



**Nagarjuna College of Engineering &
Technology, Bengaluru**
An Autonomous Institute, Affiliated to VTU Belagavi

2024 Batch

Scheme & Syllabus
of
IV Sem

As per the NEP 2020 Guidelines,
Choice-Based Credit System
&
Outcome-Based Education

CSE (Data Science)

Academic Year 2025-2026

Vision:

To build a strong technical environment and foster leadership and problem-solving abilities in the domain of Data Science, creating professionals capable of addressing social and technical challenges.

Mission:

1. To equip and expose students with the latest tools and technologies.
2. To instill critical problem-solving capabilities, leadership qualities, research capabilities and to prepare them for global challenges.
3. To establish state-of-the-art laboratories and foster collaborations with leading industries in the field of Data Science.

PROGRAM OUTCOMES (POs): Graduates of the Computer Science and Engineering – Data Science

Program will be able to achieve the following

- PO1: Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in Washington Accord Knowledge 1 (WK1) to Washington Accord Knowledge 4 (WK4) respectively to develop the solution of complex engineering problems.
- PO2: Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
- PO3: Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
- PO4: Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8)
- PO5: Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
- PO6: The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
- PO7: Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
- PO8: Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
- PO9: Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for

- i) Independent and life-long learning.
- ii) Adaptability to new and emerging technologies and
- iii) Critical thinking in the broadest context of technological change. (WK8)

Program Specific Outcome (PSO)

PSO1: Analyze complex computing problems and apply to derive appropriate solutions.

PSO2: Design, implement, and evaluate database-oriented, computing-based solutions that address a broad range of requirements in the field of Data Science.

PSO3: Communicate and work effectively within diverse teams and professional environments.

Program Educational Objectives (PEOs)

PEO1: To work as Data Scientist with an ability to solve wide range of computational problems.

PEO2: To work effectively in a diverse and multi-disciplinary field, as a team member or leader to solve the societal problems.

PEO3: Engage in self-directed and lifelong learning, continuously updating their skills by adapting emerging techniques, advancing in research and higher studies.

Nagarjuna College of Engineering and Technology, Bangalore

B.E. in CSE (Data Science)

Scheme of Teaching and Examinations 2023

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

(Effective from the academic year 2025-26)

IV SEMESTER													
Sl. No	Course and Course Code		Course Title	Teaching (TD) Department and Question Paper Setting Board (PSB)	Teaching Hours / Week				Examination				Credits
					Theory Lectur	Tutorial	Practical /Drawing	Self -Study	Duration in hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	IPCC	24CDI41	Data Wrangling using Python	TD and PSB	3	0	2		03	50	50	100	4
2	IPCC	24CDI42	Analysis and Design of Algorithms	TD and PSB	3	0	2		03	50	50	100	4
3	IPCC	24CDI43	Object-Oriented Programming Using Java	TD and PSB	3	0	2		03	50	50	100	4
4	PCCL	24CDL44	Web Technology based Mini project	TD and PSB	0	0	2		03	50	50	100	1
5	ESC	24CDT45x	ESC/ETC/PLC		3	0	0		03	50	50	100	3
6	AEC/ SEC	24CDL46x	Ability Enhancement Course/Skill Enhancement Course- IV	TD and PSB: Concerned department	If the course is Theory				01	50	50	100	1
					1	0	0						
					If the course is a lab				02				
					0	0	2						
7	BSC	24BET47	Bio-Engineering and Innovations	TD / PSB: Any Branch/ BT, CHE,	2	0	0		03	50	50	100	2
8	UHV	24UHV48	Universal human values	Any Department	1	0	0		01	50	50	100	1
9	MC	24NS49	National Service Scheme (NSS)	NSS coordinator	0	0	2			100	---	100	0
		24PE49	Physical Education (PE) (Sports and Athletics)	Physical Education Director									
		24YO49	Yoga	Yoga Teacher									
Total									500	400	900	20	

PCC: Professional Core Course, **PCCL:** Professional Core Course laboratory, **UHV:** Universal Human Value Course, **MC:** Mandatory Course (Non-credit), **AEC:** Ability Enhancement Course, **SEC:** Skill Enhancement Course, **L:** Lecture, **T:** Tutorial, **P:** Practical **S= SDA:** Skill Development Activity, **CIE:** Continuous Internal Evaluation, **SEE:** Semester End Evaluation. **K :** This letter in the course code indicates common to all the stream of engineering.

Ability Enhancement Course / Skill Enhancement Course - IV

24CDL461	Git and GitHub	24CDL462	Object Oriented Modelling lab using Star UIML
----------	----------------	----------	---

Engineering Science Course (ESC/ETC/PLC)

24CDT451	Discrete Mathematical Structures	24CDT452	Graph Theory
----------	----------------------------------	----------	--------------

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching– Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering /Technology (B.E./B.Tech.) 2022-23

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the courses is mandatory for the award of degree.

DATA WRANGLING USING PYTHON			
Course Code	24CDI41	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 T + 13 Lab Slots	Total Marks	100
Credits	04	Exam Hours	03
Prerequisites: Python Programming.			
Course Learning Objectives			
<ul style="list-style-type: none"> • To learn different methods for Data Cleanup • To learn basics of Python. • To learn Data aggregation Techniques. • To learn about packages Numpy and Pandas • To learn concepts of Web scraping. 			
Teaching-Learning Process (General Instructions)			
<p>These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. Lecturer method (L) need not to be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes. 2. Use of Video/Animation to explain functioning of various concepts. 3. Encourage collaborative (Group Learning) Learning in the class. 4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking. 5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it. 6. Introduce Topics in manifold representations. 7. Show the different ways to solve the same problem with different circuits/logic and encourage the students to come up with their own creative ways to solve them. 8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding. 			
Module – I			
Introduction to Data Wrangling and Python Programming			
Overview of Data Wrangling – Definition, importance, and real-world applications			
Python Basics – Syntax, data types, control structures, and functions			
Essential Libraries – Pandas, NumPy, and Matplotlib			
Setting Up the Environment – Installing Python, Jupyter, and relevant packages			
Hands-on Exercises – Writing basic Python scripts and working with datasets			
08 Hours			
Module – II			
Data Acquisition and Cleaning			
Data Sources – Importing CSV, JSON, Excel, and databases			
Handling Missing and Duplicate Data – Identification, removal, and imputation techniques			
Data Type Conversions & Parsing – Ensuring consistency and correctness			
String Operations and Data Cleaning – Formatting textual data efficiently			
08 Hours			
Module – III			

