

**24MA102**  
**Mathematics-II**  
**(DIFFERENTIAL EQUATIONS AND NUMERICAL METHODS)**  
**COMMON TO ALL BRANCHES**

<b>Course Category:</b>	<b>Basic Sciences (BS)</b>	<b>Credits:</b>	<b>4</b>
<b>Course Type:</b>	<b>Theory</b>	<b>Lecture-Tutorial-Practice:</b>	<b>3-0-2</b>
<b>Pre-requisites:</b>	<b>10+2 Mathematics</b>	<b>Continuous Assessment:</b> <b>Summative Assessment:</b> <b>Total Marks:</b>	<b>40</b> <b>60</b> <b>100</b>

**Course Description:**

An overview of the fundamental concepts and methods to solve differential equations, Numerical methods to solve algebraic and transcendental equations, initial, boundary value problems and polynomial estimation using interpolation technique with a focus on the applications in solving engineering problems

**Course Objectives:**

- Introduce techniques for solving differential equations
- Explain methods to solve algebraic and transcendental equations and estimate function values through interpolation /extrapolation
- Acquaint the knowledge of solving initial and boundary value problems

**Course Outcomes:**

At the end of the course, the students will be able to

<b>CO</b>	<b>Course Outcomes</b>	<b>BTL</b>
<b>CO1</b>	Apply suitable approaches to solve first order and first-degree ordinary differential equations with engineering applications	<b>K3</b>
<b>CO2</b>	Solve higher order ordinary linear differential equations with engineering applications	<b>K3</b>
<b>CO3</b>	Solve the partial differential equations	<b>K3</b>
<b>CO4</b>	Use iterative methods for solving algebraic & transcendental equations and compute the functional values by polynomial interpolation	<b>K3</b>
<b>CO5</b>	Apply numerical methods to solve initial and boundary value problems	<b>K3</b>

**Course articulation matrix**

<b>COs</b>	<b>POs</b>											<b>PSOs</b>	
	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PSO1</b>	<b>PSO2</b>
<b>CO1</b>	2	2			1								
<b>CO2</b>	2	2			1								
<b>CO3</b>	2	2			1								
<b>CO4</b>	2	2			1								
<b>CO5</b>	2	2			1								

(1-Low, 2 -Medium, 3-High)

**Course Content**

**Unit-I**

**Differential Equations of First Order[T1]**

Linear equations, Bernoulli's equation, Exact equations, Equations reducible to exact equations, Newton's Law of cooling, Law of growth and decay.

**Unit-II**

**Higher Order Linear Differential Equations with Constant Coefficients**

**[T1]**

Definitions, Working procedure to solve the equation, Wronskian, Method of variation of

parameters, Simultaneous linear equations, Electrical circuits.

### Unit-III

#### Partial Differential Equations

[T1]

Introduction, Formation of partial differential equations, Solutions of a linear equation of the first order using Lagrange's method, Homogeneous linear partial differential equations with constant coefficients: Working procedure to solve the equation.

### Unit-IV

#### Numerical Methods and Interpolation[T1]

**Root finding methods:** Bisection method, Method of Iteration, Newton-Raphson method

**Interpolation:** Introduction, Finite Differences, relation between the operators, Newton's interpolation formulae: forward and backward differences, Interpolation with unequal intervals: Lagrange's and Newton's divided difference formulae.

### Unit-V

#### Initial and Boundary value problems

[T1]

Numerical Solution of ordinary differential equations: Taylor's series, Euler's, modified Euler's and 4th order Runge-Kutta methods, Elliptic partial differential equations: Laplace's and Poisson's equations.

#### Text Books:

- [1]. Grewal B. S. (2017). Higher Engineering Mathematics. ( 44<sup>th</sup> Edition). Khanna Publishers.
- [2]. SankaraRao K (2014). Numerical Methods for Scientists and Engineers.(3<sup>rd</sup> Edition). PHI Learning Private Limited.

#### Reference Books:

- [1]. Kreyszig Erwin. (2013). Advanced Engineering Mathematics.(9<sup>th</sup> Edition). Wiley Publishers.
- [2]. Ramana B.V.(2007). Higher Engineering Mathematics. Tata Mc.Graw Hill.
- [3]. Sastry S.S. (2012). Introductory Methods of Numerical Analysis.(5<sup>th</sup> Edition). PHI Learning Private Limited.

#### Web Resources:

- [1]. NPTEL: Differential Equations For Engineers  
<https://nptel.ac.in/courses/111106100>
- [2]. NPTEL: Numerical Analysis  
<https://nptel.ac.in/courses/111101165>
- [3]. MIT OpenCourseware: Differential Equations  
<https://ocw.mit.edu/courses/18-03-differential-equations-spring-2010/>
- [4]. MIT OpenCourseWare: Introduction to Partial Differential Equations  
<https://ocw.mit.edu/courses/18-152-introduction-to-partial-differential-equations-fall-2011/>
- [5]. MIT OpenCourseware: Introduction to Numerical Analysis  
<https://ocw.mit.edu/courses/18-330-introduction-to-numerical-analysis-spring-2012/>

<b>24MA102</b>											
<b>DIFFERENTIAL EQUATIONS AND NUMERICAL METHODS</b>											
<b>CourseCategory:</b>	<b>HS</b>	<b>Credits:</b>	<b>1C</b>								
<b>CourseType:</b>	<b>Integrated Lab</b>	<b>Lecture-Tutorial-Practice:</b>	<b>0L-0T-2P</b>								
<b>Pre-requisites:</b>	<b>Mathematics-II</b>	<b>Continuous Assessment:</b>	<b>40</b>								
		<b>Summative Assessment:</b>	<b>60</b>								
		<b>TotalMarks:</b>	<b>100</b>								
<b>CourseDescription:</b>											
AN OVERVIEW OF THE FUNCTIONS OF MATHWORKS SYMBOLIC MATH TOOL BOX TO SOLVE FIRST ORDER DIFFERENTIAL EQUATIONS, HIGHER ORDER LINEAR PARTIAL DIFFERENTIAL EQUATIONS WITH CONSTANT COEFFICIENTS, ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS WITH INITIAL AND BOUNDARY CONDITIONS AND FIND ROOT OF TRANSCENDENTAL EQUATIONS.											
<b>CourseObjectives:</b>											
<ol style="list-style-type: none"> <li>1. Find solution of first order differential equations.</li> <li>2. Solve higher order linear differential equations with constant coefficients.</li> <li>3. Solve elliptic partial differential equations with initial and boundary conditions</li> <li>4. Find root of transcendental equations.</li> <li>5. Solve numerically first order initial value problems.</li> </ol>											
<b>CourseOutcomes:</b>											
At the end of the course, the students will be able to											
<b>CO</b>	<b>CourseOutcomes:</b>		<b>BTL</b>								
<b>CO1</b>	Find solution of first order differential equations		(K3)								
<b>CO2</b>	Solve higher order linear differential equations with constant coefficients.		(K3)								
<b>CO3</b>	Solve elliptic partial differential equations with initial and boundary conditions.		(K3)								
<b>CO4</b>	Find root of transcendental equations.		(K3)								
<b>CO5</b>	Solve numerically first order initial value problems.		(K3)								
<b>Course articulation matrix</b>											
<b>Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2 – Medium, 3 – High)</b>											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	2			1						
CO2	2	2			1						
CO3	2	2			1						
CO4	2	2			1						
CO5	2	2			1						
<b>Course Content</b>											
<b>Unit-I</b>											
dsolve() function to solve first order differential equations.											
<b>Unit-II</b>											
dsolve() function to solve higher order linear differential equations with constant coefficients											

### Unit-III

Laplace's pde solution using finite difference method  
Poisson's pde solution using finite difference method

### Unit-IV

fzero() function to find roots of an algebraic and transcendental equations  
interp1() function for interpolation

### Unit-V

Euler's method to solve first order initial value problem  
Runge-Kutta 4<sup>th</sup> order method to solve first order initial value problem  
Ode78() function to solve first order initial value problems

