RN SHETTY TRUST ®

RNS INSTITUTE OF TECHNOLOGY

Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi. Accredited by NAAC 'A+' Grade
RR Nagar, Dr. Vishnuvardhan Road, Bengaluru - 560 098
Ph:(080)28611880,28611881 URL: www.rnsit.ac.in



Dr. R N Shetty
Founder

ESTD.: 2001
An Institute with a Difference



25 RNSIT

DEPARTMENT OF CSE - CYBER SECURITY

BACHELOR OF ENGINEERING

Scheme and Syllabus of III & IV Semesters

2025 SCHEME

INSTITUTION'S INNOVATION COUNCIL

www.rnsit.ac.in

Scheme and Syllabus
Batch 2025 - 2029

(As per NEP-2020)
(Effective from the academic year 2024 – 25)

About RNSIT

RNS Institute of Technology, familiar as RNSIT was established in the year 2001. It is the brainchild of Dr. R. N. Shetty, a well-known industrialist, and a great philanthropist. The institution was established with the sole purpose of imparting quality education in Engineering, Technology and Management. Being one of the premier engineering colleges in the country, RNSIT is marching towards placing itself in the global map by excelling in academics and placements.

Lush green campus, state of the art infrastructure, well equipped laboratories, hundreds of activities, numerous new initiatives, consistent VTU ranks, path breaking placements, cultural extravaganza, Sports events, 5600 + students and 300 + faculty – with all these RNSIT has become dream destination for engineering aspirants across the country.

Vision

Building RNSIT into a World Class Institution.

Mission

To impart high quality education in Engineering, Technology and Management with a difference, enabling students to excel in their career by

- Attracting quality Students and preparing them with a strong foundation in fundamentals so as to achieve distinctions in various walks of life leading to outstanding contributions.
- Imparting value based, need based, choice based and skill based professional education to the aspiring youth and carving them into disciplined, World class Professionals with social responsibility.
- Promoting excellence in Teaching, Research and Consultancy that galvanizes academic consciousness among Faculty and Students.
- Exposing Students to emerging frontiers of knowledge in various domains and make them suitable for Industry, Entrepreneurship, Higher studies, and Research & Development.
- Providing freedom of action and choice for all the Stakeholders with better visibility

.

About CSE (Cyber Security)

The Department of CSE – Cyber Security was started with the notion of establishing a complete cyber security ecosystem. In an era of exponential digital expansion, the corporate landscape has transitioned into a digitally driven operational framework. This metamorphosis has ushered in a parallel surge in cyber threats, compelling organizations to fortify their cybersecurity posture. As the demand for cybersecurity expertise reaches new heights, our program, in collaboration with industry luminaries, offers a specialized curriculum. It equips students with cutting-edge skills, spanning cryptography, API security, software fortification, network defense, identity and access management, cloud security, hardware safeguards, malware forensics, ransomware mitigation, vulnerability assessments, penetration testing, and more. This program empowers graduates to become agile defenders against the relentless evolution of cyber threats, ensuring the security of invaluable organizational assets in the contemporary industrial landscape.

Vision

To be a global leader in imparting Cyber Security education, research & development and empowering young minds to safeguard the digital world

Mission

- Empower students with essential Cyber Security through hands-on facilities and a strong ethical foundation.
- Facilitate collaborative learning environment, teamwork, and global certifications for real-time challenges.
- Drive innovation in Cyber Security through state-of-the-art research, fostering a culture of exploration through higher learning and entrepreneurship.
- Promote students to possess qualities of interpersonal, interdisciplinary, leadership, and societal responsibilities.

Programme Educational Objectives

A Graduate of Computer Science and Engineering (Cyber Security) will be able to:

- **PEO1**: Demonstrate technical skills and problem-solving capabilities in the field of cyber security, effectively addressing complex challenges.
- **PEO2**: Develop strong communication skills, collaborative spirit to convey technical concepts and findings effectively to stakeholders, to foster employability in cyber security and related fields.

- **PEO3**: Incorporate ethical, legal, and social aspects into their professional practice, recognizing the implications of their actions on individuals, organizations, and society.
- **PEO4**: Actively engage in activities that foster the continuous development of their computing and cyber security skills, staying abreast of emerging technologies, best practices, and evolving threats to maintain their effectiveness as professionals in the field.

Program Outcomes (Revised)

1. **PO1: Engineering Knowledge**: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
2. **PO2: Problem Analysis**: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
3. **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)
4. **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).
5. **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)
6. **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).
7. **PO7: Ethics**: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
8. **PO8: Individual and Collaborative Teamwork**: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

9. **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

10. **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.

11. **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)

Programme Specific Outcomes (PSOs)

The CSE – Cyber Security graduates will have

1. **PSO1**: Proficiency in applying principles of software development to deliver efficient software solutions.
2. **PSO2**: Competency in securing information, systems, and networks through the implementation of cybersecurity best practices.

B.E. in CSE (Cyber Security)
Autonomous Scheme
III – IV Semester

B.E. in Computer Science and Engineering (Cyber Security): Autonomous Scheme and Syllabus
III SEMESTER

Sl. No	Course	Course Code	Course Title	Teaching Department (TD)and Question Paper setting Board (PSB)s		Teaching & Learning Scheme					Examination				Credits
						CI		I	TW & SL	# TH/S	Duration in hours	CIE Marks	SEE Marks	Total Marks	
						L	T	P	SL						
1	ASC	25BCSTH301	STATISTICS ANDPROBABILITY DISTRIBUTIONS	TD: CS ALLIED/MATHS PSB: CS ALLIED		45	30	0	45	120	3	50	50	100	4
2	IPCC	25BCSIC302	OBJECT ORIENTED PROGRAMMING USING JAVA	CSE-CY		45	0	30	45	120	3	50	50	100	4
3	PCC	25BCSTH303	LOGIC DESIGN AND COMPUTER ORGANIZATION	CSE-CY		45	30	0	45	120	3	50	50	100	4
4	PCC	25BCSTH304	DATA STRUCTURES AND APPLICATIONS	CSE-CY		45	0	0	45	90	3	50	50	100	3
5	PCC	25BCSTH305	OPERATING SYSTEMS	CSE-CY		45	0	0	45	90	3	50	50	100	3
6	PCCL	25BCSLB306	DATA STRUCTURES LABORATORY	CSE-CY		0	0	30	00	30	3	50	50	100	1
7	AEC	25BCYAE307X	ABILITY ENHANCEMENT COURSE (LABORATORY) - BUCKET LIST B	CSE-CY		0	0	30	00	30	3	50	50	100	1
8	SDC	25BCBPR308	COMMUNITY PROJECT (PROJECT-BASED LEARNING) /SOCIETAL PROJECT	CSE-CY		0	0	0	30	30	3	50	50	100	1
9	NCMC	25BNSSP309	NATIONAL SERVICE SCHEME	Campus	NSS coordinator	0	0	0	30	30	--	100	-	100	PP
		25BPEDU309	PHYSICAL EDUCATION		Physical Education Director										
		25BYOGA309	YOGA		Yoga Teacher										
		25BMUSK309	MUSIC		Music Teacher										
Total										**660**		**500**	**400**	**900**	**21**
10	NCMC	25BMADP310	ADDITIONAL MATHEMATICS - I	TD -Maths Dept		14	0	0	14	28	3	100	-- -	100	PP
Total										**688**		**600**	**400**	**1000**	**21**
11	SKLAB	25BCYSL001	SKILL LAB : CYBERSECURITY FOUNDATIONS & SECURITY TOOLS LAB	Respective Engineering Dept.		0	0	30	0	30	3	50	50	100	1

BUCKET LIST B	
25BCYAE307A	**PROJECT MANAGEMENT WITH GIT**
25BCYAE307B	**DEVOPS-1**

B.E. in Computer Science and Engineering (Cyber Security): Autonomous Scheme and Syllabus
IV SEMESTER

Sl. No	Course and Course Code		Course Title	Teaching Department (TD)and Question Paper setting Board (PSB)		Teaching & Learning Scheme					Examination				Credits
						CI		LI	# TW & SL	TH/S	Duration in hours	CIE Marks	SEE Marks	Total Marks	
						L	T	P							
1	ASC	25BCSTH401	DISCRETE MATHEMATICS AND GRAPH THEORY	TD: CS ALLIED/ MATHS PSB: CS ALLIED		45	30	0	45	120	3	50	50	100	4
2	IPCC	25BCSIC402	DATABASE MANAGEMENT SYSTEMS	CSE-CY		45	0	30	45	120	3	50	50	100	4
3	PCC	25BCSTH403	DESIGN AND ANALYSIS OF ALGORITHMS	CSE-CY		45	30	0	45	120	3	50	50	100	4
4	PCC	25BCYTH404	ELEMENTS OF CYBER SECURITY	CSE-CY		45	30	0	45	120	3	50	50	100	4
5	PCCL	25BCSLB405	DESIGN AND ANALYSIS OF ALGORITHMS LAB	CSE-CY		0	0	30	00	30	3	50	50	100	1
6	AEC	25BCYAE406X	ABILITY ENHANCEMENT COURSE (LABORATORY) - BUCKET LIST C	CSE-CY		0	0	30	00	30	3	50	50	100	1
7	BSC	25BFETH407	BIOLOGY FOR ENGINEERS	TD: CHEMISTRY/BIOTECH/CS ALLIED; PSB: CS ALLIED		30	0	0	30	60	2	50	50	100	2
8	SDC	25BESPR408	ENVIRONMENTAL SCIENCE PROJECT	CSE-CY		0	0	0	30	30	3	50	50	100	1
9	NCMC	25BNSSP409	NATIONAL SERVICE SCHEME (NSS)	Campus	NSS coordinator	0	0	0	30	30	--	100	---	100	PP
		25BPEDU409	PHYSICAL EDUCATION (PE)		Physical Education Director										
		25BYOGA409	YOGA		Yoga Teacher										
		25BMUSK409	MUSIC		Music Teacher										
Total										**660**		**500**	**400**	**900**	**21**
10	NCMC	1BMADP410	ADDITIONAL MATHEMATICS - II	TD -Maths Dept		14	0	0	14	28	--	100	--	100	PP
Total										**688**		**600**	**400**	**1000**	**21**
11	SKLAB	25BCYSL002	SKILL LAB (BUCKET LIST SDL)	Respective Engineering Dept.		0	0	30	0	30	3	50	50	100	1

BUCKET LIST C			
25BCYAE406A	**ADVANCED JAVA**	**25BCYAE406B**	**INTRODUCTION TO UNIX**
25BCYAE406C	**DEVOPS-2**	**25BCYAE406D**	**INTRODUCTION TO CYBER SECURITY TOOLS**

B.E. in CSE (Cyber Security)
Autonomous Syllabus
III Semester

B.E. (Common to all CSE streams)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

Course Title:	**STATISTICS AND PROBABILTY DISTRIBUTIONS**		
Course Code:	**25BCSTH301**	**CIE MARKS**	**50**
(L: T: P) +SL	**(3:2:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:
The purpose of the course is to facilitate the learner to:

1. Apply curve fitting, correlation, regression analysis, and sampling techniques to solve engineering and data analysis problems.
2. Analyze probability concepts and probability distributions to model and interpret uncertainty in engineering systems.
3. Familiarize with the fundamental concepts of probability theory and probability distributions to build a strong foundation for engineering and data analysis applications.
4. Solve engineering problems involving random phenomena using stochastic processes and probabilistic modelling techniques.

Module-1 **9 Hours**

Curve fitting, Correlation and Regression analysis
Curve Fitting – Curve fitting by the method of least squares- fitting the curves of the form $y = ax + b$, $y = ax^b$ and $y = ax^2 + bx + c$.
Correlation and regression – Karl Pearson's coefficient of correlation and rank correlation, problems. Regression analysis, lines of regression, problems.
Text book 1: Chapter-24 (24.3 - 24.6(1), 24.8) Chapter-25 (25.12 - 25.14, 25.16)

Module-2 **9 Hours**

Sampling Theory
Introduction to sampling theory, standard error, Type-I and Type-II errors. Test of hypothesis for means- Z test, student's t- test, chi-square test of goodness of fit.
Text book 1: Chapter-27 (27.1 - 27.8, 27.12 – 27.18)

Module-3 **9 Hours**

Probability Theory
Definition of Probability, Mutually exclusive and Independent events, Addition law of Probability, Independent events- Multiplication law of probability, Conditional Probability, Bayes' Theorem and Problems.
Text book 1: Chapter-26 (26.1 - 26.6)

Module-4 **9 Hours**

Probability Distribution
Random variable, discrete and continuous probability distribution, expectation, mean and variance Binomial distribution, Poisson distribution, Normal distribution, Exponential distribution, (Derivation for Mean and Standard Deviation of Binomial and Poisson Distribution only).
Text book 1: Chapter-26 (26.7 - 26.10, 26.14 – 26.17, 26.19(6))

Module-5	9 Hours
Stochastic process Joint Probability Distribution (Discrete Random Variable only), Stochastic process, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains (Finite, discrete only), higher transition probability. **Text book 1: Chapter-31**	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Employ correlation and regression methodologies to model complex relationships.
CO2	Integrate statistical tools and inferential methods to analyse engineering systems and make data-driven decisions with quantified confidence.
CO3	Develop probabilistic models using discrete and continuous distributions to quantify uncertainty.
CO4	Apply probabilistic models and statistical reasoning to analyse random phenomena.
CO5	Compute joint probability distributions, determine state transition probabilities in Markov chains, and apply stochastic modelling methods for decision-making and forecasting.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 30 marks.
Continuous and Comprehensive Evaluation (B)	20	-	Any two Assessment methods as per 24RNBE4.2 (*if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	50	20	CIE Evaluation is done for a maximum of 50 marks
SEE (C)	50	18	SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components **A, B and C.**
• The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE			

Text Books:

1. B. S. Grewal, Higher Engineering Mathematics, 42nd Edition, Khanna Publishers, 2017.
2. B V Ramana, Higher Engineering Mathematics, Tata McGraw-Hill, 2006
3. Montgomery, D. C. & Runger, G. C., Applied Statistics and Probability for Engineers, 7th Edition., Hoboken, NJ: John Wiley & Sons, 2018.

Reference Books:

1. Gupta, S. C., 'Fundamentals of Statistics', Himalaya Publishing House, 8th Edition, 2023.
2. Kreyszig, "Advanced Engineering Mathematics", John Wiley & Sons, 10th Edition., 2018.
3. N.P Bali and Manish Goyal, "A Textbook of Engineering Mathematics" Laxmi Publications, 10th Edition., 2022.
4. Srimanta Pal & Subodh C. Bhunia, "Engineering Mathematics" Oxford University Press, 3rd Edition., 2016.
5. J. K. Blitzstein and J. Hwang, Introduction to Probability, 2nd Edition., Boca Raton, FL, USA: CRC Press, 2019.