Data Transformation and Feature Engineering Data Reshaping – Merging, joining, pivoting, and melting datasets Feature Scaling & Encoding – Normalization, standardization, and encoding Time Series Data Handling – Working with date/time features AI in Feature Engineering – Using AI for automated feature selection <p style="text-align: right;">08 Hours</p>	
Module – IV	
Exploratory Data Analysis (EDA) and Visualization Descriptive Statistics – Summary statistics and data distributions Data Visualization – Matplotlib, Seaborn, and interactive plots Identifying Patterns and Outliers – Statistical and visual techniques AI-Assisted EDA – Using AI tools to uncover insights <p style="text-align: right;">08 Hours</p>	
Module – V	
Advanced Topics and Capstone Project Automation in Data Wrangling – Creating reusable pipelines Big Data Tools – Introduction to PySpark and Trifacta (Alteryx) Ethics in AI and Data Handling – Bias, privacy, and best practices Capstone Project – End-to-end project on real-world data <p style="text-align: right;">08 Hours</p>	
Teaching-Learning Process for all modules	Chalk and board, Active Learning, PPT Based presentation, Video
PRACTICAL COMPONENTS	
Sl. No	Experiments
1	Write a Python script to read each row from a given csv file and print a list of strings.
2	Write a Python program to read a given CSV file as a dictionary.
3	Write a Python program to convert Python dictionary object (sort by key) to JSON data. Print the object members with indent level 4
4	Write the python script to Read the XML file
5	Write a Pandas program to import excel data into a Pandas data frame and process the following a. Get the data types of the given excel data b. Display the last ten rows. c. Insert a column in the sixth position of the said excel sheet and fill it with NaN values
6	Develop the python script to parse the pdf files using pdfminer.
7	Write a Python data wrangling scripts to insert the data into SQLite database
8	Develop the Python Shell Script to do the basic data cleanup on excel data to a. Check duplicates and missing data b. Eliminate Mismatches c. Cleans line breaks, spaces, and special characters
9	Develop a python program to parse data representing information about individuals. df1 contains the columns 'ID' and 'Name', while df2 contains the columns 'ID' and 'Age'. Perform the following joins and describe the resulting DataFrames: a. inner join b. left join

	c. right join d. outer join
10.	Develop a python program to use the merge operation in order to merge the data and provide it meaning for the following scenario A teacher has two types of Data, the first type of Data consists of Details of Students and the Second type of Data Consist of Pending Fees Status which is taken from the Account Office.
11	Write a python script for the following scenario There is a Car Selling company and this company have different Brands of various Car Manufacturing Company like Maruti, Toyota, Mahindra, Ford, etc., and have data on where different cars are sold in different years. So the Company wants to wrangle only that data where cars are sold during the year 2010.(Grouping)

Course Outcomes:

On completion of this course, the students will be able to,

- CO1:** Identify and execute the basic data format.
- CO2:** Analyze the computations with Excel and pdf files.
- CO3:** Apply the concepts of data cleanup.
- CO4:** Analyze and explore the Image and video data.
- CO5:** Apply the concepts of web scraping.

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	Total 25 Marks : Reduced to 15 Marks	
	CCE-1	25 Marks
	CCE-2	25 Marks
	Average of two CCEs	25 Marks
	Total 25 Marks : Reduced to 10 Marks	
Lab Component	Lab Record and execution of programs	15 Marks
	Lab Test at the end of 15th week	10 Marks
	Total	25 Marks
Grand Total of IAT Marks		50 Marks
Obtaining 40% of marks in both theory and lab component is essential for appearing for SEE		

Text Books:

1. Jacqueline Kazil & Katharine Jarmul,|| Data Wrangling with Python||, O'Reilly Media, Inc,2016.

References:

1. Dr. Tirthajyoti Sarkar, Shubhadeep,|| Data Wrangling with Python: Creating actionable data from raw sources||, Packt Publishing Ltd,2019.
2. Stefanie Molin,|| Hands-On Data Analysis with Pandas||, Packt Publishing Ltd,2019.
3. Allan Visochek,|| Practical Data Wrangling||, Packt Publishing Ltd,2017.
4. Tye Rattenbury, Joseph M. Hellerstein, Jeffrey Heer, Sean Kandel, Connor Carreras,|| Principles of Data Wrangling: Practical Techniques for Data Preparation||, O'Reilly Media, Inc, 2017.

E-Books:

1. <http://www.gbv.de/dms/ilmenau/toc/827365454.PDF>

E-Resources:

1. <https://www.kaggle.com/datasets>

MOOC:

1. <https://www.udemy.com/course/data-wrangling-with-python/>
2. <http://www.openculture.com/free-online-data-science-courses>
3. <https://www.classcentral.com/course/dataanalysiswithpython-11177>

ANALYSIS AND DESIGN OF ALGORITHMS

Course Code	24CDI42	CIE Marks	50
Teaching Hours/Week (L: T: P: S) (3:0:2:0)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 T + 13 Lab Slots	Total Marks	100
Credits	04	Exam Hours	03

Course Objectives:

1. To learn the methods for analyzing algorithms and evaluating their performance.
2. To demonstrate the efficiency of algorithms using asymptotic notations.
3. To solve problems using various algorithm design methods, including brute force, greedy, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking, and branch and bound.
4. To learn the concepts of P and NP complexity classes.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) does not mean only the traditional lecture method, but different types of teaching methods may be adopted to achieve the outcomes.
2. Utilize video/animation films to illustrate the functioning of various concepts.
3. Promote collaborative learning (Group Learning) in the class.
4. Pose at least three HOT (Higher Order Thinking) questions in the class to stimulate critical thinking.
5. Incorporate Problem-Based Learning (PBL) to foster students' analytical skills and develop their ability to evaluate, generalize, and analyze information rather than merely recalling it.
6. Introduce topics through multiple representations.
7. Demonstrate various ways to solve the same problem and encourage students to devise their own creative solutions.
8. Discuss the real-world applications of every concept to enhance students' comprehension.

Module – I

INTRODUCTION: What is an Algorithm? Fundamentals of Algorithmic Problem Solving.

FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non recursive Algorithms, Mathematical Analysis of Recursive Algorithms.

BRUTE FORCE AND EXHAUSTIVE SEARCH: Selection Sort, Sequential Search, Brute Force String matching, Exhaustive Search (Travelling Salesman problem).

Chapter 1 (Sections 1.1, 1.2), Chapter 2 (Sections 2.1, 2.2, 2.3, 2.4), Chapter 3 (Section 3.1, 3.2, 3.4)

08 Hours

Module – II

BRUTE FORCE AND EXHAUSTIVE SEARCH: (contd.): Depth First Search and Breadth First Search.