#### TextBooks:

MatlabLabManual

#### WebResources:

- MathWorks Ordinary Differential Equations Documentation  
[Solve nonstiff differential equations — medium order method - MATLAB ode45 - MathWorks India](#)
- MathWork Symbolic Calculus Tool Box  
[Solve system of differential equations - MATLAB dsolve - MathWorks India](#)
- MathWorks Linear Algebra Documentation  
[Linear Algebra - MATLAB & Simulink - MathWorks India](#)
- MathWork Tool Box  
[Root of nonlinear function - MATLAB fzero - MathWorks India](#)
- MathWork Tool Box  
[1-D data interpolation \(table lookup\) - MATLAB interp1 - MathWorks India](#)
- MathWork ODE Tool Box  
<https://in.mathworks.com/help/matlab/ref/ode78.html#d126e1197289>

**24CY101**  
**Applied Chemistry**

<b>Course Category:</b>	<b>Basic Sciences</b>	<b>Credits:</b>	<b>3</b>
<b>Course Type:</b>	<b>Theory</b>	<b>Lecture-Tutorial-Practice:</b>	<b>3-0-0</b>
<b>Pre-requisites:</b>	<b>10+2 Chemistry</b>	<b>Continuous Assessment:</b> <b>Summative Assessment:</b> <b>Total Marks:</b>	<b>40</b> <b>60</b> <b>100</b>

**Course Description:**

The course is designed for B.Tech. computer science & engineering, artificial intelligence and information technology students, providing essential and applied knowledge of Chemistry relevant to their fields. Understanding the specific application of materials based on their characteristic properties, which are determined by their structural and bonding aspects, is crucial for appropriate utilization. With this focus, the course covers the working of electrodes, sensors, batteries, fuel cells and super capacitors which are application of principles of electrochemistry. Also, the course enlightens the students on mechanistic aspects of corrosion and its control, and chemical aspects of conducting materials. Further, it focuses on the chemistry of materials for memory and display systems used in hardware of computer science.

**Course Objectives:**

1. Impart knowledge on the functioning of electrodes, potentiometric and conductometric sensors.
2. Discuss construction and working of different types of batteries, fuel cells and super capacitors based on the principles of electrochemistry.
3. Analyse various corrosion processes and propose control methods depending on the principles of corrosion.
4. Explore the chemical aspects of various conducting materials useful in hardware components of computer systems.
5. Explain chemical aspects of different materials that are commercially used in memory and display systems

**Course Outcomes:**

At the end of the course, the students will be able to

<b>CO</b>	<b>Course Outcomes</b>	<b>BTL</b>
<b>CO1</b>	Apply the principles of electrochemistry to analyse working of electrodes and sensors	<b>K3</b>
<b>CO2</b>	Analyse various electrochemical energy systems for their application in engineering	<b>K4</b>
<b>CO3</b>	Assess the challenges arising due to corrosion of electronic devices	<b>K5</b>
<b>CO4</b>	Demonstrate the knowledge of conducting materials for their use in manufacture hardware components of computer systems	<b>K3</b>
<b>CO5</b>	Compare chemical aspects of different types of materials used in memory and display systems[	<b>K4</b>

**Course articulation matrix**

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

(1-Low, 2-Medium, 3-High)(WK1)

**Course Content****Unit-I  
Electrochemistry**

- Electrodes, electrode potentials and electrochemical cells
- The Nernst equation with numerical problems for calculating electrode potential and emf
- Reference electrodes – Calomel and Ag/AgCl electrodes, Ion-selective electrodes, glass electrode - construction, working, advantages, and disadvantages
- Potentiometry – redox titrations
- Conductometry (acid-base reactions)
- Electrochemical sensors – principle and applications

**Unit-II  
Electrochemical Energy Systems**

- Types of electrochemical energy systems – charging vs. discharging
- Primary vs. secondary batteries
- Lithium-ion batteries – Lithium iron phosphate and lithium cobalt oxide – construction and working of the batteries including cell reactions
- Fuel cells – hydrogen-oxygen fuel cell and polymer electrolyte membrane fuel cell
- Super capacitors – principle, classification and applications
- Chemistry of fast charging EVs.

**Unit-III  
Corrosion and Its Control**

- Introduction to corrosion, causes and examples
- Electrochemical corrosion: hydrogen evolution and oxygen absorption corrosion
- Differential aeration corrosion
- Galvanic corrosion and its control, including the galvanic series

- Corrosion in computer and microelectronic devices
- Factors influencing corrosion
- Electroplating and electroless plating

**Unit-IV**  
**Chemistry of Conducting Materials**

- Conducting polymers: Types of conducting polymers, mechanisms of conduction in undoped, doped polyacetylene and engineering applications of conducting polymers.
- Other materials of conduction: Production of electronic grade silicon from quartz and its applications, metal compounds as semiconductors, applications of carbon nanotubes and graphene in electrical and electronic industry.

**Unit-V**  
**Materials for Memory and Display Systems**

- Basic concepts of electronic memory, classification of electronic memory devices (Transistor-type, Capacitor-type, Resistor-type and Charge transfer type), types of organic memory devices (organic molecules, polymeric materials, organic-inorganic hybrid materials).
- Photoactive and electroactive materials – principle, Organic materials – Light absorbing materials – polythiophenes, Light emitting materials – Poly(9-vinylcarbazole) – their uses in optoelectronic devices, Organic light emitting diodes (OLED) and Quantum light emitting diodes (QLED) – properties and applications

**Text Books:**

1. Ramesh, S. (2013). *Engineering chemistry* (2nd ed.). Wiley India.
2. Shikha Agarwal, (2015). *Engineering chemistry: fundamentals and applications* (1st ed.). Cambridge University Press.
3. Jain, P.C. (2018). *Engineering chemistry* (17th ed.). Dhanpat Rai.

**Reference Books:**

1. PrasanthaRath, & ArunaKumari, S. (2023). *Engineering chemistry* (1st ed.). Cengage.
2. ArunBahl, Bahl, B. S., & Tuli, G. D. (2020). *Essentials of physical chemistry* (28th ed.). S. Chand.
3. Hagi, A. K., Mercader, A. G., Balkoese, D., & Mukbaniani, O. V. (2021). *Applied chemistry and chemical engineering*, (1st ed.). CRC Press, Taylor & Francis Group.
4. Skoog, D. A., West, D. M., Holler, F. J., & Crouch, S. R. (2022). *Fundamentals of analytical chemistry* (10th ed.). Cengage.
5. Fontana, M. G. (2017). *Corrosion engineering* (3rd ed.). McGraw-Hill Education.
6. Swaminathan, P. (2017). *Semiconductor materials, devices, and fabrication*. Wiley.
7. Banwell, C. N., & McCash, E. M. (2017). *Fundamentals of molecular spectroscopy* (4th ed.). McGraw-Hill Education.

**Web Resources:**

- <https://www.sciencedirect.com/topics/chemistry/electrochemistry>
- <https://wme-z1.pwr.edu.pl/wp-content/uploads/2017/05/Basics-of-Electrochemistry.pdf>
- <https://ocw.mit.edu/courses/10-626-electrochemical-energy-systems-spring-2014/pages/lecture-notes/>
- [https://pressbooks.online.ucf.edu/chemistryfundamentals/chapter/batteries-and-fuel-cells- 2/](https://pressbooks.online.ucf.edu/chemistryfundamentals/chapter/batteries-and-fuel-cells-2/)
- <https://pesjournal.net/journal/v3-n1/2.pdf>
- [https://www.researchgate.net/publication/275028997\\_Corrosion\\_and\\_Corrosion\\_Control](https://www.researchgate.net/publication/275028997_Corrosion_and_Corrosion_Control)
- <https://pubs.rsc.org/en/content/articlehtml/2021/ra/d0ra07800j>
- <https://nanografi.com/blog/application-areas-of-nanotechnology-in-display-and-communication-technology/>
- [https://www.researchgate.net/publication/338692940\\_Memory\\_materials\\_and\\_devices\\_From\\_concept\\_to\\_application/link/64dd18fe25837316ee143491/download](https://www.researchgate.net/publication/338692940_Memory_materials_and_devices_From_concept_to_application/link/64dd18fe25837316ee143491/download)
- <https://www.azom.com/article.aspx?ArticleID=285>

**24BY101**  
**Biology for Engineers**

<b>Course Category:</b>	<b>Basic Sciences</b>	<b>Credits:</b>	<b>3</b>
<b>Course Type:</b>	<b>Theory</b>	<b>Lecture-Tutorial-Practice:</b>	<b>3-0-0</b>
<b>Pre-requisites:</b>	-	<b>Continuous Assessment:</b>	<b>40</b>
		<b>Summative Assessment:</b>	<b>60</b>
		<b>Total Marks:</b>	<b>100</b>

**Course Description:**

The course introduces engineering students to key concepts in biology essential for understanding biological systems and their integration into engineering practices. It covers a wide range of topics from basic biological principles to advanced applications in biomedical engineering, bio materials, synthetic biology, and bioinformatics.