Web links and Video Lectures (e-Resources):

- http://academicearth.org/
- VTU e-Shikshana Program
- VTU EDUSAT Program
- https://nptel.ac.in/courses/111105160
- https://nptel.ac.in/courses/127106019
- https://ocw.mit.edu/courses/18-335j-introduction-to-numerical-methods-spring-2019/
- https://ocw.mit.edu/courses/18-330-introduction-to-numerical-analysis-spring-2012/pages/syllabus/

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

1. Programming Assignment
2. Tool/Software Exploration
3. Problem-Based Learning (PBL)
4. Assignments

B.E (CSE and Allied Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

Course Title:	**OBJECT ORIENTED PROGRAMMING USING JAVA**		
Course Code:	**25BCSIC302**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:0:2) + (30 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Identify and recall core object-oriented programming (OOP) concepts and Java language features.
2. Construct Java programs using methods, exception handling, string manipulation, and object-oriented features to solve programming problems.
3. Develop multithreaded Java applications using synchronization methods.
4. Implement efficient data management using Java Collections Framework, including Lists, Sets, Maps, Iterators, and related interfaces.

Module-1 **9 Hours**

Introducing Classes: Class fundamentals, declaring objects, assigning object reference variables, introducing methods, constructors and this keyword and garbage collection.

Methods and Classes: Overloading methods, objects as parameters, argument passing, returning objects, recursion, access control, understanding static, introducing final, introducing nested and inner classes.

Textbook 1: Chapter-6 (6.1 - 6.7) Chapter-7 (7.1 - 7.7,7.9)

Module-2 **9 Hours**

Inheritance and Interfaces: Inheritance basics, using super, creating a multilevel hierarchy, constructor execution, method overriding, dynamic method dispatch, abstract classes, using final with inheritance, local variable type inference, the Object class, creating and implementing interfaces, extending interfaces and using interfaces for multiple inheritance.

Textbook 1: Chapter-8 , Chapter-9 (9.4.1 - 9.4.2, 9.4.6)

Module-3 **9 Hours**

Exceptions: Exception-handling fundamentals, exception types, try and catch, multiple catch clauses, throw, throws, finally and user-defined exception subclasses.

Multithreaded Programming: Main thread, Creating threads, Multiple threads, Using isAlive() and join() and Synchronization.

Textbook 1: Chapter-10 (10.1,10.2,10.4,10.5,10.7 - 10.9,10.11) , Chapter-11 (11.1 - 11.5,11.7)

Module-4 **9 Hours**

String Handling: String constructors, string length, special string operations, character extraction, string comparison, searching strings, modifying strings, changing case of characters, joining strings, Data Conversion Using valueOf( ), StringBuffer and StringBuilder.

Textbook 1: Chapter-18 (18.1 - 18.10,18.12,18.13)

Module-5 **9 Hours**

The Collections and Framework: Collections overview, The Collection Interface List and Set, The Collection Classes ArrayList, LinkedList, HashSet, LinkedHashSet and TreeSet, accessing a collection Via an Iterator, The Random Access Interface and Working with Maps
Textbook 1: Chapter-20 (20.1, 20.2.3 - 20.2.4, 20.3.1 - 20.3.5, 20.4, 20.8)

PRACTICAL COMPONENT OF IPCC

Program 1: Matrix Addition Using Command-Line Arguments
A university is developing a utility program for performing matrix operations. The program should accept the size of a square matrix **N × N** from the command-line arguments. After receiving the value of N, the program should read two matrices from the user, compute their sum, and display the resulting matrix.
Program 2: Implementation of Stack Using Arrays
A software application requires the implementation of a stack data structure to store integer values. The stack should have a maximum capacity of 10 integers. The system must support standard stack operations such as push, pop, and display, and demonstrate the functionality using a Java main method.
Program 3: Employee Management Using Classes and Objects
A company maintains employee records for its staff members. Each employee has details such as **ID, name, and salary**. The system should allow increasing an employee's salary by a given percentage using a method called raiseSalary(percent) and display the updated employee details
Program 4: Implementation of Abstract Classes for Geometric Shapes
A software system is being developed to handle different geometric shapes. Each shape should be able to calculate its **area and perimeter**, but the formula differs for each shape. Therefore, a common abstract class is required to define the structure, while specific implementations are provided by individual shape classes like **Circle and Triangle**.
Program 5: Implementation of Interfaces Using a Rectangle Class
A graphics software system is being developed where different shapes need to be dynamically resized. The system requires a common contract that allows resizing of width and height for shapes. A class representing a **Rectangle** should implement this contract and allow modification of its dimensions.
Program 6: Demonstration of Inner Classes in Java
A software system is designed to demonstrate the concept of inner classes in Java. The system contains an outer class and an inner class, and both classes define a method named display(). The program should demonstrate how both methods are executed separately to show the difference between outer and inner class behavior.
Program 7: Custom Exception Handling for Division by Zero
A calculator application is being developed where division operations are performed by users. The system must ensure that division by zero is not allowed. If a user attempts to divide a number by zero, a **custom exception (DivisionByZero)** should be raised and handled using try, catch, throw, and finally blocks.

Program 8: Multithreading Using the Runnable Interface
A system is being developed to demonstrate multithreading in Java using the **Runnable interface**. The program should create a thread, execute a task inside the run() method, start the thread using start(), and pause execution using sleep(500) to simulate time delay between outputs.
Program 9: String Manipulation Using Java String Methods
A text processing system is being developed to demonstrate various built-in **String operations in Java**. The program should perform character extraction, string comparison, string searching, and string modification using standard String methods.
Program 10: Palindrome Validation Using SOLID Principles
A software system is being designed to validate strings based on different rules. One such rule is to check whether a given string is a **palindrome**. The application should be designed using **SOLID principles** so that each class has a single responsibility and the system is easy to extend and maintain.
Program 11: String Compression Using Run-Length Encoding
A text processing application is being developed to optimize storage of strings by compressing repeated consecutive characters. For example, repeated characters in a string should be replaced with the character followed by its count.
A college is organizing a technical seminar and maintains a list of students who have registered for the event. During registration, some students may accidentally register more than once. The system must ensure that each student is stored only once in the list. Therefore, a suitable collection that does not allow duplicate entries is required to store and display the list of unique student names.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	**Understand** Java classes, data members, and methods to develop solutions for specified programming scenarios.
CO2	**Apply** the concepts of inheritance and interfaces to solve real-world programming problems.
CO3	**Construct** Java programs using multithreading and exception handling techniques to manage concurrency and runtime exceptions.
CO4	**Demonstrate** string handling methods for manipulating and processing textual data in Java applications.
CO5	**Implement** the Java Collections Framework using Collection interfaces, Lists, Sets, Maps, Iterators, Comparators, and user-defined objects for efficient data management

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details

CIE-IA Tests(A)	20	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 20 marks.
Continuous and Comprehensive Evaluation (B)	10	-	one Assessment methods as per 24RNBE4.2 scaled to 10 marks.
Total CIE Theory (A+B)	**30**	**12**	**Marks of Tests and Assignments awarded for a maximum of 30.**
CIE Practical	12	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 12 marks.
CIE Practical Test	8	-	Average of marks of two tests, each conducted for 100 marks covering all experiments, scaled to 8 marks.
Total CIE Practical (C)	**20**	**8**	Marks of experiments, record, and test awarded for a maximum of 20 marks.
CIE: Theory + Practical (A+B+C)	**50**	**20**	CIE evaluation is done for a maximum of 50 marks.
SEE (D)	**50**	**18**	SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C+D)	**100**	**40**	Final marks shall be the sum of marks scored in components A, B, C, and D.
The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE. The laboratory component of the ESC shall be for CIE only. However, in SEE, the questions from the laboratory component shall be included in their respective modules only.			

Textbooks:

1. Herbert Schildt, The Complete Reference (Java), 12th Edition, McGraw-Hill, 2021.

Reference Books:

1. E. Balagurusamy, Programming with Java, 6th Edition, McGraw-Hill, 2019.
2. Bruce Eckel, Thinking in Java, 4th Edition, Prentice Hall, 2006.

Web Links and Videos:

1. https://www.youtube.com/watch?v=5gWpVJc0UG4
2. https://www.youtube.com/watch?v=-HafzawNlUo

3. https://www.youtube.com/watch?v=K3g4srbkUNM

4. https://www.youtube.com/watch?v=iUzvdDuAeSQ

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

1. LeetCode
2. Practical or programming assignments
3. Seminar / Project

Self–Study Topics:

1. SOLID Principles (GeeksforGeeks)

ESTD : 2001
An Institute with a Difference

B.E (CSE and Allied Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

Course Title:	**LOGIC DESIGN AND COMPUTER ORGANIZATION**		
Course Code:	**25BCSTH303**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:2:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Simplify digital circuits by applying the principles of Boolean algebra, fundamental logic gates and advanced gate-level minimization techniques.
2. Design, synthesize and analyze basic combinational and sequential digital circuits, including arithmetic units, data routers, latches, flip-flops and registers.
3. Comprehend basic computer architecture by evaluating functional unit interactions, bus structures, instruction sequencing and performance metrics.
4. Formulate efficient input/output strategies and memory hierarchies by analyzing interrupt handling, Direct Memory Access (DMA) and cache mapping operations.
5. Examine internal processing unit dynamics by tracking the step-by-step execution of complete instructions alongside basic pipelining concepts.

Module-1 **9 Hours**

Digital Systems and Binary Numbers: Binary Logic.
Boolean Algebra and Logic Gates: Axiomatic Definition of Boolean Algebra, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Digital Logic Gates.
Gate Level Minimization: Introduction, The Map Method, Four-Variable Map, Don't-Care Conditions, NAND and NOR Implementation, Exclusive-OR Function.
Textbook 1: Chapter-1 (1.9), Chapter-2 (2.3 - 2.5, 2.8), Chapter-3 (3.1, 3.2,3.3, 3.5, 3.6, 3.8)

Module-2 **9 Hours**

Combinational Logic: Introduction, Combinational Circuits, Design Procedure, Binary Adder- Subtractor, Decimal adder, Decoders, Encoders, Multiplexers.
Synchronous Sequential Logic: Introduction, Sequential Circuits, Storage Elements: Latches, Storage Elements: Flip- Flops.
Registers and Counters: Registers, Shift Registers, Ripple Counters, Synchronous Counters.
Textbook 1: Chapter-4 (4.1, 4.2, 4.4, 4.5, 4.6, 4.9, 4.10, 4.11) Chapter-5 (5.1 - 5.4), Chapter-6 (6.1 - 6.4)

Module-3 **9 Hours**

Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus structure, Performance.
Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Addressing Modes.

Textbook 2: Chapter-1 (1.2 - 1.4, 1.6(1.6.1 - 1.6.2, 1.6.4, 1.6.7)), Chapter-2 (2.2 - 2.5)

Module-4	9 Hours
Input/output Organization: Accessing I/O Devices, Interrupts, Direct Memory Access. **The Memory System:** Semiconductor RAM memories, Read Only Memories, Cache Memories, Virtual Memories, Memory Management Requirements. **Textbook 2 : Chapter-4 (4.1, 4.2.(4.2.1 - 4.2.3), 4.4), Chapter-5 (5.2 5.3,5.5(5.5.1), 5.7, 5.8)**	
Module-5	**9 Hours**
Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction. **Pipelining:** Basic Concepts, Data Hazards. **Textbook 2: Chapter 7 (7.1 - 7.2), Chapter 8 (8.1 - 8.2)**	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Apply Optimize digital circuits utilizing Boolean theorems, Karnaugh maps (K-maps)and universal gate implementations (NAND/NOR).
CO2	Design Construct functional combinational and sequential logic components to fulfill specific arithmetic, data transfer and data storage requirements.
CO3	Analyze Deconstruct computer architectures by mapping operationa functional units, identifying addressing modes and parsing instruction sequencing.
CO4	Evaluate Implement optimized I/O management and memory architectures through structured interrupts, Direct Memory Access (DMA) controllers and multi-level cache memory mapping.
CO5	Demonstrate Apply practical engineering knowledge to trace instruction executions within a basic processor and operate physical and virtual memory configurations.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE- IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 30 marks.
Continuous and Comprehensive Evaluation (B)	20	-	Any two Assessment methods as per 24RNBE4.2 (*if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	**50**	**20**	**CIE Evaluation is done** **For a maximum of 50 marks**

SEE (C)	**50**	**18**	**SEE is a theory exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.**
CIE+SEE (A+B+C)	**100**	**40**	**Final marks shall be the sum of marks scored in Components A, B and C**
The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE. with minimum 40% scored in every component of CIE.			

Textbooks:

1. M. Morris Mano and Michael D. Ciletti, "Digital Design with an Introduction to Verilog Design", 5th Edition, Pearson Education, 2013.
2. Carl V. Hamacher, Zvonko G. Vranesic, and Safwat G. Zaky, "Computer Organization",5th Edition, Tata McGraw-Hill, 2002.

Reference Books:

1. Stephen Brown and Zvonko Vranesic, Fundamentals of Digital Logic with Verilog Design, 3rd Edition, McGraw-Hill Education, 2013.
2. William Stallings, Computer Organization and Architecture: Designing for Performance, 11th Edition, Pearson Education, 2022.
3. John L. Hennessy, David A. Patterson and Christos Kozyrakis, Computer Architecture: A Quantitative Approach, 7th Edition, Morgan Kaufmann, 2025.

Web Links:

1. **https://onlinecourses.nptel.ac.in/**
2. **https://www.coursera.org/**
3. **https://www.coursera.org/**
4. https://matlabacademy.mathworks.com/?page=1 &sort=featured
5. **https://in.mathworks.com/products/simulink.html**

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

- **Skill-based certifications** through NPTEL, NVIDIA Deep Learning Institute, and Coursera.
- **Case studies** on real-world AI systems used in industry.

Self-Study Topics:

1. Analyze Hardware Accelerators and Parallel Architectures.
2. Compare Modern Memory Hierarchies and Performance Bottlenecks.
3. Study Edge Computing and Embedded Hardware Deployment.

Experimental Learning	
1	Realization and Minimization of Combinational Logic: MATLAB Execution: Use the Simulink Library ,Logic and Bit Operations. Drag and drop fundamental logic gates (AND, OR, NOT, XOR) into a canvas.
2	Design of Arithmetic Circuits (Adder/Subtractor): MATLAB Execution: Build a 1-bit Full Adder subsystem in Simulink out of standard gates. Cascade four of these subsystems to create a structural 4-bit Binary Adder-Subtractor.
3	Data Routing Blocks (Multiplexers and Decoders): MATLAB Execution: Build an 8x1 Multiplexer subsystem using native conditional blocks or discrete logic gates. Alternatively, utilize MATLAB's Truth Table block inside Simulink.
4	Sequential Logic & Counter Design: MATLAB Execution: Open Simulink Stateflow or use the standard memory/flip-flop delay blocks (D Flip-Flop or Memory block).
5	Memory Mapping & Cache Performance Analysis: MATLAB Execution: Write a native MATLAB App / Script (.m file) that simulates CPU interactions with Cache memory.
6	Basic Processing Unit Instruction Execution Simulation: MATLAB Execution: Build a mini-datapath model inside Simulink using Simulink Buses and Registers (Unit Delay blocks) acting as the accumulator, program counter (PC) and ALU.