DECREASE AND CONQUER: Insertion Sort, Topological Sorting, Binary Search.

DIVIDE AND CONQUER: Merge Sort, Quick Sort.

Chapter 3 (Section 3.5), Chapter 4 (Sections 4.1, 4.2, 4.4), Chapter 5 (Section 5.1, 5.2)

08 Hours

Module – III	
<p>TRANSFORM AND CONQUER: Balanced Search Trees- AVL trees, 2-3 trees, Heaps and Heapsort. SPACE-TIME TRADEOFFS: Sorting by Counting, Input Enhancement in String Matching: Horspool's Algorithm. Chapter 6 (Sections 6.3, 6.4), Chapter 7 (Sections 7.1, 7.2)</p>	
08 Hours	
Module – IV	
<p>DYNAMIC PROGRAMMING: Three basic examples, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. THE GREEDY METHOD: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes. Chapter 8 (Sections 8.1, 8.2, 8.4), Chapter 9 (Sections 9.1, 9.2, 9.3, 9.4)</p>	
08 Hours	
Module – V	
<p>LIMITATIONS OF ALGORITHMIC POWER: Decision Trees, P, NP, and NP-Complete Problems. COPING WITH LIMITATIONS OF ALGORITHMIC POWER: Backtracking (n-Queens problem, Subset-sum problem), Branch-and-Bound (Knapsack problem, Assignment problem, Travelling Salesman Problem). Chapter 11 (Section 11.2, 11.3), Chapter 12 (Sections 12.1, 12.2)</p>	
08 Hours	
PRACTICAL COMPONENTS	
Sl. No	LIST OF LABORATORY PROGRAMS
1	Design and implement C Program to print all the nodes reachable from a given starting node in a digraph using BFS method.
2	Design and implement C Program to sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of $n > 5000$ and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3	Design and implement C Program to sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of $n > 5000$, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
4	Design and implement C Program to solve All-Pairs Shortest Paths problem using Floyd's algorithm .
5	Design and implement C Program to solve 0/1 Knapsack problem using Dynamic Programming method.
6	Design and implement C Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm .
7	Design and implement C Program to find shortest paths from a given vertex in a weighted connected graph to other vertices using Dijkstra's algorithm .
8	Design and implement C Program for N Queen's problem using Backtracking .
Course outcome (Course Skill Set)	
<p>At the end of the course, the student will be able to:</p> <ol style="list-style-type: none"> 1. Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity. 2. Demonstrate divide & conquer approaches, decrease & conquer approaches to solve 	

computational problems.

3. Make use of transform & conquer and input enhancement methods design approaches to solve the given real world or complex computational problems.
4. Apply greedy and dynamic programming to solve graph & string based computational problems.
5. Illustrate backtracking, branch & bound and approximation methods.

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	Total 25 Marks : Reduced to 15 Marks	
	CCE-1	10 Marks
	CCE-2	10 Marks
	Average of two CCEs	10 Marks
	Total 10 Marks : Reduced to 10 Marks	
Lab Component	Lab Record and execution of programs	15 Marks
	Lab Test at the end of 15th week	10 Marks
	Total	25 Marks
Grand Total of IAT Marks		50 Marks
Obtaining 40% of marks in both theory and lab component is essential for appearing for SEE		

Suggested Learning Resources:

Text Books:

1. Anany Levitin: "Introduction to The Design and Analysis of Algorithms", Pearson Education, Delhi, 3rd Edition, 2007, ISBN-13 : 9780132316811.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI, London, England, 3rd Edition, 2010, ISBN-13: 9780262033848.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", (Chapters 1,3-8,10-12), Universities Press, Hyderabad, 2nd Edition, 2007, ISBN-10: 8173716129.
3. R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: "Introduction to the Design and Analysis of Algorithms A Strategic Approach", McGraw-Hill Higher Education, USA, 3rd International Edition, 2005, ISBN-13: 978- 0071243469.

Reference Online Resources:

1. <https://www.geeksforgeeks.org/fundamentals-of-algorithms/>
2. <http://www.citc.ui.ac.ir/ze/moni/c ls.pdf>

POS COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	1	-	1	-
CO2	-	3	-	-	-	-	-	-	-	-	2	2	2	2
CO3	-	3	-	-	-	-	-	-	-	-	2	2	2	2
CO4	-	-	3	-	-	-	-	-	-	-	2	2	2	2
CO5	-	-	-	2	-	-	-	-	-	-	2	2	2	2

OBJECT-ORIENTED PROGRAMMING USING JAVA

Course Code	24CDI43	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 T + 13 Lab Slots	Total Marks	100
Credits	04	Exam Hours	03
Pre-Requisites: Basic knowledge of programming concepts, including variables, datatypes, operators, conditional statements, loops, arrays.			
Course Objectives: <ol style="list-style-type: none"> 1. Understand the basic concepts of object-oriented programming. 2. Learn the concepts of Java programming using classes and objects. 3. Gain the knowledge of applying Inheritance and Interfaces for implementing real world problems. 4. Expose to the concepts of Packages and Exceptions while implementing solutions for complex problems. 5. Acquire the knowledge of multi-threaded programming and string handling in Java. 			
Teaching-Learning Process (General Instructions) <p>These are sample Strategies, used in OOP with Java to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. Chalk and talk. 2. Every concept can be related to the real-world examples, is taught in the class which helps improve the students' understanding. 3. Practical oriented with hands on session. 4. Analyse the code snippet with different variations in the class with respect to the topics to know their understanding level and which also promotes critical thinking. 			
Module – I			
Introduction to Object Oriented Concepts: Procedure Oriented Programming, Object-Oriented Programming, Comparison of Object-Oriented Language with C. Introduction to Java, Java Buzzwords, The Byte code, Java Development Kit (JDK), Data types, Variables and Arrays, Operators, Control statements, Simple Java programs. <p style="text-align: right;">08 Hours</p>			
Module – II			
Classes, Methods: Classes fundamentals, Declaring Objects, this keyword, Garbage Collection. Methods: Method Prototyping, Member Functions and Data Members, Constructors, Objects and Methods, Method Overloading, Access specifiers and modifiers. <p style="text-align: right;">08 Hours</p>			
Module – III			
Inheritance: Inheritance basics, using super, creating Multi-Level hierarchy, Method Overriding, using Abstract classes, using final. Interfaces: Defining an Interface, Implementing an Interface. <p style="text-align: right;">08 Hours</p>			
Module – IV			
Packages: Access Protection, Importing Packages. Exceptions: Exception handling fundamentals, Exception types, using try and catch, using multiple catch clauses, nested try statements, throw, throws, finally, Java's built-in exceptions. <p style="text-align: right;">08 Hours</p>			