**Course Objectives:**

1. Introduce engineers to foundational concepts in biology essential for understanding biological systems and their relevance to engineering applications.
2. Explore the properties and behavior of biomaterials and biomechanics principles that influence the design and functionality of biomedical devices.
3. Investigate bio-inspired design principles, synthetic biology techniques, and biotechnological applications for engineering innovation.
4. Explore bioinformatics tools and systems biology approaches to analyse complex biological data and understand biological systems at a molecular and cellular level.
5. Investigate advanced topics in biological engineering, such as Nano biotechnology, regenerative medicine, or emerging biotechnologies, to expand students' knowledge and skills.

**Course Outcomes:**

At the end of the course, the students will be able to

<b>CO</b>	<b>Course Outcomes</b>	<b>BTL</b>
<b>CO1</b>	demonstrate a fundamental understanding of basic biological principles, such as cell biology, genetics, and physiology, and their application in engineering contexts.	K3
<b>CO2</b>	analyse and evaluate biomaterials based on their mechanical properties and design biomedical devices using biomechanical principles.	K4
<b>CO3</b>	apply bio-inspired design principles and synthetic biology techniques to propose innovative engineering solutions and understand their practical applications in biotechnology.	K3
<b>CO4</b>	develop proficiency in using bioinformatics tools, constructing computational models, and applying systems biology approaches to study and interpret biological data and systems.	K3
<b>CO5</b>	critically analyse and evaluate current research trends and innovations in advanced areas of biological engineering, demonstrating the ability to apply advanced concepts in real-world engineering scenarios.	K4

**Course articulation matrix**

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

(1-Low, 2 -Medium, 3–High) (WK1)

**Course Content**

**Unit-I**

**Introduction to Biology for Engineers**

- Overview of interdisciplinary nature and importance of biology for engineers
- Basic biological concepts relevant to engineering applications
- Introduction to engineering challenges addressed by biology-inspired solutions
- Introduction to the structure and functions of Biomolecules

**Unit-II**

**Biomaterials and Biomechanics**

- Properties and applications of biomaterials in engineering (e.g., tissue engineering, drug delivery systems)
- Biomechanical principles applied to engineering (e.g., mechanics of bones and joints, biomechanics of tissues)
- Case studies: Design and development of biomaterials for medical implants, prosthetics, and tissue engineering

**Unit-III**

**Bio inspired Design, Synthetic Biology and Biotechnology**

- Introduction to bioinspired design, biomimicry, synthetic biology and genetic engineering (Genetically modified Crops & Animals)
- Examples of biological systems inspiring engineering innovations (e.g., bird flight, gecko adhesion, shark skin, human eye camera)
- Applications of biotechnology in engineering (e.g., bioprocessing, bioremediation, biosensors)

- Case studies: Biomimetic design in robotics, materials science, architecture, Development of biofuels, bioplastics, and biopharmaceuticals

**Unit-IV**  
**Bioinformatics and Systems Biology**

- Introduction to bioinformatics tools and databases
- Engineering applications of bioinformatics (e.g., genome sequencing, protein structure prediction, drug discovery)
- Systems biology approaches to understanding complex biological systems and networks
- Case studies: Computational modelling of biological systems, personalized medicine, and synthetic biology

**Unit-V**  
**Advanced Topics in Biology for Engineers**

- Emerging trends and cutting-edge research in biology for engineers
- Advanced applications and case studies in specific engineering fields (e.g., nanotechnology, energy, environmental engineering)
- Ethical considerations (Bio-Safety) and societal implications (Biopiracy & Bio patent)

**Text Books:**

1. Johnson, A. T. (2011). Biology for Engineers. CRC Press.
2. Renneberg, R. (2017). Biotechnology for Engineers: Biological Processes and Technologies (1st ed.). Elsevier.
3. Mount, D. W. (2004). Bioinformatics: Sequence and Genome Analysis (2nd ed.). Cold Spring Harbor Laboratory Press.

**Reference Books:**

1. Enderle, J., & Bronzino, J. (2011). Introduction to Biomedical Engineering (3rd ed.). Academic Press.
2. Ratner, B. D., Hoffman, A. S., Schoen, F. J., & Lemons, J. E. (2012). Biomaterials Science: An Introduction to Materials in Medicine (3rd ed.). Academic Press.
3. Benyus, J. M. (2002). Biomimicry: Innovation Inspired by Nature. Harper Perennial.

**Web Resources:**

- 1: Introduction to Biology for Engineers (<https://archive.org/details/biologyforengine0000john>)

- 2: The Interdisciplinary Role of Biology in Engineering (<https://onlinecourses.nptel.ac.in/>)
- 3: Biomaterials in Engineering (<https://link.springer.com/book/10.1007/978-3-031-35832-6>)
- 4: Biomechanics Principles (<https://www.cambridge.org/core/books/introductory-biomechanics/79733231375F00B75C84A47CCABB9386>)
- 5: Bioinspired Design (<https://link.springer.com/article/10.1007/s00163-020-00333-w>)
- 6: Biomimicry Applications (<https://www.cambridge.org/core/books/abs/bioinspired-structures-and-design/bioinspired-and-biomimetic-design-of-multilayered-and-multiscale-structures/D14F3C8BD06061B5983B7DAB55D6B1FA>)
- 7: Fundamentals of Synthetic Biology (<https://www.nature.com/subjects/synthetic-biology>)
- 8: Biotechnology Applications (<https://open.umn.edu/opentextbooks/textbooks/820>)
- 8: Bioinformatics Tools and Databases (<https://microbenotes.com/bioinformatics-databases-software-tools/>)
- 9: Systems Biology Approaches ([https://link.springer.com/protocol/10.1007/978-981-99-6913-5\\_9](https://link.springer.com/protocol/10.1007/978-981-99-6913-5_9))
- 10: Emerging Trends in Bioengineering (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7392392/>)
- 11: Advanced Applications in Bioengineering (<https://ibecbarcelona.eu/about-us/applications-in-bioengineering/>)
- 12: Ethical Considerations in Biotechnology (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6612373/>)

**24IT102**  
**DATA STRUCTURES**

<b>Course Category:</b>	Engineering Sciences (ES)	<b>Credits:</b>	<b>3</b>
<b>Course Type:</b>	Theory	<b>Lecture-Tutorial-Practice:</b>	<b>2-0-2</b>
<b>Pre-requisites:</b>	Programming Fundamentals	<b>Continuous Assessment:</b> <b>Summative Assessment:</b> <b>Total Marks:</b>	<b>30</b> <b>70</b> <b>100</b>

**Course Description:**

This course provides an in-depth understanding of data structures and their applications. Students will learn to design, analyze, and implement various data structures such as arrays, linked lists, stacks, queues, trees, and graphs. The course also covers algorithms for sorting, searching, and manipulation of data, emphasizing the importance of selecting appropriate data structures for efficient problem-solving.

**Course Objectives:**

1. *Develop a solid understanding of fundamental data structures and their operations.*
2. *Design algorithms for sorting, searching, and basic data manipulation.*
3. *Evaluate the efficiency of various data structures and algorithms.*
4. *Implement data structures to solve practical computational problems.*
5. *Apply data structures to enhance problem-solving abilities in software development.*

**Course Outcomes:**

At the end of the course, the students will be able to

<b>CO</b>	<b>Course Outcomes</b>	<b>BTL</b>
<b>CO1</b>	Identify and differentiate between various data structures and their use cases.	<b>K2</b>
<b>CO2</b>	Implement data structures like stacks, queues, linked lists, and trees using C	<b>K3</b>
<b>CO3</b>	Analyze the time and space complexity of data structures and algorithms	<b>K4</b>
<b>CO4</b>	Solve problems using dynamic data structures like linked lists, trees, and graphs	<b>K3</b>
<b>CO5</b>	Apply sorting and searching techniques to real-world problems	<b>K3</b>

**Course articulation matrix**

COs	POs												PSO 1	PSO 2
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11			
CO1	2	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	1	2	1	-	-	-	-	-	-	-	-	-	1	-
CO3	1	2	2	-	-	-	-	-	-	-	-	-	1	-
CO4	1	2	2	-	-	-	-	-	-	-	-	-	1	-
CO5	1	2	2	-	-	-	-	-	-	-	-	-	1	-

(1-Low, 2-Medium, 3-High) (WK8)

**Course Content**

**Unit-I**

**Introduction to Data Structures**

- **Basic Concepts:** Importance of data structures, Overview: system Life Cycle, Algorithm Specification, Data Abstraction, Performance Analysis: Space and Time Complexity.
- **Classification of Data Structures:** Primitive Vs Non-Primitive data structures, Linear Vs Non-linear data structures.