ESTD : 2001
An Institute with a Difference

B.E (Computer Science and Engineering)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

Course Title:	**DATA STRUCTURES AND APPLICATIONS**		
Course Code:	**25BCSTH304**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:0:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**3**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Explain the fundamental concepts, classifications, and operations of data structures.
2. Implement arrays, pointers, and strings for efficient data representation and manipulation.
3. Apply linear data structures such as stacks, queues, and linked lists to solve problems.
4. Construct and traverse non-linear data structures including trees and graphs.
5. Utilize advanced data structures such as hashing and balanced trees to design optimized solutions.

Module-1 **9 Hours**

Introduction to Data Structures: Data structures, classifications (primitive and non-primitive), data structure operations, review of pointers and dynamic memory allocation.
Arrays and Structures: Arrays, sorting using arrays (insertion sort, radix sort), dynam-ically allocated arrays, structures and unions, polynomials, sparse matrices, representation of multidimensional arrays, strings.

Textbook 1: Chapter-2 (2.4–2.5)
Textbook 2: Chapter-2 (2.1–2.3), Chapter-3 (3.1–3.13), Chapter-4 (4.1–4.4), Chapter-5 (5.1–5.8), Chapter-14 (14.8–14.12)

Module-2 **9 Hours**

Stacks: Stacks, stacks using dynamic arrays, evaluation and conversion of expressions, recursion.
Queues: Queues, circular queues using dynamic arrays, priority queue, multiple stacks and queues.
Textbook 1: Chapter-3 (3.1–3.7)

Module-3 **9 Hours**

Linked Lists: Singly linked lists and chains, representing chains in C, linked stacks and queues, polynomials, additional list operations, sparse matrices, doubly linked lists.
Textbook 1: Chapter-4 (4.1–4.5, 4.7–4.8)

Module-4 **9 Hours**

Trees: Introduction, binary trees, binary tree traversals, threaded binary trees, binary search trees, selection trees, forests.
Textbook 1: Chapter-5 (5.1–5.3, 5.5, 5.7–5.9)

Module-5 **9 Hours**

Multiway Search Trees: B-trees, B+ trees.
Graphs: Graph abstract data types, elementary graph operations.
Hashing: Introduction, static hashing, dynamic hashing.
Textbook 1: Chapter-6 (6.1–6.2), Chapter-8 (8.1–8.3),Chapter-11 (11.2, 11.3)

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Implement fundamental data structures and perform operations using arrays, pointers, and strings.
CO2	Apply stacks and queues to evaluate expressions and solve linear data structure problems.
CO3	Develop memory-efficient programs using singly and doubly linked lists.
CO4	Construct and analyze tree-based structures to model hierarchical relation-ships.
CO5	Design solutions using graphs, balanced trees, and hashing techniques to ad-dress real-world computational problems.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of two Internal Assessment Tests shall be scaled down to 30 marks.
Continuous and Com-prehensive Evaluation (B)	20	-	Any two assessment methods as per 24RNBE4.2 (if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	50	20	CIE evaluation is done for a maximum of 50 marks.
SEE (C)	50	18	SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in components A, B, and C.
The minimum marks to be secured in CIE are 20 (40%) out of allotted 50 marks to appear for SEE, with a minimum of 40% scored in every component of CIE.			

Textbooks:

1. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2014.
2. Reema Thareja, Data Structures Using C, 2nd Edition, Oxford Press, 2014.

Reference Books:

1. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Edition, Mc-Graw Hill, 2014.
2. Gilberg and Forouzan, Data Structures: A Pseudo-code Approach with C, 2nd Edition, Cengage Learning, 2014.

3 Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Edition, McGraw Hill, 2013.

4 A. M. Tanenbaum, Data Structures Using C, PHI, 1989.

Web Links:

1. https://youtu.be/S47aSEqm_0I?si=34alu47wCWGdsxwN
2. https://ds2-iiith.vlabs.ac.in/exp/selection-sort/index.html
3. https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/index.html
4. https://ds1-iiith.vlabs.ac.in/exp/linked-list/basics/overview.html
5. https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html
6. https://ds1-iiith.vlabs.ac.in/exp/tree-traversal/index.html
7. https://ds1-iiith.vlabs.ac.in/exp/tree-traversal/depth-first-traversal/dft-practice.html
8. https://onlinecourses-archive.nptel.ac.in/noc18_cs25/preview

Self-Study Topics:

1. Recursion
2. AVL Trees
3. Red Black Trees.

ESTD : 2001
An Institute with a Difference

B.E (CSE and Allied Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

Course Title:	**OPERATING SYSTEMS**		
Course Code:	**25BCSTH305**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:0:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**3**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand the fundamental concepts and architecture of operating systems, including system structure, services, and system calls.
2. Analyse and applyconcepts of process, thread, and CPU scheduling, along with interprocess communication and synchronization techniques.
3. Explain and evaluate memory management strategies, including paging, segmentation, and virtual memory techniques.
4. Demonstrate knowledge of file systems, storage structures, and protection mechanisms, and use relevant system-level commands and APIs for managing OS resources

Module-1 **9 Hours**

Introduction to operating systems, System structures: What operating systems do, Computer System organization, Computer System architecture, Operating System structure, Operating System operations.
Operating System Services: User - Operating System interface, System calls, Types of system calls, System programs, Operating system design and implementation, Operating System structure.
Textbook 1: Chapter-1 (1.1-1.5), Chapter-2 (2.2-2.7)

Module-2 **9 Hours**

Process Management: Process concept, Process scheduling, Operations on processes, Inter process communication.
Multi-threaded Programming: Overview, Multithreading models, Thread Libraries, Threading issues.
Process Scheduling: Basic concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-processor scheduling.
Textbook 1: Chapter-3(3.1-3.4), Chapter-4(4.1-4.4), Chapter-5(5.1 -5.3,5.5)

Module-3 **9 Hours**

Process Synchronization: Synchronization: The critical section problem, solution, Synchronization hardware, Semaphores, Classical problems of synchronization.
Deadlocks: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and recovery from deadlock
Textbook 1: Chapter-6 (6.1-6.6), Chapter-7 (7.1 -7.7)

Module-4 **9 Hours**

Memory Management: Memory management strategies: Background, Swapping, Contiguous memory allocation, Paging, Structure of page table, Segmentation.
Virtual Memory Management: Background, Demand paging, Copy-on-write, Page replacement,

Allocation of frames, Thrashing. **Textbook 1: Chapter-8 (8.1-8.6), Chapter-9 (9.1-9.6)**
Module-5 **9 Hours**
File System, Implementation of File System: File system: File concept, Access methods, Directory and Disk structure. **Implementing File system:** File system structure, File system implementation, Directory implementation, Allocation methods, Free space management. **Secondary Storage Structure:** Mass storage structures, Disk structure, Disk Attachment, Disk scheduling. **Protection:** Goals of protection, Principles of protection, Domain of protection, Access matrix. **Textbook 1: Chapter-10 (10.1-10.3), Chapter-11 (11.1-11.5), Chapter-12 (12.1-12.4), Chapter-14 (14.1-14.4)**

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Interpret the fundamental concepts, architecture, services, and design principles of operating systems.
CO2	Analyze process management techniques and resource allocation strategies for efficient system operation.
CO3	Apply synchronization and coordination mechanisms to manage concurrent processes and system resources effectively.
CO4	Analyze memory management strategies to optimize resource utilization and system performance.
CO5	Examine storage management, file systems, and security mechanisms in operating systems.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 30 marks.
Alternate Assessment (B)	20	-	Any two Assessment methods as per 24RNBE4.2 (*if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	**50**	**20**	**CIE Evaluation is done for a maximum of 50 marks**

SEE (C)	**50**	**18**	**SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.**
CIE + SEE (A+B+C)	**100**	**40**	**Final marks shall be the sum of marks scored in Components A, B and C.**
• The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, " Operating System Concepts", 8th Edition, Wiley-India, 2015.

Reference Books:

1. P.C.P. Bhatt, "An Introduction to Operating Systems: Concepts and Practice", 4th Edition, PHI(EEE), 2014.
2. William Stallings, "Operating Systems: Internals and Design Principles", 6th Edition, Pearson.
3. A.S. Tanenbaum, H. Bos ,2014, "Modern Operating Systems", 4th Edition, Prentice-Hall of India, New Delhi.

Web links:

1. **https://pages.cs.wisc.edu/~remzi/OSTEP/**
2. **https://www.geeksforgeeks.org/operating-systems/**
3. **https://www.tutorialspoint.com/operating_system/index.htm**
4. **https://www.scaler.com/topics/operating-system/**
5. **https://www.youtube.com/watch?v=yKl1uBHPdp30**
6. **https://www.youtube.com/watch?app=desktop&t=1s&v=dOiA2nNJpc0**

Self-Study Topics:

1. Virtual Machines: Study the concept of virtualization, including the architecture and functionality of virtual machines (VMs), and understand how VMs help in efficient resource utilization and system isolation.
2. Basic Linux Commands: Gain hands-on experience with essential Linux terminal commands used for file management, user permissions, process control, and system monitoring.
3. Multithreading Demonstration using Programming Languages: Implement and compare multithreading concepts using:
   a. Java using the Thread class and Runnable interface.
   b. Python using the threading module.
4. C Programming using POSIX Threads (pthreads): Implement multithreading using POSIX threads (pthreads).
5. Overview of Different Operating Systems: Compare the architecture, scheduling, memory management, and file systems of Linux, Windows, Unix, Android, macOS, iOS, RTOS, Chrome OS, Solaris, Haiku OS, and TinyOS.

B.E. (Computer Science and Engineering) Outcome Based Education (OBE) and Choice Based Credit System(CBCS) SEMESTER – III			
Course Title:	**DATA STRUCTURES LABORATORY**		
Course Code:	**25BCSLB306**	**CIE MARKS**	**50**
(L:T:P) + SL	**(0:0:2)**	**SEE MARKS**	**50**
Credits:	**1**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand and implement dynamic memory management techniques in C.
2. Apply linear data structures (stacks, queues, linked lists) to solve computational problems.
3. Analyse and utilize non-linear data structures such as trees and graphs for applications.
4. Design practical solutions to real-world problems using suitable data structures.

Program 1-Weekly Planner

A student needs a simple Weekly Planner System to keep track of their weekly schedule. For each day, the planner should store: name of the day (e.g., Monday, Tuesday. . .), date of the day (numeric format) and the activity planned for that day (e.g., class, event, meeting).

Develop a C program to implement this weekly planner using the following functions.

a) create() to initialize the weekly planner.

b) read() to input the details (day name, date, activity) from the user.

c) display() to generate and print the Weekly Activity Report on screen.

Program 2- Pattern Matching

A publishing company is developing a simple tool to edit documents by replacing certain words or phrases. Design a C program that performs find-and-replace operations on text without using built-in string functions.

a) Read a main string (STR), a pattern string (PAT) and a replacement string (REP) from the user.

b) Search the main string for all occurrences of PAT.

c) Replace every occurrence of PAT in STR with REP.

d) If PAT does not exist in STR, display an appropriate message.

Program 3- Parcel Storage System

A courier office needs a simple Parcel Storage System to manage parcels efficiently. Each parcel is assigned an integer ID for identification. Design a C program that supports the following operations:

a) push() – Add a parcel ID onto the stack when a new parcel arrives.

b) pop() – Remove the last parcel ID from the stack when a parcel is taken out.

c) display() – Show all parcel IDs currently stored in the stack.

d) Display suitable messages when the stack is full (overflow) or empty (underflow).

Program 4 - Calculator Application

In modern calculators, mathematical expressions are usually entered in infix notation, where operators are written between operands (e.g., 3 + 4 * 2). However, calculators and computer

systems evaluate these expressions more efficiently in postfix notation, where operators follow operands (e.g., 3 4 2 * +).

To simulate this process, develop a C program that converts any valid infix expression into its equivalent postfix expression. The program should support alphanumeric operands and the following operators: -, +, *, /, %, ^.

Program 5 - Call Center Help Desk

A call center assigns a ticket number to every incoming call. Since only a limited number of calls can be handled at a time, the calls are placed in a queue. When a representative becomes free, the next call in the queue is served.

Design and implement a C program to simulate this system using a circular queue. The program should support the following operations:

a) Add a new call to the queue (enqueue).
b) Serve the next call (dequeue).
c) Display all waiting calls.

The program must also handle queue overflow and underflow conditions properly.

Program 6 - Student Information Management System

A university wants to maintain a dynamic record of its students for academic and administrative purposes. Each student record contains the following details: USN, Name, Programme, Semester, and Phone Number. Since the number of students may vary frequently due to admissions and withdrawals, a singly linked list (SLL) is chosen for efficient memory usage and flexible data management.

To simulate this system, develop a menu-driven C program that performs the following operations on the student database:

a) Create a Singly Linked List of N student records using front insertion.
b) Display all student records and count the number of nodes in the list.
c) Insert or delete a student record at the end of the list.
d) Insert or delete a student record at the front of the list (demonstrating stack operations).

Program 7 - Employee Record Management System

A company maintains details of its employees for administrative and payroll purposes. Each employee record contains the following fields: SSN, Name, Department, Designation, Salary, and Phone Number. Since employees may frequently join, leave, or change positions, a Doubly Linked List (DLL) provides a flexible way to manage these records efficiently in both directions.

To simulate this process, design and implement a menu-driven C program that performs the following operations on the employee database:

a) Create a Doubly Linked List of N employee records using end insertion.
b) Display all employee records and count the total number of nodes.
c) Insert or delete an employee record at the end of the list.
d) Insert or delete an employee record at the front of the list.

Demonstrate how the DLL can be used as a Double Ended Queue (Deque).

Program 8 - Library Book Management System
Design and implement a menu-driven C program to manage a Library Book Database using a Binary Search Tree (BST). Each record should store the following details: ISBN Number and Book Title. The program should support the following operations: a) Insert a new book into the BST. b) Traverse the BST using Inorder, Preorder, and Postorder. c) Search for a book by its ISBN number and print the title.
Program 9 - City Connectivity using Graph Traversal
In a city transportation system, each city can be represented as a node and the direct roads between them as edges of a graph. To analyse connectivity, it is important to know which cities can be reached from a given starting city. Develop a C program to: a) Represent N cities using an Adjacency Matrix. b) Display all cities reachable from a specified city using DFS or BFS traversal.
Program 10 - Collision Handling in Vehicle Registration System
In a vehicle registration system, each vehicle is assigned a unique 4-digit registration number. To support quick storage and retrieval, these records are organized in a hash table with m storage locations. Develop a C program to: a) Store vehicle registration numbers in the hash table using linear probing and quadratic probing for collision handling. b) Display the final state of the hash table after all insertions.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to

CO1	Demonstrate the working of basic operations on fundamental data structures.
CO2	Develop programs that use linear and non-linear data structures for problem solving.
CO3	Apply dynamic allocation techniques for efficient resource utilization.
CO4	Design practical solutions to real-world problems using appropriate data structures.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE Practical (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.

CIE Practical Test(B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE Practical (A+B)	50	20	Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.
SEE (C)	50	20	SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in Components A, B and C.
The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

ESTD : 2001
An Institute with a Difference

B.E (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title	**PROJECT MANAGEMENT WITH GIT**		
Course Code	**25BCYAE307A**	**CIE MARKS**	**50**
L: T : P	**0:0:2**	**SEE MARKS**	**50**
Credits	**01**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. To familiarise with basic commands of Git.
2. To create and manage branches.
3. To understand how to collaborate and work with Remote Repositories.
4. To familiarise with various controlling commands.