Module - V	
<p>Multi-Threaded Programming: What are threads? How to make the classes threadable, Extending threads, Implementing runnable, Synchronization.</p> <p>String Handling: String Constructors, String Operations, Character Extraction, String Comparison.</p> <p style="text-align: right;">08 Hours</p>	
Teaching-Learning Process	Chalk and board, Active Learning, Demonstration
LIST OF EXPERIMENTS	
1.	Develop a Java program for an advanced arithmetic calculator that takes two integer operands and an operator from the user. The program should be capable of performing addition, subtraction, multiplication, and division. Ensure that the program handles input validation, including checking for the validity of the operator and operands. After each calculation, ask the user if they want to perform another operation and provide a history of previous calculations upon request.
2.	Write a Java program to generate the first 'n' terms of the Fibonacci series and determine the following: <ol style="list-style-type: none"> 1. Calculate the sum of all even terms in the series. 2. Find the product of all odd terms in the series. 3. Find the sum of all terms in the series. 4. Calculate the average of the entire series.
3.	Develop a Java program showcasing method overloading with a class "Phone" containing the three dial() method with different parameters, The program should demonstrate and print the result.
4.	Develop a Java program illustrating constructor overloading for calculating the area of a rectangle and a circle using appropriate constructors.
5.	Create a Java program with a vehicle hierarchy, including Vehicle, Car, SportsCar, and Truck classes. Implement methods for starting and stopping in the base class and specialized methods for accelerating, adding turbo boost, and loading cargo in the subclasses, with appropriate method overrides.
6.	Create a Java program that models electronic devices (e.g., smartphones, laptops, and tablets) using a common interface for power management. The program should allow users to interact with the devices and control their power state.
7.	Develop a Java program that emulates a library system. Create two packages, `library` and `patron`. In the `library` package, define a `Book` class with a private title field. In the `patron` package, implement a `Patron` class that can borrow books. Demonstrate the use of packages, access protection, and class imports. Ensure that the book title remains inaccessible from outside the `library` package due to the `private` access modifier. Create a scenario where a patron, Alice, borrows a book from the library.
8.	Develop a Java lab program that handles exceptions for division by zero and invalid input. Use `try-catch` blocks to catch `Arithmetic Exception` for division by zero and `Input Mismatch Exception` for non-integer input and provide user-friendly error messages.
9.	Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number.
10.	Design a Java lab program to demonstrate string handling, including creating strings using constructors and literals, concatenating strings, extracting characters at a specified index, and comparing strings for equality.
<p>Course Outcomes: The student will be able to : Define and explain the concepts of object-oriented programming. CO1: Explain the fundamental concepts of object-oriented programming.</p>	

- CO2:** Apply the knowledge of object-oriented programming in solving real time problems.
CO3: Analyze the problem and implement solutions for problems.
CO4: Develop solutions using different features of object-oriented programming.
CO5: Demonstrate the programs on object oriented programming concepts in Java using different tools

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	Total 25 Marks : Reduced to 15 Marks	
	CCE-1	10 Marks
	CCE-2	10 Marks
	Average of two CCEs	10 Marks
	Total 10 Marks : Reduced to 10 Marks	
Lab Component	Lab Record and execution of programs	15 Marks
	Lab Test at the end of 15th week	10 Marks
	Total	25 Marks
Grand Total of IAT Marks		50 Marks
Obtaining 40% of marks in both theory and lab component is essential for appearing for SEE		

Text books

- Herbert Schildt, "Java The Complete Reference", 7th Edition, TataMcGrawHill, 2013, ISBN 13:978-0072263855, (Chapters 1-11).

Reference Books:

- E Balagurusamy, "Programming with Java-A primer", 2nd Edition, TataMcGraw Hillcompanies, 2009, ISBN-13:978-9351343202.

E - Resources:

- www.geeksforgeeks.org/java/
- www.tutorialspoint.com/java/

CO-PO-PSO Mapping

PO'S	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO1 1	PSO 1	PSO 2	PSO 3
CO1	-	-	-	-	-	-	-	-	1	-	-	2	1	-
CO2	3	-	-	-	-	-	-	-	-	-	-	2	1	-
CO3	-	3	-	-	-	-	-	-	-	-	-	2	1	-
CO4	-	-	2	-	-	-	-	-	-	-	-	2	1	-
CO5	-	-	2		3	-	-	-	-	-	-	2	1	-
AVG	3	3	2		3	-	-	-	-	-	-	2	1	

DISCRETE MATHEMATICAL STRUCTURES

Course Code	24CDT451	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	2:2:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Course objectives:

This course will enable students to:

- Understand an intense foundational introduction to fundamental concepts in discrete mathematics.
- Interpret and solve the language associated with logical reasoning, relations, and functions.
- In applying principles of Recurrence Relations to find the generating functions and solve the Recurrence relations.
- Understand the knowledge of groups and their properties to understand the importance of algebraic properties relative to various number systems.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the delivered lessons shall develop students' theoretical and applied mathematical skills.
2. State the need for Mathematics with Engineering Studies and Provide real-life examples.
3. Support and guide the students for self-study.
4. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress.
5. Encourage the students for group learning to improve their creative and analytical skills.
6. Show short related video lectures in the following ways:
 - As an introduction to new topics (pre-lecture activity).
 - As a revision of topics (post-lecture activity).
 - As additional examples (post-lecture activity).
 - As an additional material of challenging topics (pre-and post-lecture activity).
 - As a model solution of some exercises (post-lecture activity).

Module – I

Fundamentals of Logic :

Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. Quantifiers and the Proofs of Theorems. **8 Hours**
(RBT Levels: L1, L2 and L3)

Module – II

Properties of the Integers:

Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions.

Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations and Combinations with Repetition. **8 Hours**
(RBT Levels: L1, L2 and L3)

Module – III

Relations and Functions:

Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions.

Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. **8 Hours**
(RBT Levels: L1, L2 and L3)

Module – IV

The Principle of Inclusion and Exclusion:

The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place.

Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients. **8 Hours**

(RBTLevels: L1,L2andL3)

Module – V

Introduction to Groups Theory:

Definitions and Examples of Particular Groups Klein 4-group, Additive group of Integers modulo n , Multiplicative group of Integers modulo- p and permutation groups, Properties of groups, Subgroups, cyclic groups. **8 Hours**

(RBTLevels: L1,L2andL3)

Teaching-Learning Process for all modules

Chalk and Talk/PowerPoint presentation/YouTube videos.

Course Outcomes:

After successfully completing the course, the students will be able:

1. Apply concepts of logical reasoning and mathematical proof techniques in proving theorems and statements.
2. Apply mathematical induction for recursive and non-recursive concepts.
3. Apply the basic concepts of relations, functions and partially ordered sets for computer representations.
4. Solve problems involving recurrence relations and generating functions.
5. Illustrate the fundamental principles of Algebraic structures with the problems related to computer science & engineering.

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	CCE-1	25 Marks
	CCE-2	25 Marks
	Average of two CCEs	25 Marks
Grand Total of IAT Marks (min marks 20 / 50)		50 Marks
SEE conducted for 100 and scaled down to 50 (min marks 18/50)		50 Marks
IAT + SEE (min marks 40)		100 Marks

Suggested Learning Resources:

Text Books:

1. **Ralph P. Grimaldi and B V Ramana:** Discrete and Combinatorial Mathematics- An Applied Introduction, Pearson Education, Asia, Fifth edition – 2007. ISBN 978-81-7758-424-0.
2. **Ralph P. Grimaldi:** “Discrete and Combinatorial Mathematics”, 5th Edition, Pearson Education. 2004.

Reference Books:

1. **Basavaraj S Anami and Venakanna S Madalli:** “Discrete Mathematics – A Concept-based approach”, Universities Press, 2016.
2. **Kenneth H. Rosen:** “Discrete Mathematics and its Applications”, 6th Edition, McGraw Hill, 2007.
3. **Jayant Ganguly:** “A Treatise on Discrete Mathematical Structures”, Sanguine-Pearson, 2010.
4. **D.S. Malik and M.K. Sen:** “Discrete Mathematical Structures Theory and Applications, Latest Edition, Thomson, 2004.
5. **Thomas Koshy:** “Discrete Mathematics with Applications”, Elsevier, 2005, Reprint 2008.

GRAPH THEORY			
Course Code	24CDT452	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03
<p>Course objectives:</p> <ul style="list-style-type: none"> ● Understand the basic concepts of graphs and their properties, and operations of graphs. ● Hamiltonian and Euler graphs, trees and matrix representation of the graph. ● Apply the concepts of a planar graph, matching and colouring in computer science engineering. 			
<p>Teaching-Learning Process (General Instructions)</p> <p>These are sample Strategies, which teacher can use to accelerate the attainment of the various course outcomes.</p> <ol style="list-style-type: none"> 1. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the delivered lessons shall develop students theoretical and applied mathematical skills. 2. State the need for Mathematics with Engineering Studies and Provide real-life examples. 3. Support and guide the students for self-study. 4. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress. 5. Encourage the students for group learning to improve their creative and analytical skills. 6. Show short related video lectures in the following ways: <ul style="list-style-type: none"> ● As an introduction to new topics (pre-lecture activity). ● As a revision of topics (post-lecture activity). ● As additional examples (post-lecture activity). ● As an additional material of challenging topics (pre-and post-lecture activity). ● As a model solution of some exercises (post-lecture activity). 			
Module – I			
<p>Introduction to Graphs: Introduction- Basic definition – Application of graphs – finite, infinite and bipartite graphs – Incidence and Degree – Isolated vertex, pendant vertex and Null graph. Paths and circuits – Isomorphism, sub-graphs, walks, paths and circuits, connected graphs, disconnected graphs and components. (8 Hours)</p> <p>[Text 1: 1.1, 1.2, 1.3, 1.4, 1.5, 2.1, 2.2, 2.4, 2.5] (RBT Levels: L1, L2 and L3)</p>			
Module – II			
<p>Eulerian and Hamiltonian graphs: Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Travelling salesman problem. Directed graphs – types of digraphs, Digraphs and binary relation. (8 Hours)</p> <p>[Text 1: 2.6, 2.7, 2.9, 2.10, 9.1, 9.2, 9.3] (RBT Levels: L1, L2 and L3)</p>			
Module – III			
<p>Trees – properties, pendant vertex, Distance and centres in a tree - Rooted and binary trees, counting trees, spanning trees.</p> <p>Connectivity Graphs: Vertex Connectivity, Edge Connectivity, Cut set and Cut Vertices, Fundamental circuits. (8 Hours)</p> <p>[Text 1: 3.1 - 3.7, 4.5, 4.1- 4.4]</p>			

(RBT Levels: L1, L2 and L3)

Module – IV

Planar Graphs: Planar graphs, Kuratowski's theorem (proof not required), Different representations of planar graphs, Euler's theorem, Geometric dual.

Graph Representations: Matrix representation of graphs-Adjacency matrix, Incidence Matrix, Circuit Matrix, Path Matrix.

(8 Hours)

[Text 1: 5.2 – 5.4, 5.6, 7.1 – 7.9]

(RBT Levels: L1, L2 and L3)

Module – V

Graph Colouring: Colouring- Chromatic number, Chromatic polynomial, Matchings, Coverings, Four colour problem and Five colour problem.

(8 Hours)

[Text 1 : 8.1 – 8.6]

(RBT Levels: L1, L2 and L3)

Teaching-Learning Process for all modules

Chalk and Talk/PowerPoint presentation/YouTube videos.

Course Outcomes:

After successfully completing the course, the students will be able:

- Explain the fundamental concepts of properties and representation of graphs.
- Solve the problems involving characterization and operations on graphs.
- Apply concepts of trees and graph connectivity to solve real world problems.
- Apply the concepts of planar graph and graph representations to solve the given problem.
- Use the concepts of matching and coloring of graphs to solve the real world problems.

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	CCE-1	25 Marks
	CCE-2	25 Marks
	Average of two CCEs	25 Marks
Grand Total of IAT Marks (min marks 20 / 50)		50 Marks
SEE conducted for 100 and scaled down to 50 (min marks 18/50)		50 Marks
IAT + SEE (min marks 40)		100 Marks

Text Books:

1. Narsingh Deo, Graph theory with the applications to engineering & Computer Science, Dovers Publications, 2016
2. J.A. Bondy and U.S.R. Murty. Graph theory with Applications, Springer, 1st edition, 2008.

