensuring structured and secure data flow.
4. **Subnet a Class C IPv4 Network and Allocate Subnets to Departments**
   Case Study: A company has received a Class C IP block and wants to segment it across four departments: HR, Finance, IT, and Support. Subnet the IP block accordingly using classful addressing and assign ranges to each department.
5. **Design and Simulate a Small Business Network with Departmental Segmentation**
   Case Study: A startup company requires an efficient and secure internal network. Design the network with logical segmentation for departments such as Admin, Sales, and Development, ensuring communication both within and across departments.
6. **Develop a TCP Client-Server Application using Python Sockets**
   Case Study: An internal communication system is needed for employees in a local network. Write and simulate a TCP-based client-server Python application to exchange text messages between users.
7. **Configure DHCP for Dynamic IP Assignment in a Multi-PC Network**
   Case Study: A school's IT lab administrator wants to eliminate manual IP assignments. Configure a DHCP server in Cisco Packet Tracer to assign IPs dynamically to 30 connected PCs and verify the setup.
8. **Implement VLANs and Inter-VLAN Routing using Layer 3 Switches**
   Case Study: A software company wants to separate department traffic for security. Implement VLANs for HR, Development, and Admin. Use a Layer 3 switch to configure Inter-VLAN routing for selective communication.
9. **Configure RIP Protocol for Dynamic Routing**
   Case Study: A retail chain with three branch offices needs to maintain up-to-date routing information. Configure the RIP protocol in Cisco Packet Tracer to allow dynamic routing between their networks and test for route updates.
10. **Configure Network Security Using Access Control Lists (ACLs)**
    Case Study: A university IT admin wants to block students from accessing staff resources. Use ACLs on routers to implement network access policies that restrict traffic based on IP addresses.

Suggest an assessment Scheme:
Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

CVV	GD/PPT	ESE(W)
30	30	40

CVV – Comprehensive Viva
GD/PPT – Group Discussion/PowerPoint Presentation
TH(O) – Online Examination

Reference Books: (As per IEEE format)

1. Andrew S. Tanenbaum ,"Computer Networks", Pearson, 1994, ISBN-13: 978-0-13-212695-3
2. Stallings William., "Data and Computer Communications", Sixth Edition, Prentice Hall of India, 2014, 10th edition, ISBN 978-0-133-50648-8

3. Fourouzan B., "Data Communications and Networking", 5th edition, McGraw- Hill Publications, Fifth Edition - 1 July 2017. ISBN-13: 978-1259064753 ISBN-10: 1259064751.
4. Atul Kahate, “Cryptography and Network Security”, McGraw Hill Publication, 2nd Edition, 2008, ISBN : 978-0-07-064823-4.

Moocs Links and additional reading material:

1. Introduction to Networking, Instructor: NVIDIA Training
https://www.coursera.org/learn/introduction-to-networking-nvidia
2. Computer Networks and Internet Protocol, by Prof. Soumya Kanti Ghosh, Prof. Sandip Chakraborty IIT Kharagpur https://onlinecourses.nptel.ac.in/noc22_cs19/preview

Course Outcomes:

Student should be able to

1. Describe basic concepts of switching, network topologies, and architectures.
2. Differentiate circuit-switched and packet-switched networks.
3. Identify DSL technologies and physical layer connecting devices.
4. Analyze OSI and TCP/IP models with addressing types in networks.

CO PO Map

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO:1	3				2				2	2		2	2	2
CO:2	3				2							2	2	2
CO:3	3	2	2		2							3	2	2
CO:4	3	3	2	2	2					2		2	2	2

CO attainment levels

CO	Attainment level
CO .1	
CO .2	
CO .3	
CO .4	
CO .5	
CO .6	

Future Courses Mapping:

Advance Computer Networks, Introduction to IoT.

Job Mapping:

Software Developer / Network Engineer, Network Programmer, Embedded Systems Engineer Game Developer, Backend Developer, Machine Learning Engineer, AI Developer, Database Developer, Cybersecurity Analyst

FF No.: 654

CD21235: Data Visualization

Credits: 3 **Teaching Scheme: Theory: 2** Hours/Week
Tut: 1 Hour/Week

Course Prerequisites:

Python programming, basic programming skills.

Course Objectives:

1. Understand the significance and impact of visual representation of data.
2. Recognize different data types (categorical, numerical, temporal, spatial) and apply appropriate visualization strategies.
3. Apply core principles of effective data graphics design, considering perceptual and cognitive aspects.
4. Select and implement suitable charts and graphs for diverse data contexts, avoiding common visualization pitfalls.
5. Gain hands-on experience with leading visualization tools such as Excel, Google Sheets, Tableau, and Python libraries (Matplotlib, Seaborn, Plotly).