**Programming Exercises**

1. A meteorological department wants to analyze temperature data for a week. Write a program for the following using the known suitable data structure.

- Stores daily temperature readings in an array.
- Calculates the average temperature for the week.
- Identifies the days with the highest and lowest temperatures.

2. A school needs a program to analyze students' grades. Implement an array-based solution to:

- Store the grades of students for a subject.
- Calculate the class average grade.
- Find the highest and lowest grades.
- Count the number of students who passed and failed (assuming passing is 50% or higher).
- 

3. An e-commerce website needs to analyze the prices of products to offer discounts. Write a program to:

- Store product prices in an array.
- Calculate the total cost of all products.
- Identify products that are above a certain price threshold and apply a discount.

- Sort the products by price in ascending order.

4. A flight booking system requires a feature to track seat reservations. Implement an array where:

- Each index represents a seat on the flight (0 for available, 1 for booked).
- Users can book a seat by setting the corresponding index to 1.
- Users can cancel a reservation by setting the index back to 0.
- The program displays the current status of all seats.

## **Unit-II Stacks and Queues**

**The Stack ADT:** Stack Model, Array implementation of stacks, Applications: Balancing the symbols, Infix to Postfix Conversion, Postfix Evaluation.

**The Queue ADT:** Queue Model, Array implementation of Queues, Applications of Queues, Circular Queues

### **Programming Exercises**

1. Implementation of possible operations on stacks using arrays
2. Check for Balanced Brackets in given expression using Stack
3. Convert the given infix expression to postfix using stacks
4. Evaluate given postfix expression using stacks
5. Implementation of possible operations on Queue using arrays
6. Implementation of possible operations on circular queue using arrays

## **Unit-III Linked Lists**

**The List ADT:** Singly Linked List, Representing chains in C.

**Other Linked Lists:** Doubly Linked Lists, Circularly Linked Lists, Linked Stacks and Queues

**Application of Linked List:** Polynomials- Polynomial Representation, Adding of Polynomials.

### **Programming Exercises**

1. Implementation of all possible operations on single linked list.
2. Implementation of all possible operations on double linked list.
3. Implementation of all possible operations on circular linked list.
4. Implementation of possible operations on stacks using list
5. Implementation of possible operations on queue using list
6. Addition of two polynomials using linked list

## **Unit-IV Trees and Graphs**

**Trees:** Introduction-Terminology, Representation of Trees, Binary Trees, Binary Tree Traversals, Binary Search Trees (BST)

**Graphs:** The Graph Abstract Data Type-Introduction, Definition, Graph representation (Adjacency Matrix/List), Elementary Graph Operations- Depth First Search (DFS) and Breadth First Search (BFS).

**Programming Exercises:**

1. Implement tree traversal techniques using recursion.
2. Implementation of Binary search tree operations.
3. Implementation of DFS traversal for the given graph.
4. Implementation of BFS traversal for the given graph.

**Unit-V**

**Sorting and Hashing**

**Sorting:** Merge Sort, Quick Sort, Heap Sort, Comparison of Sorting Techniques

**Hashing:** General Idea, Hash Function, Separate Chaining, Open Addressing, Rehashing, Extendible Hashing

**Programming Exercises**

1. Implement Merge Sort and Quick Sort, Heap Sort.
2. Create a Dictionary using hashing (Use Separate Chaining to handle the collision)
3. Create a Dictionary using hashing (Use open addressing to handle the collision)

**Text Books:**

1. Horowitz, Sahni, and Anderson-Freed, Fundamentals of Data Structures in C, 2nd edition, Universities Press, 2011.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd Edition, Pearson Education, 2002
3. Reema Thereja, Data Structures using C, Oxford Publications, 2nd Edition, 2014.

**Reference Books:**

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B. A. Forouzan, Cengage Learning.
2. Data Structures Through C, Yashavant Kanetkar, 5th Edition, bpb Publications, 2023.
3. Data Structures Using C And C++, Y. Langsam, M. Augenstein And A. M. Tenenbaum, 2nd edition, Prentice - Hall Of India Pvt. Ltd, 2013.
4. Data Structures And Algorithmic Thinking With Python, Narasimha Karumanchi, Career Monk publications, 2016

**Web Resources:**

1. Computer Science and Engineering - Noc: Introduction To Data Structures and Algorithms. <https://archive.nptel.ac.in/courses/106/102/106102064/>
2. Computer Science and Engineering- NOC: Programming, Data Structures and Algorithms <https://archive.nptel.ac.in/courses/106/106/106106127/>
3. Computer Science and Engineering- NOC: Programming, Data Structures and

Algorithms using Python. <https://archive.nptel.ac.in/courses/106/106/106106145/#>

4. Coursera: Data Structures and Algorithms Specialization

5. <https://www.coursera.org/specializations/data-structures-algorithms>

1.

<b>24IT103</b>													
<b>PROBLEM SOLVING WITH PYTHON</b>													
<b>Course Category:</b>	<b>Engineering Science (ES)</b>	<b>Credits:</b>	<b>3</b>										
<b>Course Type:</b>	<b>Integrated Course</b>	<b>Lecture-Tutorial-Practice:</b>	<b>2-0-2</b>										
<b>Pre-requisites:</b>	Programming through C	<b>Continuous Assessment:</b>	<b>40</b>										
		<b>Summative Assessment:</b>	<b>60</b>										
		<b>Total Marks:</b>	<b>100</b>										
<b>Course Description:</b>													
<p>This course introduces fundamental programming and problem-solving concepts using Python. It provides students with a solid foundation for computational thinking, program design, and logical reasoning. Students will learn Python syntax, data handling, and modular design techniques. The course emphasizes problem-solving through programming constructs, functions, data structures, and object-oriented principles.</p>													
<b>Course Objectives:</b>													
<ul style="list-style-type: none"> <li>• Introduce the foundational concepts of Python programming, syntax, and control structures for computational problem solving. (K1)</li> <li>• Explain the use of iteration (for, while, nested loops) and loop control statements (break, continue, pass) along with Python string operations for effective sequential processing. (K2)</li> <li>• Develop proficiency in manipulating strings and core data structures such as lists, tuples, sets, and dictionaries for effective data handling. (K3)</li> <li>• Demonstrate modular program design using user-defined functions, modules, and exception handling for reliable and reusable software development. (K3)</li> <li>• Apply object-oriented programming principles — abstraction, encapsulation, inheritance, and polymorphism — to design efficient and scalable applications. (K4)</li> </ul>													
<b>Course Outcomes:</b>													
At the end of the course, the students will be able to													
<b>CO</b>	<b>Course Outcomes</b>		<b>BTL</b>										
<b>CO1</b>	Explain the syntax, semantics, and control structures of Python programming for solving computational problems. (K1)		K1										
<b>CO2</b>	Explain iteration constructs and loop control statements, with Python string operations for sequential processing. (K2)		K2										
<b>CO3</b>	Implement Python's built-in functions and data structures (lists, tuples, sets, dictionaries) for structured data processing. (K3)		K3										
<b>CO4</b>	Develop modular and reusable programs using functions, packages, and exception handling mechanisms to improve code efficiency and reliability. (K3)		K3										
<b>CO5</b>	Design object-oriented Python solutions applying abstraction, encapsulation, inheritance, and polymorphism for real-world problem-solving. (K4)		K4										
<b>Course Articulation Matrix</b>													
<b>COs</b>	<b>POs</b>											<b>PSOs</b>	
	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PSO1</b>	<b>PSO2</b>
<b>CO1</b>	2	3	2								1	3	2
<b>CO2</b>	3	2	2								2	2	2
<b>CO3</b>	2	2	3								2	2	2
<b>CO4</b>	1	3	3								2	3	1
<b>CO5</b>	1	2	3								2	1	2
<b>(1-Low, 2 -Medium, 3-High)</b>													
<b>Course Content</b>													
<b>Unit-I</b>													
<b>Introduction to Python Programming</b>													
<ul style="list-style-type: none"> <li>• <b>Overview of Python:</b> Features, installation, execution model, Literals, variables, identifiers, and data types, Input/Output operations and indentation, Operators and expressions</li> <li>• <b>Control structures:</b> Conditional and looping statements: Selection: if, if-else, nested if, if-elif-else</li> </ul>													