REFERENCES

1. Garry Chartand and Ping Zhang, Introduction to Graph Theory, Tata McGraw-Hill, 2006.
2. Frank Harary, Graph Theory, Narosa Publishing House, Latest edition.
3. R. Diestel, Graph Theory, free online edition, 2016: diestel-graph-theory.com/basic.html.
4. Douglas B. West, Introduction to Graph Theory, Prentice Hall India Ltd.,2001
5. Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd.,2010

E-Resources:

- <http://nptel.ac.in/courses.php?disciplineID=111>
- [http://www.class-central.com/subject/math\(MOOCs\)](http://www.class-central.com/subject/math(MOOCs))

- <http://academicearth.org/>
- VTU e-Shikshana Program
- VTU EDUSAT Program

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

- Quizzes
- Assignments
- Seminars

GIT AND GITHUB

Course Code	24CDL461	CIE Marks	50
Teaching Hours /Week(L:T:P)	(0:0:2)	SEE Marks	50
Total Hours of Pedagogy	12 Lab slots	Total Marks	100
Credits	01	Exam Hours	03

Pre-requisite: Basic Command-Line Skills, Basic Text Editing Skills

Course objectives:

This course will enable students to:

1. Understand the concept of version control to track changes in your codebase.
2. Familiarize yourself with basic Git commands (init, add, commit, status, log).
3. Learn to create, switch, and merge branches.
4. Understand remote repositories and how to collaborate with others.
5. Learn how to handle merge conflicts when multiple contributors modify the same file.

LIST OF LABORATORY PROGRAMS

1.	Introduction to Git, Git setup: Install Git and SourceTree (on Windows only) on user machine, configure and create a workspace
2.	Creating a project: Create a account in Github and create a new repository in github. Link to Course Version Control with Git (Coursera):
3.	First commit, Branching and merge requests: Edit a file, commit and push (default IDE will be Source Tree), Create a branch, make changes, push and create merge request Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
4.	Git commands: Git config, git init, git clone, git add, git commit, git status Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
5.	Git commands: Git branch, git merge, git push, git pull, git check out Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
6.	First commit, Branching and merge requests: Edit a file, commit and push (default IDE will be Source Tree), Create a branch, make changes, push and create merge request Link to Course GitHub Ultimate: Master Git and GitHub - Beginner to Expert (Udemy):
7.	Merging branches by merge: Create a branch, make conflicting changes on master and branch and merge from branch to master Link to Course Introduction to Git and GitHub (edX):
8.	Cherry-picking, Reset and revert:

	Use GitLab GUI to make two changes to a file on a branch and then cherry-pick it the first change to master, Change a file, commit and reset the commit. Change a file, commit and revert the commit. Link to Course Interactive Learning Platforms: Git and GitHub (Codecademy):
9.	Issue Tracking Create an issue in an github. Link to Course Interactive Learning Platforms: Git and GitHub (Codecademy)
10.	Intro to Wikis; Create a wiki page for an simple project on github. Link to Course Interactive Learning Platforms: Git and GitHub (Codecademy)

Course Outcomes:

At the end of the course, the student will be able to :

1. Create the ability to track changes, revert to previous states, and manage different versions of the codebase.
2. Analyze the seamless collaboration among team members, with the ability to work on different features or fixes simultaneously through branching.
3. Demonstrate code integrity by preventing accidental overwrites and conflicts through branching and merging strategies.
4. Create clear and traceable history of changes, making it easier to identify when and why specific modifications were made.
5. Solve parallel development of features or bug fixes without disrupting the main codebase.

TEXT BOOKS:

1. "Git and GitHub Guide: The Basics" Jerry N P, Kindle Edition.
2. "Introduction to Git and GitHub" Bobby Iliev.
3. "GitHub For Dummies" Sarah Guthals, Phil Haack, and Zachary Voase, 2nd Edition, Dummies.
4. "Beginning Git and GitHub : A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer" Mariot Tsitoara , Apress.
5. "Git Prodigy: Mastering Version Control with Git and GitHub" Ebenezer Don.
6. "Git Pocket Guide" Richard E. Silverman, 1st Ed, O'Reilly Media.
7. "Learning GitHub" Aidan Feldman, Michaël Sels, and Edward Thomson, 1st Ed, O'Reilly Media.

E-Resources:

1. <https://github.com/bobbyiliev/introduction-to-git-and-github-ebook>
2. <https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>
3. <https://www.freecodecamp.org/news/git-and-github-for-beginners>
4. <https://www.datacamp.com/tutorial/github-and-git-tutorial-for-beginners>

Assessment Details(both IAT and SEE)

Continuous Internal Assessment of Laboratory/Practical Courses		
Lab Test 1	Lab Test 2	Lab Records
15 marks	15 marks	20 marks
Semester End Examination(SEE)		50 marks

Conduct of Practical Examination:

Experiment distribution :

For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the

changed part only.

- Marks Distribution (Need to change in accordance with university regulations)

a) For laboratories having only one part → Procedure + Execution + Viva-Voce:

$$15+70+15 = 100 \text{ Marks}$$

b) For laboratories having PART A and PART B

i. Part A – Procedure + Execution + Viva = $6 + 28 + 6 = 40$ Marks

ii. Part B – Procedure + Execution + Viva = $9 + 42 + 9 = 60$ Marks

OBJECT ORIENTED MODELLING LAB USING STAR UML

Course Code	24CDL462	CIE Marks	50
Teaching Hours /Week(L:T:P)	(0:0:2)	SEE Marks	50
Total Hours of Pedagogy	12 Lab slots	Total Marks	100
Credits	01	Exam Hours	03

Course objectives: Basics of object-oriented programming principles and software modeling concepts.

This course will enable students to:

- To Write the Problem Statement for the given System.
- To Specify Software Requirement Using Data Flow Diagram.
- To Draw structural and Behavioral diagrams for the given Specifications.

LIST OF LABORATORY PROGRAMS

Perform the following Experiments on following Domains
 Identify the software system that need to be Developed.
 Document the SRS for the identified System.
 Draw the level 0, level 1, and level 2 DFD for the Identified System.
 Draw the Class Diagrams and show Various Class Relationships.
 Draw the UML Component and Deployment Diagram for the identified System.
 Identify the use cases and Develop the Use case Model with include and external Relationships.
 Using the identified Scenarios find interaction between objects and represent using Sequence diagram.
 Using the identified Scenarios find interaction between objects and represent using Collaboration diagram.
 Draw the relevant Activity Diagram for the Same System
 Draw the relevant State Chart diagram for the same System

1.	Online Course Reservation System
2.	Railway Reservation System
3.	Exam Registrations
4.	Stock Maintenance Systems
5.	Recruitment Systems
6.	Library Management Systems.
7.	Student Information Systems.
8.	ATM

Course Outcomes:

At the end of the course the student will be able to :

- Design and develop the Problem Statement for the given system
- Analyze the Requirements Specification for an intended Software System Using DFD
- Design and capture the requirements specification for an intended Software system Using use case Modeling
- Design the Structural and behavioral Diagrams for the given specification.
- Design the Sequence Diagram, Activity Diagram and State Chart Diagram for the given specifications.