SL No.	Experiments
1	**Setting Up and Basic Commands** Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.
2	**Creating and Managing Branches** Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."
3	**Creating and Managing Branches** Write the commands to stash your changes, switch branches, and then apply the stashed changes.
4	**Collaboration and Remote Repositories** Clone a remote Git repository to your local machine.
5	**Collaboration and Remote Repositories** Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.
6	**Collaboration and Remote Repositories** Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.
7	**Git Tags and Releases** Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.
8	**Advanced GIT Operations** Write the command to cherry-pick a range of commits from "source-branch" to the current branch.
9	**Analysing and Changing Git History** Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?
10	**Analysing and Changing Git History** Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."

11	**Analysing and Changing Git History** Write the command to display the last five commits in the repository's history.
12	**Analysing and Changing Git History** Write the command to undo the changes introduced by the commit with the ID "abc123.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Use the basics commands related to git repository, Create and manage the branches.
CO2	Apply commands related to Collaboration and Remote Repositories.
CO3	Use the commands related to Git Tags, Releases and advanced git operations.
CO4	Analyse and change the git history.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE Practical (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
CIE Practical Test (B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE Practical (A+B)	**50**	**20**	**Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.**
SEE (C)	**50**	**20**	**SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.**
CIE + SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components **A, B and C.**
• The minimum marks to be secured in CIE are 20 (40%) out of allotted 50 marks to appear for SEE.			

Suggested Learning Resources:

- Version Control with Git, 3rd Edition, by Prem Kumar Ponuthorai, Jon Loeliger Released October 2022, Publisher(s): O'Reilly Media, Inc.

- Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, https://git-scm.com/book/en/v2
- https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013094443347369984278 2_shared/overview
- https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013301347121774592119 26_shared/overview

ESTD : 2001
An Institute with a Difference

B.E (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title	**DEVOPS-1**		
Course Code	**25BCYAE307B**	**CIE MARKS**	**50**
L: T : P	**0:0:2**	**SEE MARKS**	**50**
Credits	**01**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand DevOps principles and culture for enhanced collaboration and continuous delivery.
2. Applyversion control using Git and GitHub for effective source code management.
3. Implement CI pipelines using GitHub Actions and Jenkins for automated builds and testing.
4. Gain foundational knowledge of Ansible for configuration management and automation.
5. Use Docker to containerize applications for consistent deployment across environments.

SL No.	Experiments
1.	**Setup a Git repository with branching strategy** Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message. Create a new branch called feature. Switch to the feature branch and make changes to the file. Commit the changes and merge the feature branch into the main branch. Resolve any conflicts if they occur. **Textbook: Chapter-4 : Everything is Code pg. 42-47** E-Resources: • Git Official Documentation: Getting Started • https://www.atlassian.com/git/tutorials/using-branches • https://rogerdudler.github.io/git-guide/
2.	**Setup Jenkins server and create a basic build pipeline** **(a) *Installation and Setup*** Download and install Jenkins on your local machine or a virtual machine. Start Jenkins and complete the initial setup process by installing the recommended plugins. **(b) *Creating and Running a Build*** Create a new Freestyle project in Jenkins. Configure it to pull code from a Git repository and add a build step to compile a simple program. Run the build manually and observe the build output. **Textbook: Chapter-5: Building the Code pg. 65-74**

	E-Resources: • https://www.jenkins.io/doc/book/installing/ • https://www.jenkins.io/doc/tutorials/build-a-java-app-with-maven/
3.	**Write unit test suite and integrate with Jenkins** Create a simple project with testable logic (e.g., a function to add two numbers). Write unit tests using a framework (e.g., unit test in Python or JUnit in Java). Configure Jenkins to run the tests as part of the build process. Add a post-build action to archive the test results or display test reports on the Jenkins dashboard. **Textbook: Chapter-6: Testing the Code pg. 88-90** E-Resources: • https://docs.python.org/3/library/unittest.html • https://junit.org/junit5/docs/current/user-guide/ • https://plugins.jenkins.io/junit/
4.	**Perform Selenium-based automated GUI testing** **(a) *Selenium Script for Login Testing*** Install Selenium WebDriver and write a script to test a web page's login functionality using appropriate locators to interact with input fields and buttons. Execute the script in a browser and verify if the login test passes. **(b) *Jenkins Integration*** Integrate the Selenium test script into a Jenkins job to run automatically after each code commit or at a scheduled interval and observe the test results through the Jenkins interface. **Textbook: Chapter-6: Testing the Code pages 98-100** • https://www.selenium.dev/documentation/webdriver/ • https://selenium-python.readthedocs.io/ • https://www.guru99.com/jenkins-selenium.html
5.	**Containerize a simple application using Docker** Create a simple web application (static HTML or a Python Flask app). Write a Dockerfile that sets up the necessary environment and dependencies. Build the Docker image and verify it using docker images. Run the image using docker run and access the application via browser. Optionally push the image to DockerHub. **Textbook: Chapter-7: Deploying the Code page 132** • https://docs.docker.com/get-started/ • https://docs.docker.com/develop/develop-images/dockerfile_best- practices/ • https://docs.docker.com/docker-hub/
6.	**Deploy a web app using Ansible playbooks** **(a) *Installation and Playbook Setup*** Install Ansible and create a host inventoryfile. Write a playbook to install a web server (e.g., Apache or Nginx) and deploy a sample HTML page to the appropriate directory.

	(b) ***Execution and Verification*** Run the playbook using the ansible-playbook command. Verify the result by opening a web browser and checking if the deployed HTML page is being served by the web server. **Textbook: Chapter-7: Deploying the Code page 122** • https://docs.ansible.com/ansible/latest/user_guide/index.html • https://www.redhat.com/sysadmin/ansible-playbooks • https://www.udemy.com/course/learn-ansible/
7.	**Monitor system resources using Nagios/Telegraf & InfluxDB** **(a)** ***Installation and Configuration*** Install Telegraf and InfluxDB on the monitoring machine. Configure Telegraf to collect system metrics such as CPU usage, memory usage, and disk I/O, and ensure it's set to send data to InfluxDB. **(b)** ***Execution and Data Visualization*** Start the services and verify that Telegraf is pushing metrics into InfluxDB. View the stored time-series data using the InfsluxDB CLI or its web UI. **Textbook: Chapter-8: Monitoring the Code page 391-407** • https://docs.influxdata.com/telegraf/latest/ • https://docs.influxdata.com/influxdb/latest/ • https://medium.com/@mmin18/monitor-linux-with-telegraf-influxdb-grafana-7c9e29db1a9d
8.	**Configure Graphite dashboards for app-level metrics** **(a) Installation and Data Simulation** Install Graphite on your system. Write a script using Python or net cat to simulate sending custom metrics to Graphite's Carbon service at regular intervals. **(b) Dashboard and Visualization** Access the Graphite web interface to monitor incoming data. Create a custom dashboard to visualize application-level or system-level metrics in real-time. **Textbook: Chapter-8: Monitoring the Code page 152** • https://graphiteapp.org/. • https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-graphite-on-ubuntu-18-04 • https://www.metricfire.com/blog/monitoring-custom-metrics-with- graphite/
9.	**Use GitHub Issues or JIRA to manage Agile workflows** Create a GitHub repository. Enable GitHub Projects and Issues. Create at least three issues related to a sample software project. Assign labels, assignees, and link commits or pull requests to issues. Create a project board using Kanban to organize tasks in To Do, In Progress, and Done. **Textbook: Chapter-9: Issue Tracking pages 161-168** • https://docs.github.com/en/issues

	• https://docs.github.com/en/issues/organizing-your-work-with-project- boards • https://www.atlassian.com/software/jira/guides
10.	**Create a CI/CD pipeline with Jenkins for containerized deployment** Create a Jenkinsfile to define a pipeline with the stages: Checkout, Build, Test, Dockerize, and Deploy. Configure the pipeline job in Jenkins to use this script. The pipeline should pull source code from a Git repository, run unit tests, build a Docker image, and deploy the image using Ansible to a target machine. Verify each stage of the pipeline execution. **Textbook: Chapters-5, Chapter-6, Chapter-7: Build → Test → Deploy** • https://www.jenkins.io/doc/pipeline/tour/getting-started/ • https://www.jenkins.io/doc/tutorials/build-a-docker-image/ • https://www.redhat.com/sysadmin/jenkins-ansible-ci-cd-pipeline
Capstone Project Batch formation with 2 – 3 students in a batch and topic finalization. Automate Git → Build → Test → Deploy → Monitor flow for a containerized app with feedback loop using Jenkins, Docker, Ansible, and Graphite.	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Understand DevOps principles and culture to promote collaboration and accelerate software delivery.
CO2	Apply version control techniques using Git and GitHub for efficient source code management and team collaboration.
CO3	Design and implement Continuous Integration (CI) pipelines using GitHub Actions and Jenkins for automated building and testing.
CO4	Acquire foundational knowledge of Ansible for infrastructure automation and configuration management.
CO5	Utilize Docker to containerize applications, ensuring consistent deployment across development, testing, and production environments.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE Practical (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
CIE Practical Test (B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.

Total CIE Practical (A+B)	**50**	**20**	**Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.**
SEE (C)	**50**	**20**	**SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.**
CIE + SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components **A, B and C.**
• The minimum marks to be secured in CIE are 20 (40%) out of allotted 50 marks to appear for SEE.			

Textbook:

1. Learning DevOps – Continuously Deliver Better Software by Mikael Krief

Weblinks:

1. https://github.com/rootusercop/Free-DevOps-Books-1/blob/master/book/Learning%20DevOps%20-%20Continuously%20Deliver%20Better%20Software.pdf

E- Learning Resources:

1. **CICD – DevOps Automation and DevSecOps:**
   https://elearn.nptel.ac.in/shop/iit-workshops/completed/cicd-devops-automation-and-devsecops-automation/
2. DevOps Tutorial by 3RI Technologies:
   https://www.youtube.com/playlist?list=PLAkvN1xnxGzGwWfep3tvUzuX11pBEGOex

Practical Based learning Mini Project:

1. Project Based Learning.

B.E. (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title:	**COMMUNITY PROJECT (PROJECT-BASED LEARNING) / SOCIETAL PROJECT**		
Course Code:	**25BCBPR308**	**CIE MARKS**	**50**
CI (L /T): LI(P):TW&SL:TH/S	**0:0:0:30**	**SEE MARKS**	**50**
Credits:	**01**	**EXAM HOURS**	**03**

COURSE OBJECTIVE:

The Community Project / Societal Project is a practical, project-based learning course that builds students' awareness on Educational Social Responsibility (ESR). It focuses on real issues such as the adaptation of technology for rural and urban development, the application of tools for enhanced governance in civic societies, and sustainable technological solutions for better living. Students may work individually or in groups of up to four, including interdisciplinary topics, and assessment is based on the prescribed rubrics.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Identify and analyze societal problems through field interaction and technical assessment.
CO2	Develop skills in project planning, field survey, data collection, implementation, documentation, and reporting.
CO3	Apply principles of social responsibility in engineering practice through community-oriented interventions
CO4	Exhibit multidisciplinary learning, problem-solving ability, and communication skills based on collected data (requirements), stakeholder feedback, and baseline conditions.
CO5	Choose feasible societal problem statements and solutions considering practical constraints and sustainability aspects.
CO6	Create a product by collaborating with stakeholders and within the team to deliver a product/project to the intended community/organization.

SET OF PROJECT ACTIVITIES (TW/SL = 30 HOURS)
Methods for project identification: Students may identify the project topic using one or more of the following methods (not limited to): Interaction with local rural or urban stakeholders such as local leaders, corporation/municipality/Panchayath officers, NGOs, etc.; review of issues highlighted in print and electronic media and social networks. After selecting a broad topic, collect data and requirements, conduct a feasibility survey, estimate timelines and resources, approximate expenditure (if applicable), and define the problem statement with objectives clearly.

List of Project Topics/Titles (Not Limited):				
Sl. No.	**Broad Areas/Project titles**	**Measurable engineering outcome**	**Specific POs**	**SDGs**
1	Computer-Based Learning Tool for Mentally Challenged Students	Design and develop a web/browser-based learning tool as per the requirements of the special school.	5, 6	4
2	Medication management system for an old-age home.	Design and develop a web-based application to monitor the administration of medicine to senior citizens residing at an old-age home.	6	3
3	Virtual Labs for Learning Science	Design and Development simulation environment to do the experimentation list in Physics/Chemistry/Biology for high school in a rural area	5, 6	4
4	Local Skill Development and Employment Information Portal	Design and develop a web-based portal that provides information on local skill development programs, vocational training, internships, and employment opportunities for youth in the community.	6	4, 8
5	Digital Complaint Portal for Gram Panchayat	Design and develop a web-based complaint portal (Specific nature) for Gram Panchayat	6	16
6	Plant Care Reminder Application	Implement a reminder mechanism that generates timely notifications for watering, fertilizing, pruning, and other plant-care activities.	6	13,15
7	Basic Crop Recommendation System	Develop a basic crop recommendation system that recommends suitable crops based on user-provided parameters such as soil type, season, rainfall, and water availability	6	2, 13, 15
8	Lost and Found Application	Develop a Lost and Found Application that enables users to report, search, and manage lost and found items	6	11, 16

9	Campus Navigation System	Develop a campus navigation system that enables users to locate important campus facilities and obtain directions between selected locations, which provides a simple digital map and location information for important facilities on campus.	6	15
10	Public Waste Collection Monitoring System	Design and develop a web/mobile application that enables citizens to report uncollected garbage with geo-tagged photos, track complaint status, and assist local municipal authorities in monitoring waste collection and sanitation services.	6	3, 11, 16
11	Volunteer Management System for NGOs	Develop a web-based system to manage volunteer registration, event scheduling, attendance, and activity reporting for NGOs.	6	16
12	Local Farmer Market Information System	Design and develop a digital platform that connects farmers directly with consumers by providing crop availability, prices, and contact information.	6	2, 8, 12
13	Government scheme awareness for betterment of livelihood	Design a UI in the native language for people in rural areas to make best utilization of the government scheme.	6	11
14	Community Renewable Energy Awareness and Management Portal	Design and develop a web/mobile portal to provide information on renewable energy resources and estimate energy savings within the community.	6	7, 11, 13
15	Digital Learning Platform for Rural Government Schools	Design and develop a digital learning platform (Tool) that provides interactive learning content and activity-based learning materials in regional languages for rural and small village govt schools (primary govt. schools often lack interactive learning tools)	5, 6	4