## Unit-II

### Iterative Statements & String Handling

- **Iterative Statement:** while, for, nested loops, break, continue, pass, else in loops
- **Strings:** Creation, indexing, slicing, concatenation, String formatting, f-strings, and escape sequences, String methods (upper(), lower(), replace(), split(), join()), Membership operators (in, not in) and iteration, String comparison and manipulation

## Unit-III

### Python Data Structures

- **Lists:** Creation, indexing, slicing, mutability, methods (append(), insert(), sort(), remove()), nested lists
- **Tuples:** Creation, immutability, packing/unpacking, tuple operations
- **Sets:** Creating sets, uniqueness, set operations (union(), intersection(), difference()), set comprehension
- **Dictionaries:** Key-value pairs, insertion, deletion, traversal, dictionary methods (items(), keys(), values())

## Unit-IV

### Functions, Modules, and Exception Handling

- **Functions:** Definition, parameters, arguments, and return values, Variable scope and lifetime, Recursion and lambda functions
- **Modules and Packages:** Importing and creating modules and their usage.
- **Exception handling:** try, except, finally, and custom exceptions

## Unit-V

### Object-Oriented Programming(OOP) in Python

- **OOP concepts:** Classes, objects, attributes, and methods, Constructors, destructors, and encapsulation, Inheritance, polymorphism, Aggregation and composition, Real-world applications of OOP to solve engineering problems(Case Study)

## List of experiments

### Unit 1:

1. Install Python on Windows / Ubuntu.
2. Print two strings with custom separator and suppress newline; then print "ENDLINE".
3. Check whether number is Odd or Even.
4. Solve a Quadratic Equation using formula.
5. Find Area of a Triangle.
6. Swap two variables.
7. Convert kilometres to miles.
8. Check Leap Year.
9. Check Armstrong Number.
10. Check if number is Positive, Negative or Zero.
11. Attendance percentage program:  
Input working days, absent days  
Calculate attendance and check eligibility (<75%).
12. Road tax calculation based on cost of bike (Slab-based conditional).
13. Grade calculation based on score range (0.0–1.0).
14. Interactive integer reading:
  - i. Print binary for positive numbers
  - ii. Ignore negative
  - iii. Exit when input = 999

### Unit 2:

1. Nested-loop patterns:
2. Print Multiplication Table using format().
3. Count how many times a number can be divided by 3 until  $\leq 10$ .
4. Sum of first 20 terms of sequence  $1 + 2 + 4 + 8 + 16 + \dots$  using while True + break.
6. Modify 0–100 summation code to use continue and sum only odd numbers.
7. Create student name, roll no, marks inputs and print formatted output using %, %d, %f.
8. Extract substring between two given substrings.
9. Split each word into two halves based on a given percentage.

10. Find all adverbs and their positions in the given string.
11. Extract date from a string using regex (re.search, strptime).

### Unit 3:

1. Accept comma-separated numbers → generate list and tuple.
2. Find tuples divisible by k from list of tuples.
3. Find tuples with all positive elements.
4. Remove duplicate tuples from a list.
5. Row-wise custom addition in a tuple matrix.
6. Extract digits from tuple list → return flat list.
7. Sort dictionary by value (ascending & descending).
8. Convert dictionary string values → int / float.
9. Map each element of list to each item of dictionary (nested dictionary).
10. Convert list → dictionary (odd elements = keys, even = values).
11. Create dictionary from two equal-length lists (keys, values).
12. Input string & pattern → check if characters follow same order.
13. Perform and display union, intersection, difference, and symmetric difference
14. Check whether A is a subset or superset of B
15. Remove Duplicate Words from a Sentence using sets

### Unit 4:

1. Two functions:
  - Area of circle
  - Perimeter of circle
2. Rectangle: area & perimeter via functions.
3. Recursive functions for:
  - GCD
  - LCM
  - Factorial
  - Fibonacci
4. showEmployee(name, salary=10000) — default argument.
5. Function to accept variable number of subject marks (variable-length args).
6. Find day of the week given date (use datetime).
8. Program to perform x / y and handle ZeroDivisionError with message.
9. Input list & n — sum first n values; if n invalid → raise IndexError.
10. Student class:
  - Method: check\_marks
  - Raise custom exception NotEligibleException if marks < 90.
11. Create a module mymath.py with functions:
  - add(a, b)
  - mul(a, b)Write a main program that imports the module and calls these functions.
12. Create a Custom Module With Constants  
Task:  
File: const.py  
Contains:  
PI = 3.14  
GRAVITY = 9.8  
Import it and print both constants.

### Unit 5:

- 1) Design a class **Person** that contains the public attributes:
  - name
  - ageCreate an object and display the stored values.
- 2) Create a class **Person** with the private attributes:

- `__name`
  - `__age`
- Implement:
- A constructor to initialize the attributes
  - A `__str__()` method to return a formatted representation of the object

3) Develop a class **Person** with:

- Private attributes: `__first_name`, `__last_name`, `__age`
- A **class variable** `domain_name` used to auto-generate the email
- A method `fullname()` returning “First Last”
- A `__str__()` method showing full details including the generated email ID

4) Create an abstract class **AbstractStall** with an abstract method `display()`. Implement three derived classes:

- **Stall**
- **ExecutiveStall**
- **PremiumStall**

Each subclass should override `display()` to show details relevant to its type (demonstrating polymorphism).

5) Inheritance with Constructor Execution Order (Employee → Developer)

Define a base class **Employee** and a derived class **Developer**.

Tasks:

- Demonstrate the sequence of constructor calls when the child class is instantiated.
- Print and compare the class attributes of Employee and Developer.

6) Multi-Level OOP with `@property` and Aggregation (Employee → Developer, Manager)

Build a class hierarchy:

- **Employee** class with `@property` for the name attribute
- **Developer** class derived from Employee with an additional attribute (e.g., programming language)
- **Manager** class derived from Employee containing a list of employees managed

Include:

- Methods to `addEmployee()` and `removeEmployee()`
- A separate **Utility** class to generate a formatted report of employees under each manager

7) Operator Overloading (Bank Entry Counter)

Implement a **Bank** class that tracks the number of persons inside a bank.

Overload operators:

- `+` to increment count when a person enters
- `-` to decrement count when a person leaves

This demonstrates operator overloading in a real-world context.

#### Text Books:

1. R. Thareja, *Python Programming: Using Problem Solving Approach*, 2nd ed., Oxford University Press, 2019

#### Reference Books:

1. John Guttag, *Introduction to Computation and Programming Using Python*, MIT Press, 2016.
2. K. Beecher, *Computational Thinking: A Beginner's Guide to Problem-Solving and Programming*, BCS Learning, 2017.
3. L. Ramalho, *Fluent Python*, 2nd ed., O'Reilly Media, 2022.
4. B. Downey, *Think Complexity: Complexity Science and Computational Modeling*, 2nd ed., O'Reilly Media, 2018.
5. S. Hall, *Effective Python: 90 Specific Ways to Write Better Python*, 2nd ed., Addison-Wesley, 2019.