TEXT BOOKS:

1. **Martin Fowler**, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Addison-Wesley Object Technology Series, 2003

REFERENCES:

1. **Craig Larman**, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition, PHI, 2004

E-Resources:

1. **StarUML Documentation** : <https://docs.staruml.io/>
2. **TutorialsPoint UML Tutorial**: <https://www.tutorialspoint.com/uml/index.htm>
3. **GeeksforGeeks UML Tutorial** : <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>
4. **GitHub Repository**: <https://github.com/staruml>

Assessment Details(both IAT and SEE)

Continuous Internal Assessment of Laboratory/Practical Courses		
Lab Test 1	Lab Test 2	Lab Records
15 marks	15 marks	20 marks
Semester End Examination(SEE)		50 marks

Conduct of Practical Examination:**Experiment distribution :**

For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (Need to change in accordance with university regulations)
 - a) For laboratories having only one part → Procedure + Execution + Viva-Voce:
15+70+15 = 100 Marks
 - b) For laboratories having PART A and PART B
 - i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks
 - ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks

BIO-ENGINEERING AND INNOVATIONS

Course Code	24BET47	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	2:0:0:0	SEE Marks	50
Total Hours of Pedagogy	26	Total Marks	100
Credits	02	Exam Hours	03

Course objectives:

1. To acquire fundamental knowledge of biomolecules and their significance in biological and engineering systems.
2. To understand core biological concepts from an engineering perspective, enabling interdisciplinary problem-solving.
3. To analyze human organ systems and explore engineering solutions for diagnosis, monitoring, and replacement.
4. To understand the principles and mechanisms of bio-inspired materials and their engineering applications.
5. To explore how biological systems can be redesigned or mimicked to develop engineering alternatives to natural system.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Explanation via real-life problem, situation modeling, and deliberation of solutions, hands-on sessions, reflective and questioning /inquiry-based teaching.
2. Instructions with interactions in classroom lectures (physical/hybrid).
3. Use of ICT tools, including YouTube videos, related MOOCs, and AR/VR/MR tools.
4. Flipped classroom sessions (~10% of the classes).
5. Industrial visits, Guest talks, and competitions for learning beyond the syllabus.
6. Students' participation through audio-video-based content creation for the syllabus (as assignments).
7. Use of Gamification tools (in both physical/hybrid classes) for creative learning outcomes.
8. Students' seminars (in solo or group) /oral presentations.

Module – I

Biomolecules And Their Applications:

Carbohydrates (Bio-based Nanofiber membranes – water purification), Sustainable Alternatives to Conventional Plastics: PHA, PLA, and PHB; Nucleic acids (Vaccine Technologies for Viral Diseases (rabies and COVID) and Cancer); DNA Fingerprinting in Forensic Science; Proteins (whey protein, meat analogs) and therapeutic proteins (Insulin); Lipids: Biodiesel and Advanced Biodiesel (Algae & Waste Oils).

5 Hours

Module – II

Human Organ Systems and Bio Designs 1:

EEG, robotic arms for prosthetics, AI-based brain signal analysis, engineering solutions for Parkinson's disease, eye-as-a-camera system (bionic eye) and AI-assisted vision restoration, heart-related technologies including ECG, design of stents (Bioresorbable stent), pacemakers, and defibrillators.

5 Hours

Module – III

Human Organ Systems and Bio Designs 2:

Lungs as a purification system (Spirometry, Ventilators and Heart Lung Machine), Kidney - Dialysis and its types, Bioartificial kidney; Muscular and Skeletal Systems (Bioengineering solutions for muscular dystrophy and osteoporosis).

5 Hours

Module – IV

Nature-Bioinspired Materials and Mechanisms:

Echolocation (ultrasonography, sonars, LiDAR), Photosynthesis (photovoltaic cells, bionic leaf), Lotus

leaf effect (Super hydrophobicity and self-cleaning surfaces), Plant burrs (Velcro technology), Sharkskin (Friction reducing swimsuits), Kingfisher beak (Bullet train) and Human Blood Substitute-Hemoglobin based Oxygen Carriers (HBOCs, Emergency medicine & military applications) and Perfluorocarbon (PFCs)

6 Hours

Module – V

Trends In Bioengineering:

3D and 4D bioprinting techniques and materials; electrical tongue and electronic nose in food science; lab-on-a-chip devices and wearable biosensors; DNA origami and nanorobots; artificial intelligence for disease diagnosis; digital twins in healthcare; self-healing bio concrete; bioremediation and biomining; climate-resilient biotechnology.

5 Hours

Course outcome (Course Skill Set)

At the end of the course, the student will be able to :

1. Apply the concepts of biomolecules to different bioengineering applications.
2. Analyze the functions and mechanism of human organ system to develop the bio mimicking devices.
3. Design and implement the bio-engineering Devices to various biomedical applications
4. Evaluate the Bioinspired materials for industrial applications.
5. Build the basic biological concepts via fabricating bio-devices

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	CCE-1	25 Marks
	CCE-2	25 Marks
	Average of two CCEs	25 Marks
Grand Total of CIE Marks (min marks 20 / 50)		50 Marks
SEE conducted for 100 and scaled down to 50 (min marks 18/50)		50 Marks
CIE + SEE (min marks 40)		100 Marks

Suggested Learning Resources:

Textbooks

Sl. No.	Unit	Textbook Title	Author(s)	Publisher(s)	Edition/Year of Publication
1.	All	Human Physiology	Stuart Fox, Krista Rompolski	McGraw-Hill eBook.	16th Edition, 2022
2.	All	Biology for Engineers,	Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K.	Tata McGraw-Hill, New Delhi,	2012.
3.	All	Biology for Engineers,,,	Arthur T. Johnson	CRC Press, Taylorand Francis	2011
4.	All	Biomedical Instrumentation,	Leslie Cromwell	Prentice Hall	2011.
5.	All	Biology for Engineers	Sohini Singh andTanu	Vayu Education	2014.