16	Digital Heritage Preservation and Cultural Information Portal	Develop a digital platform to document, preserve, and promote local heritage sites, cultural events, and historical information	6	11, 16
17	Community Disaster Preparedness and Emergency Information System	Design and develop a web/mobile application for disaster preparedness and emergency alerts during natural disasters.	6	11, 13
18	Pesticide Usage Advisory and Organic Farming Awareness System	Design and develop an application that provides recommendations on appropriate pesticide usage, promotes organic farming practices, and suggests suitable organic alternatives based on soil test reports and crop information.	6	2, 3, 15
19	Cyber Security Awareness Platform for Phishing, Digital Arrest Scams, and Malicious Applications	Create a simple web app where users paste suspected text/links to get a risk score based on known phishing patterns, deceptive URLs, and high-risk keywords	5, 6	16
20	Local Public Transportation Information and Route Assistance System	Design and develop a web or mobile application that provides location-based information on available public transportation services, route maps, estimated travel options, and nearby transport facilities to improve accessibility and promote the use of public transportation.	6	11, 15
21	Digital Inventory and Sales Management System	Design and develop a web-based inventory, billing, and sales analytics system for small businesses with dashboard reports and low-stock alerts.	6	8, 9
22	Smart Financial Literacy and Budget Planner for public	Develop a mobile/web application that helps users manage household budgets, savings, government schemes, and financial planning through support.	6	1, 8
23	Local Business Promotion and Market Analytics Portal	Design a digital platform that enables local vendors and small businesses to advertise products, analyze customer trends, and improve digital visibility.	6	8, 9

24	Blood Donor and Emergency Blood Availability Portal	Develop a web/mobile platform to connect blood donors, hospitals, and patients with real-time blood availability information.	6	3
25	Women Safety and Emergency Alert Application	Develop a mobile application with SOS alerts, live location sharing, emergency contacts, and nearby police station information.	6	5, 11, 16
26	Public Grievance and Civic Issue Reporting System	Develop a web/mobile application that allows citizens to report civic issues such as potholes, streetlight failures, and drainage problems and monitor complaint resolution status.	6	11,15, 16
27	Digital Public Library and Book Exchange System	Design and develop a web-based platform that enables community members or schools to manage book donations, borrowing, returns, and book exchange activities.	6	4
28	Community Water Quality Reporting and Awareness Portal	Design and develop a web-based portal for citizens to report drinking water issues, record simple water quality observations, and provide awareness on safe water practices through educational resources.	6	3, 6
29	Community Blood Donor Registry and Emergency Request System	Design and develop a web/mobile application that maintains a local blood donor database, enables emergency blood requests, and notifies eligible donors based on blood group and location.	6	3, 11
30	Smart Community Health Camp Management System	Design and develop a web/mobile application to manage community health camps by facilitating online patient registration & doctor scheduling/digital health record maintenance/medicine inventory tracking.	6	3
31	Food Waste Reduction and Food Donation Platform	Develop a platform connecting restaurants, hotels, and households with NGOs to donate surplus food before it is wasted.	6	2, 12

32	Community Tree Plantation and Green Cover Monitoring Portal	Develop a web/mobile application to record tree plantation activities.	6	13, 15
33	Career Guidance and Information Portal	Develop an app to get information on aptitude tests Course/scholarship alerts/placement information	6	4, 8
34	NGO Donation and Resource Management System	Develop a web application to manage donations, inventory, volunteers, and financial transparency for NGOs.	6	16
35	Smart Local Tourism Business Portal	Develop a web platform promoting local tourism, homestays, guides, and cultural businesses with booking facilities.	6	8, 11, 15
36	Local Artisan and Handicraft Marketplace	Design and develop a platform promoting local artisans and traditional handicraft products.	6	8, 11
37	Waste Recycling Marketplace	Develop a digital platform connecting households, recyclers, and scrap collectors with pickup scheduling.	6	11, 12
38	Digital Accessibility Evaluation and Improvement for Public Websites	Evaluate the accessibility of at least 10 government or public-service websites using WCAG guidelines, develop accessibility improvement recommendations, and implement a prototype demonstrating compliance improvements.	6	16
39	Digital Library and Knowledge Sharing Platform for Government Schools	Develop a multilingual digital library with e-books, educational videos, and quizzes to improve access to learning resources.	6	4
40	Community Renewable Energy Awareness and Management Portal	Design and develop a web/mobile portal to provide information on renewable energy resources and promote the adoption of sustainable energy practices within the community.	6	7, 11, 15

41	Local Business Digital Marketplace	Web portal onboarding of at least 25 local vendors and product catalog	6	8, 12
42	Community Blood Donor and Emergency Volunteer Network	Web/mobile application connecting donors and volunteers	6	3
43	Mental Health Awareness and Helpline Directory App	Develop an application providing verified helpline contacts, self-help resources, and stigma-reducing content.	6	3
44	Diet & Nutrition Awareness Tool for Anganwadi Children	Develop an application to track and suggest balanced meal plans for children under a nutrition scheme.	5, 6	2, 3
45	Digital Literacy Portal for Elderly Citizens	Develop a simplified, accessible web application teaching basic smartphone and internet skills to senior citizens.	6	4, 11
46	Scholarship & Government Scheme Eligibility Checker	Develop a Web tool matching students to eligible scholarships and welfare schemes based on entered criteria.	5, 6	4
47	Digital Record-Keeping System for Self-Help Groups	Develop a Ledger and record-keeping application to help women's SHGs manage savings, loans, and meeting records.	6	5, 8
48	Elderly Companion & Wellness Check-In App	Develop an application enabling family/volunteers to schedule and log wellness check-ins for the elderly living alone.	6	3
49	Old-Age Home Volunteer Coordination Portal	Develop a Platform matching volunteers with old-age homes for scheduled visits. events, and support activities.	6	3, 11
50	Local Job & Skill Matching Portal for Daily-Wage Workers	Develop a Web application connecting unskilled and semi-skilled workers with nearby job opportunities.	6	8

51	Street Vendor Digital Registry and Support Portal	Develop an application for the registry helping street vendors to access municipal registration information and gain customer visibility.	6	8
52	Microloan and Repayment Tracker for Small Vendors	Develop an application helping small vendors track micro-loans, repayments, and daily sales.	6	8
53	Rainwater Harvesting Feasibility & Awareness Tool	Develop a tool estimating rooftop rainwater harvesting potential based on roof area and local rainfall data.	5, 6	6,13
54	Water Quality Reporting System for a Village / Ward	Develop a Web/mobile application for citizens to report water contamination issues and track resolution status.	6	6
55	Community Skill-Building Workshop Registration and Tracker App	Develop an application to register participants and track attendance for community skill-building workshops such as tailoring, computer usage, and first-aid.	6	4, 11
56	Local Event & Festival Information Aggregator App	Develop an application aggregating community event and festival schedules for a town or ward.	6	11
57	Community Sports Ground / Facility Booking App	Develop an application for local youth to check availability and book shared sports grounds or courts.	6	3, 11
58	Talent Scouting Platform for Rural Sports Players	Develop a platform where local sports talent can register profiles and gain visibility to coaches and scouts.	6	3
59	Youth Skill Development and Internship Finder App	Develop an application listing local internships, apprenticeships, and skill-training programs for youth.	6	4, 8
60	Financial Literacy Awareness App for Rural Households	Develop an application teaching basic banking, savings, and fraud-avoidance concepts with local-language support.	6	4, 8

61	Legal Aid & Rights Awareness Portal	Develop a portal listing free legal aid contacts with simplified explanations of common citizen legal rights.	6	16
62	Digital Insurance / Pension Scheme Awareness Tool	Develop a tool helping unorganized-sector workers check eligibility and enroll in government insurance/pension schemes.	5, 6	8
63	Livestock Health Record and Vaccination Tracker	Develop an application to record cattle and livestock health history and vaccination schedules for farmers.	6	15
64	Soil Testing Appointment & Report Access Portal	Develop a portal letting farmers book soil-testing slots and access digital test reports.	6	15
65	Power Outage Reporting and Tracking App for a Locality	Develop an application allowing residents to report and track power outages to the local utility board.	6	7, 11
66	Community Borewell / Water Tanker Scheduling App	Develop an application to schedule and track water tanker deliveries for water-scarce localities.	6	6, 11
67	Cyber Safety and Digital Fraud Awareness App	Develop an application educating students and senior citizens on common online scams and safe digital practices.	6	16
68	Adult Literacy Program Registration and Progress Tracker	Develop an application to register learners and track attendance and progress in a community adult-literacy drive.	6	4
69	Inclusive Learning Content tool for Differently Abled Students	Develop a tool delivering audio/visual learning content adapted for students with visual or hearing impairments.	5, 6	4
70	School Dropout Tracking and Follow-up System	Develop an application for teachers to record attendance trends and flag/follow up on students at risk of dropping out.	6	4

71	Public Park / Garden Maintenance Complaint Tracker	Develop an application for citizens to report maintenance issues in public parks and gardens and track resolution.	6	11,15
72	Digital Suggestion Box for Local Governance	Develop an application allowing citizens to submit suggestions and feedback to local governing bodies on civic matters.	5, 6	16
73	Community Health Screening Camp Scheduler	Develop an application to schedule and manage attendance for community blood-pressure/diabetes screening camps.	6	3
74	Digital Queue Management System for Health Camps	Develop a token-based appointment and queue management system/app for medical camps or PHCs.	6	3
75	Streetlight Fault Reporting System	Develop an application for citizens to report faulty streetlights using photographs and GPS location.	6	11
76	Village Asset Mapping System	Develop a GIS-enabled or map-based application to document community assets such as schools, hospitals, water tanks, and roads.	6	11
77	Food Waste Reduction & Expiry Management System	Design and develop a web/mobile application to record food items, monitor expiry dates, provide expiry reminders, and maintain food inventory to help reduce food wastage.	6	2, 12
78	Community Hall Booking & Management System	Design and develop a web-based platform to manage community hall bookings, track hall availability, process booking requests, schedule maintenance, and maintain booking records.	6	11
79	Public Infrastructure Asset Monitoring System	Design and develop a web-based platform to maintain records of public infrastructure assets, report maintenance issues, assign maintenance requests, and track their status.	6	9, 11

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment structure:

A. Continuous Internal Evaluation (CIE) – 50 marks

- Students, in batches of 3–4, shall undertake at least one project (or a combination of two related activities) aiming, as far as possible, at a logical implemented or implementable outcome.
- CIE will be conducted by a departmental committee (HoD/nominee, guide and one senior faculty) using the following rubrics totalling 50 marks:

CIE pass requirement:

A student shall secure a minimum of 20 marks out of 50 in CIE to become eligible to appear for SEE.

- Students who fail to earn the required CIE marks may be given additional opportunities before closure of the semester; no re-doing of CIE is permitted after the end of the semester.

B. Semester End Examination (SEE) – 100 marks (scaled to 50)

- Examination duration: 2 hours.
- Maximum examination marks: 100 (entered as 100 in the VTU portal; the ITI-SMU module automatically scales 100 → 50 for result processing).
- The SEE shall be conducted batch-wise by a committee of three faculty: project guide, one faculty from the same department and one from a different department.

SEE pass requirement and overall passing rule:

- For a pass in SEE, a student shall secure minimum 35 marks out of 100.
- For declaration of a pass grade, the sum of CIE marks (out of 50) and scaled SEE marks (out of 50) shall be ≥ 40; otherwise, the student is awarded grade F (Fail).

CIE Rubric — Continuous Internal Evaluation (Total 50 marks):

Criterion	Weight (Marks)	Excellent (Full marks)	Good	Fair	Needs Improvement
Problem identification & relevance to society/environment	8	Problem clearly identified with strong evidence of relevance and well-defined scope. (8)	Relevant problem with clear scope and adequate evidence. (6)	Problem defined but justification is weak. (4)	Problem unclear or not relevant. (2)
Formation of problem statement; engineering approach, innovation & feasibility	10	Precise statement; sound, innovative, feasible approach. (10)	Adequate statement; feasible with minor gaps. (8)	Partial statement; marginal feasibility. (5)	Missing or infeasible. (2)

Interaction with stakeholders; data collection, analysis & interpretation	8	Thorough engagemet; comprehen sive data and well-interpreted analysis. (8)	Good engagement; adequate data and analysis. (6)	Limited engagement ; incomplete data and superficial analysis. (4)	Little or no stakeholder contact; no data/analysis. (2)
Project documentation ; clarity, organization & problem-solving demonstrated	10	Complete, well-organized, analytical, and strong problem-solving. (10)	Complete and clear with some analytical content. (8)	Basic with omission; problem-solving limited. (5)	Missing/poorly organized; no clear problem-solving. (2)
Communication & presentation skills	6	Confident, structured, and answers queries fully; visuals effective. (6)	Clear and satisfactory; responds to most queries. (5)	Acceptable but lacks clarity or confidence; weak responses. (3)	Poor or absent; unable to respond. (2)
Contribution & benefit to society (impact, sustainability)	8	Clear, demonstrable societal benefit and realistic sustainability path. (8)	Benefit plausible; sustainability considered with minor gaps. (6)	Limited or unclear benefit; sustainability weak. (4)	No evident benefit or sustainability. (2)

Notes for CIE rubric:

- Minimum CIE pass requirement: 20/50 to be eligible for SEE. [Use departmental committee (HoD/guide + senior faculty) to apply the rubric and record marks.]

SEE Rubric — Semester End Examination (Enter marks out of 100; scaled to 50 later):

Criterion	Weight (marks)	Excellent (Full marks)	Good	Fair	Needs Improvement
Problem identification & relevance to society/enviro nment	16	Clearly articulated with strong evidence and context. (16)	Clear with reasonable evidence. (12)	Problem stated but evidence/cont ext weak. (8)	Poorly defined or irrelevant. (4)
Contribution & benefit to society; impact	16	Clear, evidence-based impact	Reasonable impact discussion and	Limited evaluation of impact or	No coherent assessment of impact or

assessment & sustainability		assessment with scaling recommendations. (16)	sustainability considerations. (12)	sustainability. (8)	sustainability. (4)

SEE pass rule:

- Student must secure at least 35/100 in SEE itself to be eligible for pass consideration.
- Final passing is based on CIE (out of 50) + SEE (scaled to 50) >= 40. Otherwise, grade F (Fail) is awarded.

Suggested implementation notes (for BoS / exam committees):

- Use numerical scores from the rubric and keep brief evaluator comments for each criterion.
- For team projects: record individual contribution within the documentation and adjust individual marks using a peer-evaluation sheet or guide's assessment.
- Maintain evidence (field notes, photos, questionnaires, DPRs, reports) as annexures for SEE moderation.
- Use standardized score sheet (spreadsheet) mapping rubric items to columns for easy aggregation and archival.
- Awareness Campaign on Single-Use Plastic and E-Waste Management

Assessment Structure:

Continuous Internal Evaluation (CIE) shall be conducted by the Departmental Project Review Committee consisting of a Senior Professor, the Project Guide, and one additional faculty member nominated by the Head of the Department. Semester End Evaluation (SEE) shall be conducted by university-appointed examiners. The evaluation shall be based on rubrics designed to assess the attainment of NBA-defined graduate attributes, including problem analysis investigation, design of solutions, communication, teamwork, and professionalism**.**

The weightage of marks for CIE and SEE is 50% each.

Pass Criteria: A candidate shall secure a minimum of 40% of the total marks prescribed for Project Work, subject to the required minimum in Continuous Internal Evaluation and Semester End Evaluation as per VTU regulations.

Suggested Learning Resources:

Reference Books:

1. Phadke, N. D., Community Engagement and Social Innovation, Oxford University Press, New Delhi.
2. Garg, N. R., Project-Based Learning in Engineering Education, Pearson India Education Services, Noida.
3. Gupta, A. and P. Singh, Service Learning and Community Projects, McGraw-Hill Education (India), New Delhi.
4. Wilson, G., Practical Guide to Community Development Projects, Routledge, London.