#### Web Resources: [Internet]

1. NPTEL. *The Joy of Computing using Python*. Available from: <https://archive.nptel.ac.in/courses/106/106/106106182/>
2. GeeksforGeeks. *Python Programming Language – Basics*. Available from: <https://www.geeksforgeeks.org/python-programming-language/>
3. W3Schools. *Python Basics Tutorial*. Available from: <https://www.w3schools.com/python/>
4. GeeksforGeeks. *Memory Management in Python*. Available from: <https://www.geeksforgeeks.org/memory-management-in-python/>

5. RealPython. *Understanding Python Objects and Memory Management* . Available from: <https://realpython.com/python-memory-management/>
6. W3Schools. *Python Strings* . Available from: [https://www.w3schools.com/python/python\\_strings.asp](https://www.w3schools.com/python/python_strings.asp)
7. GeeksforGeeks. *Python Data Structures* . Available from: <https://www.geeksforgeeks.org/python-data-structures/>
8. W3Schools. *Python Lists, Tuples, Sets, and Dictionaries* . Available from: [https://www.w3schools.com/python/python\\_lists.asp](https://www.w3schools.com/python/python_lists.asp)
9. RealPython. *Python Functions and Exceptions* . Available from: <https://realpython.com/python-functions/>
10. W3Schools. *Python Exception Handling* . Available from: [https://www.w3schools.com/python/python\\_try\\_except.asp](https://www.w3schools.com/python/python_try_except.asp)
11. AlgoDaily. *Association, Aggregation, and Composition in Python OOP* . Available from: <https://algodaily.com/lessons/association-aggregation-composition-casting>
12. GeeksforGeeks. *Python OOP Concepts* . Available from: <https://www.geeksforgeeks.org/python-oops-concepts/>

Designation	Name in Capitals	Signature with Date
Course Coordinator	Dr. G KALYANI Dr. C S PAVAN KUMAR	
Head of the Department	Dr. M SUNEETHA	

**24ME181**  
**ENGINEERING GRAPHICS**

<b>Course Category:</b>	Engineering Sciences (ES)	<b>Credits:</b>	2.5
<b>Course Type:</b>	Tutorial & Practice	<b>Lecture-Tutorial-Practice:</b>	0-1-3
<b>Pre-requisites:</b>	Nil	<b>Continuous Assessment:</b> <b>Summative Assessment:</b> <b>Total Marks:</b>	60 40 100

**Course Description:**

This course introduces students to the principles and practices of engineering graphics using Computer- Aided Design (CAD) tools. This course covers Plane and Descriptive geometry, orthographic and isometric projections. Students will learn to create, modify, and analyze 2D and 3D drawings, focusing on applications in engineering.

**Course Objectives:**

- To familiarize students with CAD software and its application in technical drawing.
- To make students proficient in creating and editing 2D and 3D engineering drawings.
- To teach the drawing of plane curves and their applications
- To explain the projection of points, lines, Planes, and solids.
- To enable students to visualize and represent 3D objects through orthographic projections, isometric views and development of surfaces.
- To cultivate skills in preparing engineering drawings following standard conventions and practices.

**Course Outcomes:**

At the end of the course, the students will be able to

CO	Course Outcomes	BTL
CO1	Construct plane curves and identify their applications.	K3
CO2	Draw orthographic projections of points, lines and planes	K3
CO3	Develop 2D drawings of solids and surfaces.	K3
CO4	Demonstrate proficiency in creating isometric projections.	K4
CO5	Identify the 3D objects through orthographic projections.	K4

**Course articulation matrix**

COs	POs											PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2		2		2				2				1
CO2	2		2		2				2				1
CO3	2		2		2				2				1
CO4	2		2		2				2				1
CO5	2		2		2				2				1

(1-Low, 2 -Medium, 3-High)

## Course Content

### Unit-I

#### Plane Curves

- Construction of **Ellipse, Parabola and Hyperbola** (General method only)
- Construction of **cycloids & Involutes**.

### Unit-II

#### Orthographic Projections:

- Projections of **Points**.
- Projections of **Lines** (First Angle Projection only)
- Projections of **Plane regular geometric figures**. (First Angle Projection only)

### Unit-III

#### Projection of Solids and Development of Surfaces

- Projections of simple solids such as Cubes, Prisms, Pyramids, Cylinders and Cones with varying positions (Limited to Solid Inclined to one of the Reference planes)
- Development of surfaces of Right Regular Solids of Prism, Pyramid, Cylinder and Cone.

### Unit-IV

#### Conversion to orthographic Projection

- Orthographic projection conventions.
- Conversion of Isometric views into orthographic projections

### Unit-V

#### Conversion to Isometric Projection

- Isometric Projection Principles, Isometric Views of Objects.
- Conversion of Orthographic Projections into Isometric Views

#### Text Books:

1. Kulkarni, D. M., Rastogi, A. P., & Sarkar, A. K. (2009). Engineering graphics with AutoCAD, (revised ed.). PHI Learning Pvt. Ltd.
2. N.D. Bhatt "Engineering Drawing", Charotar Publishing House, Anand. 53<sup>rd</sup> Edition – 2019.

#### Reference Books:

1. Bethine. A. Fishel, Jay D. Helsel, (2014). Engineering Graphics with AutoCAD, Pearson Education.
2. CADFolks, (2023). AutoCAD 2023 for Beginners, CAD Folks Publications.
3. BasanthAgrawal& C M Agrawal," Engineering Drawing", McGraw Hill Education Private Limited, New Delhi.

#### Web Resources:

- [1] [https://www.onlinecourses.nptel.ac.in/noc20\\_me79/preview](https://www.onlinecourses.nptel.ac.in/noc20_me79/preview)
- [2] <https://www.autodesk.com/education>
- [3] <https://www.freecad.org>
- [4] <https://ocw.mit.edu/>

**24CY181  
Chemistry Lab**

<b>Course Category:</b>	<b>Basic Sciences</b>	<b>Credits:</b>	<b>1</b>
<b>Course Type:</b>	<b>Practical</b>	<b>Lecture-Tutorial-Practice:</b>	<b>0-0-2</b>
<b>Pre-requisites:</b>	<b>10+2 Chemistry</b>	<b>Continuous Assessment:</b> <b>Summative Assessment:</b> <b>Total Marks:</b>	<b>60</b> <b>40</b> <b>100</b>

**Course Description:**

The course is designed for students of B.Tech., providing essential practical knowledge of Chemistry concepts relevant to their fields. This hands-on laboratory course provides the students with practical experience in applied chemical techniques and experiments. The course is useful for students to develop essential skills of proper handling of instruments and apparatus, accurate measurements and recording data, as well as interpretation of data to arrive at the correct conclusion. The course includes the experiments based on different instrumental techniques for the quantitative analysis of different solutions. Also, the course enlightens the students on preparation of polymers, and corrosion related experiments. Further, it focuses on the conventional method of quantitative determination of samples using volumetric method.

**Course Objectives:**

1. Enhance understanding of chemical principles involved in instrumental methods of chemical analysis.
2. Develop skills in handling analytical instruments that can measure various chemical parameters.
3. Promote critical thinking and problem-solving through the interpretation of data obtained from the instruments.
4. Familiarize students with advantages and limitations of instrumental methods of analysis compared to conventional volumetric analysis.
5. Explore the practical aspects of chemical processes involved in preparation of polymers, metallic coatings, etc.

**Course Outcomes:**

At the end of the course, the students will be able to

<b>CO</b>	<b>Course Outcomes</b>	<b>BTL</b>
<b>CO1</b>	Demonstrate a comprehensive understanding of various instrumental methods of chemical analysis	<b>K3</b>
<b>CO2</b>	Analyse quantitatively different redox systems and neutralization systems using volumetric analysis	<b>K4</b>
<b>CO3</b>	Compare corrosion tendencies of different metals and their protection by surface coatings	<b>K4</b>

<b>CO4</b>	Apply theoretical knowledge and skills of preparation of polymers, complexes on substrates, adsorption processes, porosity, viscosity, etc.	<b>K3</b>
------------	---	-----------

**Course articulation matrix**

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

(1-Low, 2 -Medium, 3-High) (WK1)

**Course Content**

**Experiments common to all branches:**

1. Determination of strength of acid in a lead-acid battery
2. Determination of strength of base using pH metric titration
3. Conductometric analysis of a base using a standard acid
4. Determination of ferrous iron by permanganometry
5. Comparison of corrosion rates of different metals/alloys
6. Preparation of Urea-formaldehyde resin
7. Determination of amount of iron in a solution by colorimetry
8. Chemistry of blueprinting
9. Adsorption of acetic acid on charcoal

**Experiments for CE and ME branches:**

1. Determination of total hardness of a water sample
2. Determination of calcium in Portland cement
3. Determination of porosity of a refractory material
4. Determination of viscosity of lubricating oil by Redwood viscometer

**Experiments for ECE, EEE and EIE branches:**

10. Determination of ferrous iron by dichrometry
11. Determination of ferrous iron by potentiometry
12. Verification of Lambert-Beer's law
13. Electroplating of copper on iron article

**Experiments for CSE, AI and IT branches:**

10. Determination of ferrous iron by dichrometry
11. Determination of ferrous iron by potentiometry
12. Conductometric analysis of mixture of acids
13. Preparation of conducting polyaniline from aniline

**Text Books:**

1. Mendham, J. (2009). *Vogel's Quantitative Chemical Analysis* (6<sup>th</sup> ed.). Pearson Education.
2. Theodore, J., & George Pope, F. (2021). *Elementary Practical Chemistry. Inorganic and Organic* (1<sup>st</sup> ed.). Legare Street Press.