Course Relevance:

By mastering visualization tools and principles, learners will be better prepared to work in multidisciplinary teams, contribute to data-driven strategies, and present findings effectively to both technical and non-technical audiences.

Section 1:
Introduction to Data Visualization (6 Hours)
Importance of data visualization, Data-to-visualization pipeline, Types of data: categorical, numerical, temporal, spatial, Data graphics principles (Tufte, Few, etc.) Perception and cognitive aspects in visualization: Case studies and examples
Charts and Graphs for Different Data Types (6 Hours)
Bar charts, pie charts, histograms, line plots, area charts, Scatter plots, bubble charts, box plots, Time series, hierarchical (tree maps, sunbursts), network and spatial visualizations, Dos and Don'ts of chart selection, Choosing the right chart for the story.
Section 2:
Tools, Frameworks & Coding for Visualization (6 Hours)
Spreadsheet tools (Excel, Google Sheets), Python libraries: Matplotlib, Seaborn, Plotly, Tableau: Basic layout, measures and dimensions, integrating charts with dashboards, Exporting and sharing visualizations
Interactive Dashboards and Storytelling with Visuals (6 Hours)

Designing and building dashboards (Tableau, Power BI, Streamlit basics), Interactive elements (filters, tooltips, drill-downs), Visual storytelling structuring narratives, Case studies – business, healthcare, and academic datasets, Evaluation of visualization impact

List of Tutorials:

1. Understand how to visualize a dataset using various chart types and derive insights.
2. Learn how to create a small dashboard using interactive visual elements.
3. Learn how different Python visualization library's function.
4. Design a dashboard that tells a compelling story.
5. Explore how GenAI can assist in automating dashboard generation.
6. End-to-end use of GenAI to build and narrate a data-driven story.
7. Compare patterns and trends between two different geographical areas.
8. Analyze and visualize how data changes over time.
9. Transform raw data into a visually appealing infographic.
10. Use AI tools to auto-generate insights and validate them manually.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

HA	T1(O)	T2(O)
30	35	35

HA - Home Assignment
T1(O)- Test 1 online mode
T2(O)- Test 2 online mode

Text Books: (As per IEEE format)

1. K. Wexler and N. B. Robbins, **The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios**, 1st ed. Hoboken, NJ, USA: Wiley, 2017.

Reference Books: (As per IEEE format)

1. D. Sharda, D. Delen, and E. Turban, **Business Intelligence, Analytics, and Data Science: A Managerial Perspective**, 5th ed. Pearson, 2020.
2. P. M. Joshi and P. N. Mahalle, **Data Storytelling and Visualization with Tableau: A Hands-On Approach**, 1st ed. Boca Raton, FL, USA: CRC Press, 2022.2.

Moocs Links and additional reading material:

1. J. Heer; 'Data Visualization'; Coursera – University of Washington https://www.coursera.org/learn/datavisualization
2. J. Brooks; 'Data Visualization with Python'; IBM via Coursera https://www.coursera.org/learn/python-for-data-visualization
3. K. Vohra; 'Data Visualization with Tableau'; University of California, Davis via Coursera https://www.coursera.org/learn/data-visualization-tableau

Course Outcomes:

Students should be able to

1. Explore the fundamentals of data visualization and its role in data-driven decision making.
2. Select and design appropriate charts for various types of datasets.
3. Implement data visualizations using tools and programming frameworks.
4. Develop interactive dashboards and communicate data stories effectively.
5. Analyze case studies to evaluate the impact of different visualization approaches.

CO PO Map

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO1	PSO2
CO:1	3	2	1		1	2	1			1	2	2	1	3
CO:2	3	3	2		3						2	2	2	3
CO:3	2	2	3	1	3	1		1	1	1	2	3	2	2
CO:4	2	2	3	2	3	1	1	2	3	3	3	2	2	2
CO:5	2	3	3	3	2	2	1	1	2	2	3	1	3	2

CO attainment levels

CO	Attainment level
CO .1	
CO .2	
CO .3	
CO .4	
CO .5	

Future Courses Mapping:

This course lays a strong foundation for advanced subjects like Machine Learning, Data Analytics, and Big Data. The skills in visual storytelling, dashboarding, and data interpretation directly support future courses such as AI, Data Mining, and Capstone Projects. It also enhances understanding in HCI and Web Technologies through interactive visual design concepts.

Job Mapping:

This course equips students with skills relevant to roles such as Data Analyst, Business Intelligence Developer, Data Visualization Specialist, and Dashboard Designer. Proficiency in tools like Tableau, Excel, Python (Matplotlib, Seaborn, Plotly), and Power BI aligns with industry needs in sectors like IT, finance, healthcare, and e-commerce.