Weblinks and Video Lectures (e-Resources):

1. VTU e-Learning Centre: http://elearning.vtu.ac.in
2. NPTEL courses on Project-Based Learning and Social Innovation.
3. SWAYAM courses on community engagement and sustainability.
4. UN Sustainable Development Goals portal.

5. MyGov India and local civic body portals for community initiatives.
6. YouTube resources on design thinking, social innovation, and community service learning.
7. NGO and government field-action case studies related to sanitation, water, health, education, and environment.

ESTD : 2001
An Institute with a Difference

B.E. (Common to All Branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title:	**NATIONAL SERVICE SCHEME**		
Course Code:	**25BNSSP309**	**CIE MARKS**	**100**
L : T : P: SL	**(0:0:0)+(30 Hours/Sem)**	**SEE MARKS**	**0**
Credits:	**0**	**EXAM HOURS**	**-**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Identify the needs and problems of the community and participate in problem-solving activities.
2. Develop sense of social & civic responsibility and utilize knowledge to find practical solutions to individual and community problems.
3. Develop the competencies required for group living and sharing responsibilities, and gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.

Module-1
Introduction to NSS, Life Competencies, Youth Leadership and NSS.
Module-2
Social connect and responsibilities.
Module-3
Activities -1 (Choose from Activity list)
Module-4
Activities -2 (Choose from Activity list)
Module-5
Activities -3 (Choose from Activity list)

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Understand the motto and concepts of NSS.
CO2	Develop overall personality of volunteers and make them as leaders and responsible citizens of our nation.
CO3	Analyze the environmental and societal problems/issues and will be able to design solutions for the same.
CO4	Communicate effectively with the community and with the officials.

ACTIVITY LIST	
Plantation Drive: Adopt a Tree	Developing & managing an efficient garbage disposal system.
Food Preservation/ Packaging awareness	Tourism Promotion: Innovative Approaches.
Facilitating 100% digitized money transactions	Contribution to any national-level initiative of the Government of India. E.g. Digital India/ Skill India/ Swachh Bharat Internship etc.,

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	-	Sum of Two Internal Assessment Tests (MCQs) each con-ducted for 30 marks.
Continuous and Compre-hensive Evaluation (B)	40	-	Any two Assessment methods as per 24RNBE4.2, scaled to 40 marks.
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks.
• The minimum marks to be secured in CIE is 40 (40%) out of allotted 100 marks to Pass.			

Reference Books:

1. NSS Course Manual, Published by NSS Cell, VTU Belagavi.
2. Government of Karnataka, NSS cell, activities reports and its manual.
3. Government of India, NSS cell, Activities reports and its manual.

Weblinks:

1. https://nss.gov.in/
2. https://pledge.mygov.in/

ESTD : 2001
An Institute with a Difference

B.E (Computer Science and Engineering)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER– III

Course Title:	**PHYSICAL EDUCATION**		
Course Code:	**25BPEDU309**	**CIE MARKS**	**100**
(L: T: P) + SL	**(0:0:0)+(30 Hours/Sem)**	**SEE MARKS**	**---**
Credits:	**----**	**EXAM HOURS**	**---**

COURSE LEARNING OBJECTIVES:
The purpose of the course is to facilitate the learner to:

1. Understand the fundamental concepts of Physical Education, games and sports skills, Health, Nutrition and Fitness
2. Familiarization of health-related Exercises, Sports for overall growth and development
3. Participate in the competition at regional/state / national / international levels.
4. Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

Module-1: Title: Athletics-Track Events-110 and 400 Mtrs Hurdles **3 Hours**

110 Mtrs and 400Mtrs:

1. Hurdling Technique: Lead leg Technique, Trail leg Technique
2. Crouch start (its variations) use of Starting Block.

Module-2: Title: Athletics –Jump **3 Hours**

High Jump

a) Approach Run
b) Take-off
c) Bar Clearance (Straddle)
d) Landing

Module-3: Title: Athletics – Throws **3 Hours**

Discus Throw

e) Holding the Discus
f) Initial Stance Primary Swing
g) Rotation
h) Release and Recovery (Rotation in the circle)

Module-4: Title: Football **3 Hours**

i) Fundamental skills

a) Kicking: Kicking the ball with inside of the foot, Kicking the ball with Full Instep of the foot, Kicking the ball with Inner Instep of the foot, Kicking the ball with Outer Instep of the foot and Lofted Kick.
b) Trapping: Trapping the Rolling ball and the Bouncing ball with sole of the foot.
c) Dribbling: Dribbling the ball with Instep of the foot, Dribbling the ball with Inner and Outer Instep of the foot.
d) Heading: In standing, running and jumping condition.

e) Throw-in: Standing throw-in and Running throw-in. f) Tackling: Simple Tackling, Slide Tackling. g) Goal Keeping: Collection of Ball, Ball clearance—kicking, throwing and deflecting. h) Game practice with application of Rules and Regulations.
Module-5: Title: Hockey **3 Hours**
i) Fundamental skills **a)** Passing: Short pass, Long pass, Push pass, Hit. b) Trapping. c) Dribbling and Dozing. d) Penalty stroke practice. e) Penalty corner practice. f) Tackling: Simple Tackling, Slide Tackling. g) Goal Keeping: Ball clearance, kicking and deflecting. h) Game practice with application of Rules and Regulations. **iii. Rules and their interpretation and duties of the officials**

COURSE OUTCOMES

After successfully completing the course, the student will be able to:

CO1	Understand and demonstrate fundamental techniques of track events including 110m and 400m hurdles, with proper use of starting blocks and hurdling skills.
CO2	Apply technical knowledge of athletic jumps, particularly high jump, including approach run, take-off, bar clearance and landing techniques.
CO3	Develop skills and perform throwing events, specifically discus throw, with correct techniques such as grip, stance, rotation, release and recovery.
CO4	Demonstrate fundamental skills of football such as kicking, trapping, dribbling, heading, throw-in, tackling and goalkeeping along with understanding rules and officiating.
CO5	CO5 Exhibit basic skills and techniques in hockey, including passing, trapping, dribbling, tackling, penalty strokes/corners and understand rules and duties of officials.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	-----	Sum of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Project (B)	40	-----	-----
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks
• The minimum marks to be secured in CIE is 40 (40%) out of 100 marks allotted to Pass.			

Textbooks:

1. Health and Physical Education by V.K. Sharma
2. Introduction to Physical Education, Fitnessand Sport by Daryl Siedentop
3. Physical Education Activities, Sports and Games: The Origin and Background, Rules of Play, Teaching Hints and Fundamentals by Louis E. Means

Reference Books:

1. Saraswati Health and Physical Education Textbook
2. Foundations of Physical Education, Exercise Science and Sport by Deborah A.Wuest

ESTD : 2001
An Institute with a Difference

B.E. (Common to All Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

Course Title:	**YOGA**		
Course Code:	**25BYOGA309**	**CIE MARKS**	**100**
(L: T: P) + LS	**(0:0:0)+(30Hours/Sem)**	**SEE MARKS**	**---**
Credits:	**---**	**EXAM HOURS**	**---**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Physical, Mental & spiritual Objectives of Yoga
2. Explain the meaning of Yoga.
3. Discuss the importance of the Yoga and its scope.
4. Describe the aims and objectives of Yoga.
5. Clarify the misconceptions on Yoga

Module-1: Title: Suryanamaskara **3 Hours**

Suryanamaskar 12 count 6 rounds

a) Suryanamaskar prayer.
b) Meaning, need & importance.
c) Benefits of Suryanamaskara.

Module-2: Title: Yoga **3 Hours**

a) Meaning and definition.
b) Brief introduction of yogic practices for the common man.
c) Rules and regulations.
d) Misconceptions of Yoga.

Module-3: Title: Asanas - Sitting & Standing **3 Hours**

Need and importance of different Asanas, meaning by name, technique, precautionary measures and benefits of each asana.

Sitting

a) Ardha Ushtrasana.
b) Yogamudra in Padmasana.

Standing

a) Hastapadasana.
b) Utkatasana.

Module-4: Title: Asanas - Supine line **3 Hours**

Need and importance of different Asanas, meaning by name, technique, precautionary measures and benefits of each asana.

Supine line

a) Sarvangasana.
b) Chakraasana.

Module-5: Title: Pranayama **3 Hours**

a) Kapalabhati - Revision of practice 40 strokes/min - 3 rounds.
b) Pranayama:

i. Ujjayi.
ii. Sheetali.

c) Meaning, Need, importance of Pranayama, Different types, Meaning by name, technique, precautionary measures and benefits of each Pranayama.

COURSE OUTCOME

After successfully completing the course, the student will be able to:

CO1	Explain the meaning, definition, scope and objectives of Yoga and its role in physical, mental and spiritual well-being.
CO2	Demonstrate Suryanamaskara (12 counts, 6 rounds) with correct technique and understand its importance and benefits.
CO3	Apply knowledge of yogic practices, rules and regulations while identifying and correcting common misconceptions of Yoga.
CO4	Perform sitting, standing and supine Asanas with proper technique, precau-tions and awareness of their benefits.
CO5	Practice Pranayama techniques (Kapalabhati, Ujjayi, Sheetali) to improve breathing control, concentration and overall health.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	----	Sum of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Quiz (B)	40	----	Quiz for a maximum of 40 marks
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks
• The minimum marks to be secured in CIE is 40 (40%) out of 100 marks allotted to Pass.			

Textbooks

1. The Yoga Sutras of Patanjali – Sri Swami Satchidananda.
2. Light on Yoga — B.K.S Iyengar.

Reference Books

1. The Heart of Yoga — T.K.V Desikachar.
2. "Asana Pranayama Mudra Bandha" by Swami Satyananda Saraswati

B.E. (Common to All Branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title:	**MUSIC**		
Course Code:	**25BMUSK309**	**CIE MARKS**	**100**
L: T: P: S	**0:0:0**	**SEE MARKS**	**---**
Credits:	**---**	**EXAM HOURS**	**---**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand the origin, evolution, and fundamental concepts of Indian classical music.
2. Familiarize themselves with the basic forms of Carnatic and Hindustani music compositions and appreciate their musical structure and significance.
3. Identify and classify the major musical instruments used in Carnatic and Hindustani music and understand their construction and performance characteristics.
4. Recognize the contributions of eminent Indian composers and musicians to the development and preservation of Indian classical music.
5. Develop an appreciation for India's rich musical heritage, fostering creativity, cultural awareness, and lifelong interest in music.

Module-1	**3 Hours**
Origin of the Indian Music: Evolution of the Indian music system, Understanding of Shruthi, Nada, Swara, Laya, Raga Tala, Mela.	
Module-2	**3 Hours**
Compositions: Introduction to the few types of composition in Carnatic Music- Geethe, Swarajathi, Varna, Madhyama Kala Krithi,	
Module-3	**3 Hours**
Compositions: Introduction to the few types of composition in Hindustani Music : Khayal, Drupad, Tarana.	
Module-4	**3 Hours**
Music Instruments: Classification and construction of string instruments, wind instrurnents, percussion instruments in both Carnatic and Hindustani Music.	
Module-5	**3 Hours**
Composers: Biography and contributions of of the following eminent personalities Purandaradasa, Thyagaraja,. Mysore Vasudevacharya, Pt. Mallikarjun Mansur, Vidhushi Gangubhai Hangal.	

COURSE OUTCOME

After successfully completing the course, the student will be able to:

CO1	Explain the origin, evolution, and fundamental concepts of Indian classical music
CO2	Compare and distinguish the characteristics of selected Carnatic and Hindustani music compositions.

CO3	Classify the major categories of Indian musical instruments and describe their construction and applications in Carnatic and Hindustani music.
CO4	Examine the contributions of eminent Composers/Musicians and appreciate their role in preserving and promoting Indian classical music.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	-	Sum of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Project(B)	40	-	Submission of report on a topic related to music.
Total CIE + SEE (A+B)	**100**	**40**	CIE Evaluation is done for a maximum of 100 marks
• The minimum marks to be secured in CIE is 40 (40%) out of 100 marks allotted to Pass.			

Text Books:

1. Vidushi Vasantha Madhavi, “Theory of Music”, Prism Publication, 2007.
2. T Sachidevi and T Sharadha (Thirumalai Sisters), Karnataka Sangeetha Dharpana- Vol. 1 (English), Shreenivaas Prakaashana, 2018.
3. B. Chaitanya Deva, An Introduction to Indian Music, Publications Division, Government of India, New Delhi.

Reference Books:

1. Lakshminarayana Subramaniam, Viji Subramanaim, "Classical Music of India: A Practical Guige, Tranqueber 2018.
2. R. Rangaramanuja Ayyangar, "History of South Indian (Carnatic) Music", Vipanci Charitable Trust; Third edition, 2019,
3. Vamanrao H. Deshpande, Indian Musical Traditions: An Aesthetic Study of the Gharanas in Hindustani Music, Popular Prakashan, Mumbai.

Common to all Programmers Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title	**ADDITIONAL MATHEMATICS-1**		
Course Code	**25BMADP310**	**CIE Marks**	**100**
(L:T:P) + SL)	**(1:0:0)+(14 Hours/Sem)**	**SEE Marks**	**--**
Credits	**0**	**Exam Hours**	**--**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Apply the fundamental concepts of complex trigonometry, vector algebra, and differential and integral calculus to analyze mathematical and engineering problems.
2. Apply the principles of vector differentiation and solve first-order ordinary differential equations in relevant scientific and engineering contexts.

Module-1

Complex Trigonometry: Complex Numbers: Definitions and properties. Modulus and amplitude of a complex number, Argand's diagram, De-Moivre's theorem (without proof).
Vector Algebra: Scalar and vectors. Addition and subtraction and multiplication of vectors- Dot and Cross products, problems.

Module-2

Differential Calculus: Review of successive differentiation-illustrative examples. Maclaurin's series expansions-Illustrative examples.
Partial Differentiation: Euler's theorem problems on first order derivatives only. Total derivatives-differentiation of composite functions. Jacobians of order two-Problems.

Module-3

Vector Differentiation: Differentiation of vector functions. Velocity and acceleration of a particle moving on a space curve. Scalar and vector point functions. Gradient, Divergence, Curl-simple problems. Solenoidal and irrotational vector fields-Problems.

Module-4

Integral Calculus: Review of elementaryintegral calculus. Reduction formulae for $\sin^n x$, $\cos^n x$ (with proof) and $\sin^m x \cos^n x$ (without proof) and evaluation of these with standard limits-Examples.

Module-5

Ordinary differential equations (ODE's. Introduction-solutions of first order and first-degree differential equations: exact, linear differential equations. Equations are reducible to exact and Bernoulli's equation.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Apply concepts of complex numbers and vector algebra to analyze the problems arising in related areas.
CO2	Apply derivatives and partial derivatives to calculate rate of change of multivariate functions.