**Reference Books:**

1. Akhil, N., Deepak, L., Atul, B., & Chaudari, P.B. (2023). *Practical Manual of Inorganic, Organic and Medicinal Chemistry* (1<sup>st</sup> ed.). IP Innovative.
2. Venkateswaran, V. (2012). *Basic Principles of Practical Chemistry* (2<sup>nd</sup> ed.). S. Chand & Sons.

**Web Resources:**

1. [https://nitm.ac.in/ckfinder/userfiles/files/CY%20151\\_Labmanual%20Chemistry%20B\\_Tech%201st%20year.pdf](https://nitm.ac.in/ckfinder/userfiles/files/CY%20151_Labmanual%20Chemistry%20B_Tech%201st%20year.pdf)
2. [http://icv-au.vlabs.ac.in/inorganic-chemistry/Water\\_Analysis\\_Determination\\_of\\_Physical\\_Parameters/](http://icv-au.vlabs.ac.in/inorganic-chemistry/Water_Analysis_Determination_of_Physical_Parameters/)
3. [http://pcv-au.vlabs.ac.in/physical-chemistry/EMF\\_Measurement/](http://pcv-au.vlabs.ac.in/physical-chemistry/EMF_Measurement/)
4. [http://pcv-au.vlabs.ac.in/physical-chemistry/Determination\\_of\\_Viscosity\\_of\\_Organic\\_Solvents/](http://pcv-au.vlabs.ac.in/physical-chemistry/Determination_of_Viscosity_of_Organic_Solvents/)
5. [http://vlabs.iitb.ac.in/vlabs-dev/labs/nitk\\_labs/Environmental\\_Engineering\\_1/experiments/determination-of-chloride-nitk/simulation.html](http://vlabs.iitb.ac.in/vlabs-dev/labs/nitk_labs/Environmental_Engineering_1/experiments/determination-of-chloride-nitk/simulation.html)
6. <https://chemcollective.org/vlabs>

**24ME182**  
**WORKSHOP PRACTICE**  
**(ME)**

<b>Course Category:</b>	Engineering Science (ES)	<b>Credits:</b>	<b>1.5</b>
<b>Course Type:</b>	<b>Laboratory</b>	<b>Lecture-Tutorial-Practice:</b>	<b>0-0-3</b>
<b>Pre-requisites:</b>	<b>10+2 Physics</b>	<b>Continuous Assessment:</b>	<b>60</b>
		<b>Summative Assessment:</b>	<b>40</b>
		<b>Total Marks:</b>	<b>100</b>

**Course Description: (Part-A)**

This part of the course provides first-year engineering students with comprehensive hands-on experience in electrical systems, electronics, and computer science fundamentals. The course covers practical exercises in electrical wiring and circuit design, electronics component handling, and computer hardware and software configuration. Students will engage in activities such as electrical switching circuits, distribution board assembly, electronic component identification, and basic IoT (Internet of Things) projects. The course is designed to equip students with essential skills for diagnosing and constructing electrical and electronic systems, as well as understanding computer hardware and IoT applications.

**Course Description: (Part-B)**

This part of workshop practice course provides first-year mechanical engineering students with hands-on experience across multiple fundamental areas of engineering, including carpentry, welding, and mechanical components. The course is designed to develop practical skills and technical proficiency through targeted exercises in woodworking, metalworking, and mechanical assembly. Students will gain familiarity with essential tools and techniques, including woodworking tools and joinery, welding processes, and mechanical component handling. By engaging in these practical exercises, students will enhance their ability to apply theoretical concepts in real-world scenarios, laying a strong foundation for advanced engineering studies.

**Course Objectives:**

1. Understand the fundamentals of house wiring.
2. Familiarize with the various electronic components and PCB preparation
3. Apply the concepts of internet of things (IOT)
4. Familiarize with the modern manufacturing methods.
5. Understand the basic joints using wood and practice joining of metals using various types of welding and familiarize with various mechanical components such as gears,

Nuts and bolts.

**Course Outcomes:**

At the end of the course, the students will be able to

CO	Course Outcomes	BTL
<b>PART -A</b>		
CO1	Design, assemble, and test various electrical circuits, including stair case, two-way, and fan connections, and accurately measure electrical parameters such as voltage, current, power, and energy using digital meters.	<b>K3</b>
CO2	Identify and use electronic components (resistors, capacitors, diodes, switches) to construct and test a simple calling bell circuit, demonstrating understanding of circuit design and component functionality.	<b>K3</b>
CO3	Install and configure peripheral devices on a desktop system and apply basic IoT principles using an Arduino board to measure and interpret environmental data such as temperature, humidity, and distance.	<b>K3</b>
<b>PART -B</b>		
CO4	Effectively use woodworking hand and power tools, and apply joinery techniques such as cross half lap and dovetail joints to produce precise and durable wood assemblies.	<b>K3</b>
CO5	Proficiently perform arc welding on lap, butt, and corner joints, as well as gas welding processes, while producing high-quality welds with proper safety and technique; additionally, carry out bench work operations such as drilling and tapping, and assemble mechanical components including gears, bearings, and springs, demonstrating competency in precision machining and mechanical assembly.	<b>K3</b>

**Course articulation matrix**

COs	POs										
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1			2		3						
CO2			2		3						
CO3			2		3						
CO4			2		3						
CO5			2		3						

**(1-Low, 2 -Medium, 3-High)**

**Course Content**

**PART –A**

**1. Electrical Workshop Practice Exercises: (3hrs)**

- a) Stair case/Two-way /Gowdown Switching/Fan connection
- b) 3Φ Distribution board with fuse, MCB, two way switches, light indicator, 4 switches, fan regulator
- c) Measurement of voltage, current, power, Energy by using digital meters.

2. **Electronics Workshop Practice Exercises: (1 hr)**
  - a) Familiarization of electronic components such as resistors, capacitors, diodes, and switches.
  - b) Preparation of a simple calling bell circuit board and testing its operation.
3. **Computer science Workshop Practice Exercises: (2 hrs)**
  - a) **Hardware components of a desktop system:** Peripheral Installation: Install and configure peripheral devices such as printers, scanners, and external hard drives.
  - b) **Basics of Internet of things (IOT):** Demonstration of different components and pin configuration of Arduino board.
  - c) To measure Temperature, humidity and distance using Arduino board.
4. **Modern Manufacturing methods:** Demonstration of 3-D printing process.

**PART –B Branch specific (For ME students only)**

5. **Carpentry Workshop Practice Exercises: (2 hrs)**
  - a) Woodworking Tools Familiarization: Practice to work with woodworking hand tools and operations using power tools.
  - b) **Joinery Techniques:** Practice various joinery techniques such as cross half lap and Dove tail joints.
  - c) Practice to work with various wood adhesives and fasteners.
6. **Welding workshop practice exercises: (2 hrs)**
  - a) Preparation of Lap, Butt & Corner joints using Arc Welding.
  - b) Practice of Gas welding process.
7. **Introduction to Mechanical Components & operations: (3hrs)**
  - a) **Bench Work Techniques:** Practice to work with various materials like metals and plastics.
  - b) Practice of drilling, tapping, and reaming operations on the above materials.
  - c) **Mechanical Components:** Practice of assembling lap and butt joints with various types of bolts, nuts, screws and single, double studs.
  - Practice to work with various types of mechanical components such as
    - d) Gears - Bevel, Spur, Helical and Worm.
    - e) Bearings – Ball, roller, plain, needle and flexure.
    - f) Springs – Compression, Extension, Conical, Torsion and leaf.