			Allen	of India, New Delhi	
6.	All	Biomimetics: Nature- Bar-	Based Innovation, Yoseph Cohen,	1st edition, CRC Press	2012.
7.	All	Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies,	D. Floreano and C. Mattiussi,	MIT Press	2008.
8.	All	Bioremediation of heavy metals: bacterial participation	C R Sunilkumar, N Geetha A C Udayashankar	Academic Publishing, Lambert	2019.
9.	All	3D Bioprinting: Fundamentals, Principles and Applications	Ibrahim Ozbolat,	Academic Press	2016.
10.	All	Electronic Noses and Tongues in Food Science,	Maria Rodriguez Mende	Academic Press	2016.

Web links and Video Lectures (e-Resources):

<https://nptel.ac.in/courses/121106008>
<https://freevidelectures.com/course/4877/nptelhttps://ocw.mit.edu/courses/20-020-introduction-tobiological-engineering-design-spring-2009> <https://ocw.mit.edu/courses/20-010j-introduction-tobioengineering-be-010j-spring-2006> <https://www.coursera.org/courses?query=biology>
https://onlinecourses.nptel.ac.in/noc19_ge31/preview
<https://www.classcentral.com/subject/biology> <https://www.futurelearn.com/courses/biology-basic-concepts>

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Group Discussion of Case studies
- Product development, seminar and poster presentations
- Design of novel device/equipment like Biosensors, ECG and Electronic Nose.

CO PO Mapping

POS	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	-	3	-	-	-	-	-	-	-	-	-	-	-	-
CO3	-	-	3	-	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	2	2	-	-	-	-	-	-	-	-	-
CO5	-	-	-	-	1	2	-	-	1	1	2	-	-	-

UNIVERSAL HUMAN VALUES

Course Code	24UHV48	CIE Marks	50
Teaching Hours/Week (L:T:P:S)	1:0:0:0	SEE Marks	50
Total Hours of Pedagogy	15 hour Theory Session +15 hour Self study	Total Marks	100
Credits	01	Exam Hours	01

Prerequisites:

Society, Constitution, Ethics, Environment.

Course objectives:

This introductory course input is intended:

1. To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

This course is intended to provide a much-needed orientational input in value education to the young enquiring minds.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teacher can use to accelerate the attainment of the various course outcomes.

1. The methodology of this course is explorational and thus universally adaptable. It involves a systematic and rational study of the human being vis-à-vis the rest of existence.
2. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the activities will develop students' theoretical and applied skills.
3. State the need for UHV activities and its present relevance in the society and Provide real-life examples.
4. Support and guide the students for self-study activities.
5. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress in real activities in the field.
6. This process of self-exploration takes the form of a dialogue between the teacher and the students to begin with, and then to continue within the student in every activity, leading to continuous self evolution.
7. Encourage the students for group work to improve their creative and analytical skills.

Module – I

Introduction to Value Education

Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education) Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfill the Basic Human Aspirations.

03 Hours

Module – II

Harmony in the Human Being

Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health.

03 Hours

Module – III

Harmony in the Family and Society

Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.

03 Hours

Module – IV

Harmony in the Nature/Existence

Understanding Harmony in the Nature, Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence.

03 Hours

Module – V

Implications of the Holistic Understanding – a Look at Professional Ethics

Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics Holistic Technologies, Production Systems and Management Models- Typical Case Studies, Strategies for Transition towards Value-based Life and Profession

03 Hours

Teaching-Learning Process

Introduction to the concepts- Chalk and talk method, Discussion, Sharing of experiences, Live Examples and videos

Course outcomes:

CO1: Apply a holistic vision of life

CO2: Analyze the socially responsible behavior

CO3: Demonstrate the environmentally responsible work

CO4: Apply ethical human conduct

CO5: Create the Competence and Capabilities for Maintaining Health and Hygiene

Appreciation and aspiration for excellence (merit) and gratitude for all

Assessment Details (both IAT and SEE)

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Average of two IATs	25 Marks
	CCE-1	25 Marks
	CCE-2	25 Marks
	Average of two CCEs	25 Marks
Grand Total of IAT Marks (min marks 20 / 50)		50 Marks
SEE conducted for 100 and scaled down to 50 (min marks 18/50)		50 Marks
IAT + SEE (min marks 40)		100 Marks

Suggested Learning Resources:

Books

-READINGS:

Text Book and Teachers Manual

a. The Textbook

A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1

b. The Teacher's Manual

Teachers' Manual for A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G

Reference Books

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj – Pandit Sunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)
14. Susan George, 1976, How the Other Half Dies, Penguin Press. Reprinted 1986, 1991
15. Donella H. Meadows, Dennis L. Meadows, Jorgen Randers, William W. Behrens III, 1972, Limits to Growth – Club of Rome's report, Universe Books.
16. A Nagaraj, 1998, Jeevan Vidya Ek Parichay, Divya Path Sansthan, Amarkantak.
17. P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers.
18. A N Tripathy, 2003, Human Values, New Age International Publishers.
19. Subhas Palekar, 2000, How to practice Natural Farming, Pracheen (Vaidik) Krishi Tantra Shodh, Amravati.
20. E G Seebauer & Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers, Oxford University Press
21. M Govindrajran, S Natrajan & V.S. Senthil Kumar, Engineering Ethics (including Human Values), Eastern Economy Edition, Prentice Hall of India Ltd.
22. B P Banerjee, 2005, Foundations of Ethics and Management, Excel Books.
23. B L Bajpai, 2004, Indian Ethos and Modern Management, New Royal Book Co., Lucknow. Reprinted 2008.

Web links and Video Lectures (e-Resources):

1. Value Education websites, <https://www.uhv.org.in/uhv-ii>, <http://uhv.ac.in>, <http://www.uptu.ac.in>
2. Story of Stuff, <http://www.storyofstuff.com>
3. Al Gore, An Inconvenient Truth, Paramount Classics, USA
4. Charlie Chaplin, Modern Times, United Artists, USA
5. IIT Delhi, Modern Technology – the Untold Story
6. Gandhi A., Right Here Right Now, Cyclewala Productions
7. https://www.youtube.com/channel/UCQxWr5QB_eZUnwxSwxXEKQw
8. https://fdp-si.aicte-india.org/8dayUHV_download.php
9. <https://www.youtube.com/watch?v=8ovkLRYXjE>
10. <https://www.youtube.com/watch?v=OgdNx0X923I>
11. <https://www.youtube.com/watch?v=nGRcbRpvGoU>
12. <https://www.youtube.com/watch?v=sDxGXOgYEKM>

Activity Based Learning (Suggested Activities in Class)/ Practical learning