CO3	Solve problems involving position, velocity, and acceleration in two and three dimensions using vector-valued functions.
CO4	Apply the fundamental techniques of elementary integration to solve standard integrals
CO5	Identify and solve first order ordinary differential equations.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE - IA Tests (A)	60	-	Sum of Two Internal Assess-ment Tests (MCQs) each con-ducted for 30 marks.
Continuous and Comprehensive Evaluation (B)	40	-	Any two Assessment methods as per 24RNBE4.2, scaled to 40 marks.
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks.
The minimum marks to be secured in CIE is 40 (40%) out of allotted 100 marks to Pass.			

Textbook:

1. Higher Engineering Mathematics by B.S. Grewal ,Khanna Publishers ,43rd Edition, 2015

Reference Books:

1. Advanced Engineering Mathematics by E. Kreyszig, John Wiley & Sons, 10th Edition, 2015
2. Engineering Mathematics by N. P. Bali and Manish Goyal, Laxmi Publishers, 7th Edition, 2007
3. .Engineering Mathematics Vol. Iby Rohit Khurana, Cengage Learning, 1st Edition, 2015

B.E. (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title:	**SKILL LAB : CYBERSECURITY FOUNDATIONS & SECURITY TOOLS LAB**		
Course Code:	**25BCYSL001**	**CIE MARKS**	**50**
(L: T: P) + SL	**0:0:2**	**SEE MARKS**	**50**
Credits:	**01**	**EXAM HOURS**	**02**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand fundamental cybersecurity concepts, threats, vulnerabilities, attacks, and security controls.
2. Apply operating system security, authentication, access control, and cryptographic techniques. Demonstrate the use of cybersecurity tools for network analysis and vulnerability assessment.
3. Implement basic security mechanisms including firewalls, endpoint hardening, and network segmentation.
4. Perform basic cybersecurity risk assessment and prepare security controls and remediation plans.

Module-1 **8 Hours**

Cybersecurity Fundamentals & Network Security :

Cybersecurity vs Information Security; CIA Triad; Authentication, Authorization & Auditing; threats, vulnerabilities, risks and attacks; malware, phishing, password attacks, social engineering, DoS and MITM; attack surface and security controls; networking fundamentals; OSI & TCP/IP models; IPv4, MAC/IP, subnetting and CIDR; TCP/UDP; common ports and protocols; ARP, DNS, DHCP; TCP three-way handshake; firewalls, NAT, VPN, network segmentation and DMZ.

Practical: Wireshark packet analysis; basic Nmap discovery; phishing/security scenario analysis.

Module-2 **9 Hours**

Operating System Security & Cryptography :

Windows and Linux security architecture; users, groups, privileges and permissions; authentication and password security; processes, services and logs; secure configuration; patch management; least privilege; privilege escalation concepts; cryptography fundamentals; plaintext/ciphertext; encryption/decryption; symmetric and asymmetric cryptography; AES, DES, RSA, ECC; hashing – MD5, SHA-1, SHA-256; digital signatures; digital certificates; PKI; HTTPS/TLS; password hashing and secure file transfer.

Practical: Linux/Windows security commands; process/service/log analysis; SHA-256 hashing; encryption and digital signature demonstration.

Module-3 **6 Hours**

Security Tools & Vulnerability Assessment Lab :

Introduction to Kali Linux and virtual lab environment;

Experiment 1: Kali Linux environment and basic security commands;

Experiment 2: Nmap – host discovery, port scanning, service identification and OS detection;

Experiment 3: Wireshark – packet capture, filters, TCP/DNS/HTTP analysis;

Experiment 4: GVM/OpenVAS – vulnerability scanning, CVE/CVSS and report interpretation; **Experiment 5:** Hashcat/John the Ripper – password hashing, password strength and authorized password auditing; **Experiment 6:** Security investigation using Linux/Windows CLI, Wireshark and Nmap.
Module-4 **4 Hours**
Perimeter, Endpoint & Security Defense : Defense-in-depth; network security architecture; firewalls and host-based firewalls; IDS vs IPS; endpoint security; antivirus/EDR concepts; secure configuration; patch management; application security; email security; web security fundamentals; network segmentation; DMZ; introduction to Zero Trust. **Practical:** Configure basic host firewall rules; identify open ports/unnecessary services; analyze firewall logs; prepare endpoint hardening checklist; design a simple secure network architecture.
Module-5 **3 Hours**
Governance, Risk Assessment & Capstone : Cybersecurity policies – Acceptable Use, Password, Access Control, Incident Response and Backup policies; security awareness; asset identification; threat identification; risk assessment; introduction to ISO 27001, NIST Cybersecurity Framework and CIS Controls. **Capstone:** Mini cybersecurity assessment involving asset identification → threat identification → vulnerability scanning → risk assessment → security controls → remediation plan → final report and presentation.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Explain fundamental cybersecurity concepts, threats, vulnerabilities, attacks, and security controls.
CO2	Apply operating system security, authentication, access control, and cryptographic techniques.
CO3	Employ cybersecurity tools for network analysis, packet inspection, vulnerability scanning, and security investigation.
CO4	Implement basic security mechanisms including firewalls, endpoint hardening, and network segmentation.
CO5	Perform cybersecurity risk assessment and develop security controls, remediation plans, and security reports.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE Practical (A)	30	-	Marks awarded for Conduction of Experiments and Submission of Reports, scaled to 30 marks.

CIE Practical Test(B)	20	-	Test conducted for 100 marks covering all experiments, scaled to 20.
Total CIE Practical (A+B)	50	20	Sum of Marks of Experiments, Report and Test shall be CIE marks awarded for a maximum of 50.
SEE (C)	50	20	SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in Components A, B and C
The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

Web links

1. https://www.nist.gov/cyberframework
2. https://nmap.org/docs.html
3. https://www.wireshark.org/docs/wsug_html
4. https://www.kali.org/doc

ESTD : 2001
An Institute with a Difference

IV SEMESTER SUBJECTS

B.E. (Common to all CSE streams)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

Course Title:	**DISCRETE MATHEMATICS AND GRAPH THEORY**		
Course Code:	**25BCSTH401**	**CIE MARKS**	**50**
(L: T: P) +SL	**(3:2:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:
The purpose of the course is to facilitate the learner to:

1. Apply the principles of mathematical logic and relations to analyze discrete mathematical problems.
2. Demonstrate the use of counting techniques and the principle of counting in solving combinatorial problems.
3. Familiarize with the fundamental concepts of principles of inclusion and exclusion to build a strong foundation in discrete mathematics.
4. Solve engineering optimization problems using appropriate mathematical and combinatorial techniques.

Module-1 **9 Hours**

Fundamentals of logic
Basic connectives and Truth tables, Tautologies, Logical Equivalence: The laws of logic, Logical implications, Rules of inference, Proof and disproof.
Text book 1: Chapter-2 (2.1 - 2.3, 2.5)

Module-2 **9 Hours**

Relations
Cartesian Products and Relations, Properties of Relations, Zero-One Matrices and Directed Graphs, Equivalence Relations and Partitions, Partial Orders — Hasse Diagrams, Lattices.
Text book 1: Chapter-5 (5.1), Chapter-7 (7.1 – 7.4)

Module-3 **9 Hours**

Fundamental Principles of Counting
The Rules of Sum and Product, Permutations, Combinations, Binomial and multinomial theorems, Combinations with Repetition.
Text book 1: Chapter-1 (1.1 - 1.4)

Module-4 **9 Hours**

The Principle of Inclusion and Exclusion
The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials, the Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients.
Text book 1: Chapter-8 (8.1 – 8.4), Chapter-10 (10.2)

Module-5 **9 Hours**

Introduction to Graph Theory: Definitions and Examples, vertex degree, Complete graph, Regular graph, Peterson graph, bipartite graph, complete bipartite graph, Sub-graphs, Regular graphs, Wheels, Subgraphs, Proper subgraphs, Spanning subgraphs, Induced sub graphs, Isomorphism of graphs, Konigsberg bridge problem, Trees, Binary trees, prefix code and merge-sort.

Textbook 1: Chapter 10 (10.1, 10.2), Chapter 12 (12.3, 12.4)

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Apply Mathematical logic to validate propositions.
CO2	Implement properties of relations in theoretical algorithms.
CO3	Model and solve computer science problems by applying combinatorial methods.
CO4	Solve counting problems involving finite sets using the principle of inclusion and exclusion.
CO5	Apply graph-theoretic concepts to develop solutions for network design problems.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 30 marks.
Continuous and Comprehensive Evaluation (B)	20	-	Any two Assessment methods as per 24RNBE4.2 (*if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	**50**	**20**	CIE Evaluation is done for a maximum of 50 marks
SEE (C)	**50**	**18**	SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components A, B and C.
• The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

Text Books:

1. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics: An Applied Introduction. 5th Edition, Pearson Addison Wesley, 2004.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, 1974.

Reference Books:

1. Chandrasekharaiah, D. S., Discrete Mathematical Structures, 5th Edition, PRISM Books Pvt. Ltd., Bengaluru, 2018.
2. Chandrasekharaiah, D. S.Graph Theory and Combinatorics. 4th ed. Bengaluru, India: PRISM Books

Pvt. Ltd. ISBN 9788172866983

3. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.
4. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics – A Concept based approach, Universities Press, 2016

ESTD : 2001
An Institute with a Difference

B.E (CSE and Allied Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

Course Title:	**DATABASE MANAGEMENT SYSTEMS**		
Course Code:	**25BCSIC402**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:0:2) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand the fundamental concepts, architecture, and advantages of Database Management Systems and their role in data handling.
2. Design and model database schemas using Entity-Relationship (ER) diagrams and map conceptual designs to relational models.
3. Write efficient SQL queries for data definition, manipulation, and retrieval, and understand NoSQL databases and their applications.
4. Apply normalization techniques to design well-structured relational database schemas and eliminate redundancy.
5. Comprehend transaction processing, concurrency control, and recovery techniques to ensure data integrity and reliability in database systems.

s **Module-1** **9 Hours**

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.

Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence.

Conceptual Data Modelling using Entities and Relationships: Entity types, Entity sets and structural constraints, Weak entity types, ER Diagrams, Naming Conventions, and Design Issues.

Textbook 1: Chapter-1 (1.1 - 1.7), Chapter-2 (2.1 - 2.2, 2.5 - 2.6), Chapter-3 (3.1 - 3.7)

Module-2 **9 Hours**

Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping.

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.

Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.).

Textbook 1: Chapter-9 (9.1), Chapter-5 (5.1 - 5.3), Chapter-8 (8.1 - 8.4)

Module-3 **9 Hours**

SQL: SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL.

More SQL: Complex Queries, Triggers, Views, and Schema Modification: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.

NOSQL Databases and Big Data Storage Systems: Introduction to NOSQL Systems, The CAP Theorem, Document-Based NOSQL Systems and MongoDB. **Textbook 1: Chapter-6 (6.1 - 6.5), Chapter-7 (7.1 - 7.3), Chapter-24 (24.1 - 24.3)**
Module-4 **9 Hours**
Normalization: Database Design Theory - Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Nor-mal Forms, Boyce-Codd Normal Form. **Relational Database Design Algorithms:** Functional Dependencies: Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decompositions. **Textbook 1: Chapter-14 (14.1 - 14.5), Chapter-15 (15.1 – 15.2)**
Module-5 **9 Hours**
Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. **Concurrency Control in Databases:** Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering. **Textbook 1: Chapter-20 (20.1 - 20.6), Chapter-21 (21.1 - 21.2)**

PRACTICAL COMPONENT OF IPCC

Sl. No.	Experiments
	PART-A
1	Consider the following schema for a Library Database: BOOK (Book id, Title, Publisher Name, Pub Year) BOOK AUTHORS (Book id, Author Name) PUBLISHER (Name, Address, Phone) BOOK COPIES (Book id, Branch id, No-of Copies) BOOK LENDING (Book id, Branch id, Card No, Date Out, Due Date) LIBRARY BRANCH (Branch id, Branch Name, Address) BORROWER (Card no, Name, Address, Phone) Write SQL queries to a. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc. b. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017. c. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation. d. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query. e. Create a view of all books and its number of copies that are currently available in the Library.

2	Consider the following schema for Order Database: SALESMAN (Salesman id, Name, City, Commission) CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id) Write SQL queries to a. Count the customers with grades above Bangalore's average. b. Find the name and numbers of all salesmen who had more than one customer. c. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.) d. Create a view that finds the salesman who has the customer with the highest order of a day. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.
3	Consider the schema for Movie Database: ACTOR (Act id, Act Name, Act Gender) DIRECTOR (Dir id, Dir Name, Dir Phone) MOVIES (Mov id, Mov Title, Mov Year, Mov Lang, Dir id) MOVIE CAST (Act id, Mov id, Role) RATING (Mov id, Rev Stars) Write SQL queries to a. List the titles of all movies directed by 'Hitchcock'. b. Find the movie names where one or more actors acted in two or more movies. c. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation). d. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. e. Update rating of all movies directed by 'Steven Spielberg' to 5.
4	Consider the schema for College Database: STUDENT (USN, SName, Address, Phone, Gender) SEMSEC (SSID, Sem, Sec) CLASS (USN, SSID) SUBJECT (Subcode, Title, Sem, Credits) IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA) Write SQL queries to a. List all the student details studying in fourth semester 'C' section. b. Compute the total number of male and female students in each semester and in each section. c. Create a view of Test1 marks of student USN '1RN25CS100' in all subjects. d. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students. e. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If Final IA < 12 then CAT = 'Weak'. Give these details only for 8th semester

	A, B, and C section students
5	Consider the schema for Company Database: EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo) DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate) DLOCATION (DNo, DLoc) PROJECT (PNo, PName, PLocation, DNo) WORKS_ON (SSN, PNo, Hours) Write SQL queries to a. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project. b. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. c. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department. d. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator). e. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.
6	Installation of an Open-Source NoSQL Database MongoDB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute MongoDB basic Queries using CRUD operations. https://www.mongodb.com/try/download/community

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Define the fundamentals of DBMS architecture, data models, and ER modelling concepts.
CO2	Construct ER diagrams into relational schemas and apply relational algebra operations.
CO3	Formulate SQL queries and differentiate between SQL and NoSQL databases.
CO4	Apply normalization techniques to design relational schemas up to BCNF.
CO5	Analyse transaction processing and implement concurrency control and recovery techniques.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE-IA Tests(A)	**20**	**-**	**Two tests each of 50 marks shall be conducted. Average of Two Internal**

			Assessment Tests shall be scaled down to 20 marks.
Continuous and Comprehensive Evaluation (B)	**10**	**-**	**Any one Assessment methods as per 24RNBE4.2 scaled to 10 marks.**
Total CIE Theory (A+B)	**30**	**12**	**Marks of Tests and Assignments awarded for a maximum of 30.**
CIE Practical	**12**	**-**	**Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 12 marks.**
CIE Practical Test	**8**	**-**	**Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 8.**
Total CIE Practical (C)	**20**	**8**	**Marks of Experiments, Record and Test awarded for a maximum of 20 marks.**
CIE: Theory + Practical(A+B+C)	**50**	**20**	**CIE Evaluation is done for a maximum of 50 marks.**
SEE (D)	**50**	**18**	**SEE is a theory exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall before 3 hours.**
CIE+SEE (A+B+C+D)	**100**	**40**	**Final marks shall be the sum of marks scored in Components A, B, C and D**

- The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE. The laboratory component of the ESC shall be for CIE only. However, in SEE, the questions from the laboratory component shall be included in their respective modules only.