**Text Books:**

1. Basic Workshop Technology: Manufacturing Process, Felix W.; Independently Published, 2019. Workshop Processes, Practices and Materials; Bruce J. Black,

Routledge publishers, 5th Edn. 2015.

2. A Course in Workshop Technology Vol I. & II, B.S. Raghuwanshi, Dhanpath Rai& Co., 2015 & 2017.

**Reference Books:**

1. Elements of Workshop Technology, Vol. I by S. K. Hajra Choudhury & Others, Media Promoters and Publishers, Mumbai. 2007, 14th edition.
2. Workshop Practice by H. S. Bawa, Tata-McGraw Hill, 2004.
3. Wiring Estimating, Costing and Contracting; Soni P.M. &Upadhyay P.A.; AtulPrakashan, 2021-22.

**Web Resources:**

1. <https://dsceme.files.wordpress.com/2016/08/workshop-practice-manual-2016-17-1.pdf>
2. <https://www.protosystech.com/rapid-prototyping.htm>
3. <https://www.arduino.cc/en/Tutorial/Foundations>
4. <https://www.tutorialspoint.com/arduino/>

**24UC183**  
**Sports and Yoga**

<b>Course Category:</b>	<b>Mandatory (MC)</b>	<b>Credits:</b>	<b>0</b>
<b>Course Type:</b>	<b>Practical</b>	<b>Lecture-Tutorial-Practice:</b>	<b>0-0-3</b>
<b>Pre-requisites:</b>	-	<b>Continuous Assessment:</b>	<b>100</b>
		<b>Summative Assessment:</b>	<b>-</b>
		<b>Total Marks:</b>	<b>100</b>

**Course Description:**

The course is designed for first year B.Tech. students of all branches. It provides essential and applied knowledge of physical and mental health. Introducing the course is more crucial in the present days, particularly undergraduate students, who require awareness on their physical and mental health. With this focus, the course covers various aspects of physical fitness, wellness and lifestyle, fundamental aspects of anatomy, physiology in physical education, yoga and lifestyle, different types of sports, etc. Although, the basic theoretical aspects of sports and yoga are discussed, major focus will be on the practice of yoga asanas and participation in various sports by the students.

**Course Objectives:**

1. Make the students understand the importance of sound health and fitness principles as they relate to better health.
2. Expose the students to a variety of physical and yogic activities aimed at stimulating their continued inquiry about yoga, physical education, health and fitness.
3. Create a safe, progressive, methodical and efficient activity-based plan to enhance improvement and minimize risk of injury.
4. Develop among students an appreciation of physical activity as a lifetime pursuit and a means to better health.

**Course Outcomes:**

At the end of the course, the students will be able to

<b>CO</b>	<b>Course Outcomes</b>	<b>BTL</b>
<b>CO1</b>	Apply the knowledge of physical activities and Hatha Yoga on himself/herself and practice them	<b>K3</b>
<b>CO2</b>	Understand basic skills associated with yoga and physical activities including strength, flexibility, balance and coordination	<b>K2</b>
<b>CO3</b>	Practice yoga asanas, pranayama and meditation in daily life for the health of body and balance of mind	<b>K3</b>
<b>CO4</b>	Apply correctly the biomechanical and physiological principles related to exercise and training	<b>K3</b>

<b>CO5</b>	Practice techniques for increasing concentration and decreasing anxiety which leads to stronger academic performance	<b>K3</b>
------------	--	-----------

**Course articulation matrix**

COs	POs										
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
<b>CO1</b>	-	-	-	-	-	-	1	2	-	-	3
<b>CO2</b>	-	-	-	-	-	-	1	2	-	-	3
<b>CO3</b>	-	-	-	-	-	-	1	2	-	-	3
<b>CO4</b>	-	-	-	-	-	-	1	2	-	-	3
<b>CO5</b>	-	-	-	-	-	-	1	2	-	-	3

**(1-Low, 2 -Medium, 3–High) (WK8)**

**Course Content**

**Unit-I**

**Introduction to physical education, physical fitness, wellness and life style**

- Meaning and definition of physical education, aims and objectives, changing trends in physical education.
- Meaning and importance of physical fitness and wellness, components of physical fitness.
- Components of health related fitness, components of wellness.
- Preventing health threats through lifestyle change and concept of positive lifestyle.
- Awards and honours in the field of sports in India (Dronacharya Award, Arjuna Award, Dhayanchad Award, Rajiv Gandhi Khel Ratna Award, etc.)

**Unit-II**

**Fundamentals of anatomy, physiology, kinesiology, biomechanics and sports**

- Meaning of anatomy, physiology and their importance in the context of physical education and sports.
- Effect of exercise on the functioning of various body systems like circulatory system, respiratory system, neuromuscular system, etc.
- Meaning and importance of kinesiology and biomechanics in physical education and sports, Newton's law of motion and its application in sports, friction and its effects in sports.
- Concept of postures, causes of bad posture, advantages of correct posture, advantages and disadvantages of weight training, common postural deformities – knock knee, flat foot, round shoulders, lordosis, kyphosis, bow legs and scoliosis.

**Unit-III**  
**Psychology in Sports**

- Definition and importance of psychology in physical education and sports, differentiation between growth and development.
- Adolescent problems and their management, concept and types of emotions, control of emotions, meaning and types of aggressions in sports, psychological benefits of exercise.
- Anxiety and fear and their effects on sports performance, motivation – types and techniques, understanding stress and coping strategies.
- Concept and meaning of doping, prohibited substances and methods, side effects of prohibited substances.
- First aid – definition and objectives, sports injuries – classification, causes and prevention, management of injuries – soft tissue injuries and bone and joint injuries.

(Following subtopics related to any one game/sport of choice of student out of: Athletics, Badminton, Basketball, Chess, Cricket, Kabaddi, Lawn Tennis, Swimming, Table Tennis, Volleyball, etc.: History of the game/sport, latest general rules of the game/sport, specifications of play fields and related sports equipment, important tournaments and venues, sports personalities, proper sports gear and its importance.)

**Unit-IV**  
**Theoretical aspects of Yoga and practice of Asanas**

- Meaning, history and basic concepts of yoga, three bodies of human system, panchakoshas, trigunas, and ashtanga yoga.
- Meaning of asanas, practice of sukshma vyayama (warm up for asanas), asanas as preventive measures, practice of various asanas in the four categories – standing, sitting, prone and supine – at least 5 postures from each category, variations, contraindications and benefits of the asanas, Sun salutations.
- Asanas for hypertension, back pain, diabetes, asthma, obesity, etc.

**Unit-V**  
**Theory and practice of pranayama and meditation**

- Meaning of pranayama, instructions for practice of pranayama, practice of various pranayama techniques – abdominal breathing, kapalabhati, anulom-vilom pranayama, bhramari, etc.
- Relaxation techniques for improving concentration – Yognidra
- Meaning of meditation, difference between concentration and meditation, various meditation methods, heartfulness meditation practice, benefits of meditation.

**Text Books:**

1. Modern trends and physical education – Prof. Ajmer Singh.
2. Light on Yoga – B.K.S. Iyengar.

**Reference Books:**

1. Health and Physical Education – NCERT (11<sup>th</sup> and 12<sup>th</sup> Classes).
2. Light on Pranayama – B.K.S. Iyengar.

**Web Resources:**

1. <https://ncert.nic.in/textbook/pdf/iehp103.pdf>
2. [https://www.physio-pedia.com/Biomechanics\\_In\\_Sport](https://www.physio-pedia.com/Biomechanics_In_Sport)
3. [https://en.wikipedia.org/wiki/Sport\\_psychology](https://en.wikipedia.org/wiki/Sport_psychology)
4. [https://www.physio-pedia.com/Principles\\_of\\_Yoga](https://www.physio-pedia.com/Principles_of_Yoga)
5. <https://ncert.nic.in/pdf/publication/otherpublications/tiyhwlp1.pdf>
6. <https://heartfulness.org/global>
7. <https://www.biharyoga.net/asana-and-pranayama.php>