FF No.: 654

HS2002: From Campus To Corporate – 1

Credits:.2 **Teaching Scheme: Lab: 2 hours/Week**

Introduction to the Corporate World Understanding organizational structure and hierarchy, Work culture differences: campus vs. corporate,
Employer expectations from fresh graduates, Time management and ownership in corporate settings

Professional Communication Skills: Verbal and non-verbal communication, Email and business writing etiquette, Presentation skills and use of visual aids, Listening skills and telephone etiquette,
Soft Skills and Interpersonal Effectiveness: Body language, grooming, and first impressions, Conflict resolution and negotiation skills, Team dynamics and collaboration, Assertiveness vs. aggressiveness
Resume Building and Job Preparation : Building an effective resume and cover letter, Identifying strengths and achievements, Preparing for technical and HR interviews, Handling rejections and feedback
Group Discussions and Personal Interviews : Group discussion formats and evaluation criteria, Strategies for initiating, contributing, and summarizing, Mock interviews with feedback, STAR technique for answering behavioral questions,
Corporate Etiquette and Workplace Ethics: Meeting and greeting protocol, Dining and social etiquette, Work ethics, punctuality, confidentiality, Respect for diversity and inclusion in the workplace
Adaptability and Emotional Intelligence: Handling pressure, deadlines, and ambiguity, Self-awareness and emotional regulation, Empathy and workplace relationships, Managing feedback and continuous learning,
Introduction to Project Management Basics: Understanding tasks, milestones, deadlines, Collaboration using tools like Trello, Slack, Teams, Basics of Agile/Scrum concepts, Reporting and escalation protocol

Faculty are supposed to do conduct following in the class

- Resume and LinkedIn profile workshops
- Mock interviews and GD sessions
- Role plays: workplace scenarios, conflict handling
- Business email writing exercises
- Presentation and elevator pitch sessions

Books:

1. Dale Carnegie, How to Win Friends and Influence People
2. Stephen R. Covey, 7 Habits of Highly Effective People
3. Shital Kakkar Mehra, Business Etiquette: A Guide for the Indian Professional
4. Peggy Klaus, The Hard Truth About Soft Skills

FF No. : 654

HS2001: Reasoning And Aptitude Development - 3

Credits:.1 **Teaching Scheme: 1 Hour/Week**

Unit 1: English Language

Familiarity with English Language, Ability to understand written text, spoken word and effective communication through written documents; Coverage of vocabulary to cope up with general and specific terminology, syntax and sentence structure, prevention of incorrect use leading to distortion in communication; synonyms, antonyms and contextual vocabulary, Grammar – Error identification, sentence improvement and construction, Reading Comprehension

Unit 2: Logical Ability

Objective interpretation of things, ability to perceive and interpret trends to make generalizations; ability to analyze assumptions behind an argument or statement; Deductive reasoning: Assessment of ability to synthesize information and derive conclusions - Coding deduction logic, Data Sufficiency, Directional Sense, Logical word sequence, Objective reasoning, Selection and decision tables, puzzles; Inductive reasoning: Assessment of ability to learn by example, imitation or by trial – Analogy pattern recognition, Classification pattern recognition, Coding pattern recognition, Number series pattern recognition; Abductive reasoning: Critical thinking ability of seeing through logical weak links or loopholes in an argument or a group of statements; Critical reasoning: assessment of ability to think through and analyze logical arguments, assessment of ability to use logical constructs to offer reasoning in unfamiliar situations; Information Gathering and synthesis: Ability of locating information, information ordering, rule based selection and data interpretation, order and classify data, interpret graphs, charts, tables and make rule based deductions. Application of these approaches for using visual, numerical and textual data from single or multiple sources

Unit 3: Quantitative Ability

Basic numbers – decimals and fractions, factorization, divisibility: HCF, LCM, Odd, even, prime and rational numbers. Application of algebra to real world, direct and inverse proportion, common applications – Speed-time -distance, Profit-loss, percentage, age relations, mixtures, other miscellaneous quantitative combination, exponentials and logarithms, permutations and combinations, probability. Spatial reasoning: Inductive – Missing portions, Sequence and series; Deductive analysis.

Reference Books –

1. "English Grammar in Use" by Raymond Murphy, Cambridge University Press.
2. "Word Power Made Easy" by Norman Lewis, Goyal Publishers & Distributors.
3. "Objective General English" by S.P. Bakshi, Arihant Publications.
4. "English for Competitive Examinations" by K. Sinha, S. Chand Publishing.
5. "Essential English Grammar" by Philip Gucker, Wiley.
6. "English Idioms and Phrasal Verbs" by M.A. Yadav, Vikas Publishing House.
7. "The Oxford English Grammar" by Sidney Greenbaum, Oxford University Press.
8. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121903409.
9. "Logical Reasoning and Data Interpretation for the CAT" by Nishit K. Sinha, Pearson India, ISBN: 978-8131709117.
10. "Logical Reasoning and Data Interpretation for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
11. "A New Approach to Reasoning Verbal and Non-Verbal" by B.S. Sijwali & Indu Sijwali, Arihant Publications, ISBN: 978-9311124692.

12. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121900637.
13. "How to Prepare for Quantitative Aptitude for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
14. "The Pearson Guide to Quantitative Aptitude for Competitive Examination" by Pearson, Pearson India, ISBN: 978-8131709117.
15. "Quantitative Aptitude for Competitive Examinations" by Abhijit Guha, Tata McGraw Hill Education, ISBN: 978-0070666653.
16. "Data Interpretation & Data Sufficiency" by R.S. Aggarwal, S. Chand Publishing ISBN: 978-8121903515.
17. "Quantitative Aptitude for Competitive Examinations" by S. Chand, S. Chand Publishing, ISBN: 978-8121903423.

Course Outcomes:

Upon completion of the course, the student will be able to –

1. Improve the reading, writing and verbal skills, and enhance comprehension and articulation abilities
2. Develop logical reasoning abilities, enabling them to make sound decisions in problem-solving scenarios
3. Develop mathematical aptitude as well as data interpretation abilities and use them in test cases and real world problems
4. Learn to apply approaches for optimum time-management, prioritization maximizing the accuracy
5. Learn data interpretation, apply mathematical skills to draw accurate conclusions
6. Apply their knowledge of English, reasoning and quantitative skills for planning, critical thinking and real world problems

CO-PO Map

	Program Outcomes (PO)												**PSO**			
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		3	3
CO6	2	2	3	3	2				3		3	2	3		3	3
Average	**2.0**	**2.0**	**2.83**	**2.83**	**2.6**	**2.5**	**2.0**	**2.0**	**3.0**	**1.0**	**2.16**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

FF No. : 654

DS2001: Design Thinking – 3

Credits: 1 **Teaching Scheme: Tutorial 01 Hour/week**

Course Prerequisites:

Problem Based Learning, Project Centric Learning

Course Objective:

To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship

Section 2: Topics/Contents

Structure of The paper
Journal List (Top 50 Journals)
Selection of the journal
Use of various online journal selection tools
Plagiarism checking
Improving contents of the paper
Patent drafting
Patent search
Filing of patent
Writing answers to reviewer questions
Modification in manuscript
Checking of publication draft

Course Outcomes:

[Publication of paper or patent]
The student will be able to
1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

	Program Outcomes (PO)												PSO			
CO/PO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	3
CO2	1	1	1	1	1	--	--	--	--	--	--	1	2	1	1	3
CO3	2	2	3	3	2	2	1	2	2	3	--	1	2	2	3	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	-	-	2	3
CO5	1	1	1	1	1	--	--	--	--	--	--	1	-	-	1	2
CO6	2	2	2	2	2	2	1	3	2	3	--	1	2	2	2	3
C07	1	1	1	1	1	--	--	--	--	--	--	1	1	1	1	1
Average	**1.57**	**1.57**	**1.71**	**1.71**	**1.57**	**2.0**	**1.0**	**2.33**	**2.0**	**3.0**	**1.0**	**1.0**	1.66	1.66	1.71	2.5

CO-PO Map:

FF No. : 654

DS2002: Engineering Design and Innovation – III

Credits:2 **Teaching Scheme: 4 Hours/Week**

Course Prerequisites:

Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Course Relevance:

Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project centric learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:
Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.
- It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution.
  How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- · How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDD Sample Case Studies : -

With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects can be taken up:

1) Design a deployable product for soil moisture detection
2) Design a deployable product for temperature detection
3) Design a deployable product for pressure detection
3) Design a deployable product smoke detection
4) Design a deployable product for motion detection
5) Design a deployable product for collision detection
6) Design a deployable product for sound detection

...not limited to.....Faculty and students are free to include other areas which meet the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

To focus on the higher levels of the Booms Taxonomy analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1.A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2.Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: ***(As per IEEE format)***

1.De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.

2.Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

MOOCs Links and additional reading material: www.nptelvideos.in

https://worldwide.espacenet.com/

Course Outcomes:

On completion of the course, learner will be able to–

1. Identify the real life problem from societal need point of view
2. Choose and compare alternative approaches to select most feasible one
3. Analyze and synthesize the identified problem from technological perspective
4. Design the reliable and scalable solution to meet challenges
5. Evaluate the solution based on the criteria specified
6. Inculcate long life learning attitude towards the societal problems

CO PO Map

	Program Outcomes (PO)												**PSO**			
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		3	3
CO6	2	2	3	3	2				3		3	2	3		3	3
Average	**2.0**	**2.0**	**2.83**	**2.83**	**2.6**	**2.5**	**2.0**	**2.0**	**3.0**	**1.0**	**2.16**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3