Textbooks:

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

Reference Books:

1. Database Management Systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw, Hill.
2. Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, Mc-Graw Hill, 2013.

Web links and Videos:

1. NPTEL DBMS.
2. DBMS Playlist.
3. DBMS Lecture – Scaler.
4. GeeksforGeeks DBMS.

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

1. HackerRank SQL Practice
2. Case Study based flip class
3. Assignments
4. MongoDB Certification

Self-Study Topics:

1. Module 3: (Chapter 24.4 – 24.6)
   NOSQL Key-Value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j
2. Module 4: (Chapter 14.6 – 14.7)
   Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.
3. Module 5: (Chapter 21.3 – 21.5)
   Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking.

ESTD : 2001
An Institute with a Difference

B.E. (CSE and Allied Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

Course Title:	**DESIGN AND ANALYSIS OF ALGORITHMS**		
Course Code:	**25BCSTH403**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:2:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Learn the methods for analyzing algorithms and evaluating their performance.
2. Demonstrate the efficiency of algorithms using asymptotic notations.
3. 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. Learn the concepts of P and NP complexity classes

Module-1 **9 Hours**

Fundamentals: 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, Bubble Sort, Sequential Search and String Matching, Closest Pair points and Convex-Hull problems, Travelling Salesman Problem, Assignment Problem, Knapsack Problem,
Textbook 1: Chapter-2 (2.1-2.4), Chapter-3 (3.1-3.4)

Module-2 **9 Hours**

Decrease-and-Conquer: Topological Sorting, Fake Coin problem, Interpolation Search.
Divide-and-Conquer: Merge Sort, Quicksort.
Divide-and-Conquer: Max Min Algorithm.
Textbook 1: Chapter-4 (4.2, 4.4, 4.5), Chapter-5 (5.1-5.2)
Textbook 2: Chapter-3 (3.3)

Module-3 **9 Hours**

Transform and Conquer: Measure performance on data size greater 5000 and plot performance graph.
Greedy Technique: Prim's Algorithm, Kruskals Algorithm, Dijkstra's Algorithm, Huffman Trees, Knapsack Problem, Job sequencing with deadlines
Textbook 1: Chapter-2 (2.1-2.4) Chapter-3 (3.1-3.4)

Module-4 **9 Hours**

Dynamic Programming: Multistage Graph, The Knapsack Problem and Memory functions, Warshall's Algorithm and Floyd's Algorithms, Reliability Design.
Space and Time Trade-offs: Input Enhancement in String Matching: Horspool's algorithm, Boyer Moore's Algorithm.
Textbook 1: Chapter-8 (8.2-8.4), Chapter-7 (7.2)
Textbook 2: Chapter-5 (5.2-5.8)

Module-5	9 Hours
Backtracking: n-Queens Problem, Hamiltonian Circuit problem, Subset-Sum Problem, Graph coloring. **Branch-and-Bound:** Assignment Problem, Knapsack Problem, Traveling Salesman Problem, **Problems Complexity:** Limitations of Algorithm Power, P, NP and NP-Complete Problems, Approximation algorithms for NP Hard (Knapsack, Travelling, Salesman Problem). **Textbook 1: Chapter-12 (12.1-12.3), Chapter-11 (11.3), Textbook 2: Chapter-7 (7.4)**	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Explain the fundamentals of algorithm analysis, asymptotic notations, and brute-force techniques and analyze the efficiency of sorting algorithms such as Selection Sort, Merge Sort and Quick Sort.
CO2	Apply divide-and-conquer, decrease-and-conquer, transform-and-conquer, and greedy and graph algorithms (Topological Sort, Dijkstra's, Floyd's, Warshall's, Kruskal's and Prim's) to solve optimization and scheduling problems.
CO3	Apply dynamic programming, backtracking, and branch-and-bound techniques to design efficient solutions for optimization and combinatorial problems.
CO4	Analyze and compare the time and space complexities of different algorithmic paradigms to evaluate their suitability for solving various computational problems.
CO5	Analyze computational complexity classes (P, NP, NP-Complete) and implement exact, approximation and backtracking algorithms to solve computationally hard problems such as the N-Queens problem.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 30 marks
Continuous and Comprehensive Evaluation (B)	20	-	Any two Assessment methods as per 24RNBE4.2 (*if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	50	20	CIE Evaluation is done for a maximum of 50 marks
SEE (C)	50	18	SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.

CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in Components A, B and C.
The minimum marks to be secured in CIE are 20 (40%) out of allotted 50 marks to appear for SEE, with minimum 40 % scored in every component of CIE.			

Textbooks:

1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson 2017.
2. Ellis Horowitz, Sartaj Sahni, Rajasekaran, Fundamentals of Computer Algorithms, 2nd Edition 2019.

Reference Books:

1. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein Introduction to Algorithms, 4th MIT Press 2009**.**
2. Padma Reddy, Analysis and Design of Algorithms, 1st Sri Nandi Publications 2009.

ESTD : 2001
An Institute with a Difference

B.E. (CSE-CYBER SECURITY)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

Course Title:	**ELEMENTS OF CYBER SECURITY**		
Course Code:	**25BCYTH404**	**CIE MARKS**	**50**
(L: T: P) + SL	**(3:0:0) + (45 Hours/Sem)**	**SEE MARKS**	**50**
Credits:	**4**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. To learn about concepts and different types of cyber-crime and Mitigation.
2. To have an overview of the cyber security for Mobile Devices, Digital Payments, Email, Web and Wireless networks.
3. Introduction to basics of Cryptography.
4. To study the defensive techniques against Cyber attacks.

TEACHING-LEARNING PROCESS

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

1. Adopt interactive and alternative teaching methods including lectures, active learning, and case studies.
2. Use multimedia tools such as videos, animations, and manifold representations to explain concepts effectively.
3. Encourage collaborative learning, discussions, and HOT questions to promote critical thinking.
4. Implement Problem-Based Learning (PBL) and analytical approaches to enhance design and problem-solving skills.
5. Demonstrate multiple solution approaches and relate concepts to real-world applications for better understanding.

Module-1 **9 Hours**

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

Textbook 1: Chapter 1, 2
Textbook 2: Chapter 1, 2, 3, 4

Module-2 **9 Hours**

Mobile and Digital Payments Security: Security Challenges and types of attacks on Mobile devices, Security for Mobile Apps, Mobile Device Management tools and techniques.

Digital payments Security: Banking Cards, Unified Payment Interface (UPI), e-Wallets, Unstructured Supplementary Service Data (USSD), Aadhar enabled payments, Digital payments related common frauds and preventive measures.

Note: Aadhar Enabled Payments topic as a case study not for the examination point of view.

Textbook 2: Chapter 10, 11

Module-3 **9 Hours**

E-Mail Security: Pretty Good Privacy, S/MIME IP Security: IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange. **Textbook 3: Chapter 19,20**
Module-4 **9 Hours**
Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH). **Wireless Network Security:** Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security. **Textbook 3: Chapter 17, 18, 21, 22**
Module-5 **9 Hours**
Cryptography Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks. **Textbook 3: Chapter 1, 2, 3, 4, 8, 10**

COURSE OUTCOMES:

After successfully completing the course, the student will be able to

CO1	Explain fundamental cyber security concepts, threats, vulnerabilities, and attack vectors in modern computing environments.
CO2	Demonstrate the identification and simulation of common cyber attacks using appropriate security tools and techniques.
CO3	Analyze security challenges, attacks, and preventive measures associated with mobile devices and digital payment systems.
CO4	Illustrate the principles and mechanisms of email, web, and wireless network security, including secure communication protocols.
CO5	Apply fundamental cryptographic concepts and techniques to secure information and evaluate basic encryption mechanisms.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Two tests each of 50 marks shall be conducted. Average of Two Internal Assessment Tests shall be scaled down to 30 marks.
Alternate Assessment (B)	20	-	Any two Assessment methods as per 24RNBE4.2 (*if it is project based, one assignment shall be given), scaled to 20 marks.
Total CIE (A+B)	50	20	CIE Evaluation is done for a maximum of 50 marks.
SEE (C)	50	18	SEE is a theory exam conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.

CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in Components A, B and C.
The minimum marks to be secured in CIE are 20 out of allotted 50 marks to appear for SEE, with minimum 40 % scored in every component of CIE.			

Textbooks:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&F Group.
3. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education.

Weblinks and Videos:

1. https://www.npci.org.in/product/upi?utm_source=chatgpt.com
2. https://www.cert-in.org.in?utm_source=chatgpt.com
3. https://www.nist.gov/cyberframework
4. https://www.npci.org.in/what-we-do/upi/product-overview
5. https://www.ietf.org/topics/security/
6. https://owasp.org/
7. https://www.nist.gov/cryptography

Activity-Based Learning:

1. Case study about common cyber crimes- discussion.
2. Programming Assignments.

Self-Study Topics:

1. Analysis of Recent Cyber Attacks and Data Breaches
2. Security Features and Fraud Prevention in UPI and Digital Payment Systems
3. Email Security Mechanisms: SPF, DKIM, and DMARC
4. OWASP Top 10 Web Application Security Risks
5. Applications of Cryptography in Blockchain and Digital Signatures

B.E (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
Course Title:	**DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY**		
Course Code:	**25BCSLB405**	**CIE MARKS**	**50**
L :T : P	**0:0:2**	**SEE MARKS**	**50**
Credits:	**1**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Learn the methods for analysing algorithms and evaluating their performance to demonstrate the efficiency of algorithms using asymptotic notations.
2. Demonstrate the efficiency of algorithms using asymptotic notations.
3. 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. Learn the concepts of P and NP complexity classes.

Program 1. Selection Sort – Sorting Student IDs

Design and implement a C program to sort a list of student roll numbers from a university database using the Selection Sort algorithm.

Objective: Record and compare the time taken for sorting as the number of student records increases ($n > 5000$).

Application: Helps in managing ordered student data efficiently for attendance systems or result generation.

Program 2. Merge Sort – Organizing Medical Reports

Design and implement a C program to sort large sets of patient report IDs using Merge Sort.

Objective: Track execution time for datasets with $n > 5000$ and plot time vs data size.

Application: Efficient sorting of medical records in a hospital management system.

Program 3. Quick Sort – E-Commerce Order Sorting

Design and implement a C program to sort customer order IDs based on timestamps using Quick Sort.

Objective: Measure performance on data size > 5000 and plot performance graph.

Application: Fast processing of online orders in an e-commerce platform

Program 4. Topological Sort – Task Scheduling in Project Management

Implement a C program to find the topological ordering of tasks in a project where some tasks depend on the completion of others.

Application: Automating task scheduling tools like Jira or Trello where dependencies exist.

Program 5. 0/1 Knapsack – Cargo Optimization

Implement a C program to solve the 0/1 Knapsack problem to maximize the value of items packed in a delivery truck with weight constraints.

Application: Logistic companies optimizing truck loads with limited capacity

Program 6. Greedy Knapsack – Stock Refill Strategy

Implement a C/C++ program to solve discrete and fractional knapsack problems using greedy methods to choose items for shelf restocking in a supermarket based on value-to-weight ratio.

Application: Inventory management and optimization.
Program 7. Dijkstra's Algorithm – GPS Navigation System
Implement a C/C++ program to find shortest paths from a given location to all other locations in a city's road network. **Application:** GPS systems like Google Maps, for calculating the shortest routes.
Program 8a. Floyd's Algorithm – Travel Time Optimization Problem:
Implement a C/C++ program to find the shortest paths between all pairs of cities in a country using Floyd's algorithm. **Application:** Railway or airline scheduling and route planning.
Program 8b. Warshall's Algorithm – Webpage Reachability
Implement a C/C++ program to determine reachability between all webpages using Warshall's algorithm on a hyperlink graph. **Application:** Search engines identifying which pages link to others
Program 9. Kruskal's Algorithm – Designing Network Layouts
Implement a C/C++ program to find the Minimum Cost Spanning Tree of a network using Kruskal's algorithm. **Application:** Laying cables (like fiber optic) for minimum cost in a connected campus or city.
Program 10. Prim's Algorithm – Power Grid Optimization
Implement a C/C++ program to build the minimum cost electricity network using Prim's algorithm. **Application:** Designing cost-effective electricity distribution networks.
Program 11. Subset Sum – Payment Combinations
Implement a C/C++ program to find all payment combinations using a set of coin or currency denominations that sum up to a required amount. **Application:** ATM cash dispensing or vending machines.
Program 12. N-Queens – Security Camera Placement
Implement a C/C++ program to solve the N-Queens problem to place security cameras in a museum grid such that no two cameras cover the same row, column, or diagonal. **Application:** Surveillance system optimization.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Analyse and implement sorting algorithms such as Selection Sort, Merge Sort, and Quick Sort to solve domain-specific problems.
CO2	Apply graph-based algorithms such as Topological Sort, Dijkstra's, Floyd's, Warshall's, Kruskal's, and Prim's algorithms to solve real-time optimization and scheduling problems.
CO3	Design and develop efficient solutions using Dynamic Programming techniques for combinatorial problems.
CO4	Solve resource allocation problems using Greedy algorithms and demonstrate their effectiveness in constrained environments.
CO5	Implement backtracking techniques to solve constraint satisfaction problems such as the N-Queens problem.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE Practical (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
CIE Practical Test (B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE Practical (A+B)	**50**	**20**	**Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.**
SEE (C)	**50**	**20**	**SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.**
CIE + SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components **A, B and C.**
• The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

Web links:

1. Virtual Labs (CSE): http://cse01-iiith.vlabs.ac.in/

ESTD : 2001
An Institute with a Difference

B.E (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System(CBCS) SEMESTER - IV			
Course Title:	**ADVANCED JAVA**		
Course Code:	**25BCYAE406A**	**CIE MARKS**	**50**
(L: T: P) + SL	**(0:0:2) + (15 Hours/ Sem)**	**SEE MARKS**	**50**
Credits:	**1**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand and demonstrate advanced Java programming concepts such as multithreading, synchronization.
2. Apply Java collection framework and user-defined classes.
3. Design and develop web applications using Spring Boot.
4. Construct database interactions using Java Database Connectivity.

Program 1 - Thread Synchronization in Simulation Processing
Thread synchronization in simulation processing: Implement thread synchronization where one thread collects sensor data, another processes the data after collection is complete and the main thread waits for processing to finish before displaying the final simulation status. **Example:** Synchronize data collection and processing threads using thread coordination mechanisms.
Program 2 - Custom Sorting Based on Tens Place Value
Custom sorting based on tens place value: Generate a set of random numbers and sort them according to the digit present at the tens place using a custom sorting mechanism. **Example:** Sort randomly generated numbers based on their tens place values.
Program 3 - Student Record Management using Collections
Student record management using collections: Develop a student management system using a user-defined class containing USN, Name and Branch and store multiple student objects using the Java Collection Framework. **Example:** Store and display multiple student records using an ArrayList.
Program 4 - Library Book Management using Collection Framework
Library book management using collection framework: Develop a library management system to store book details such as Book ID, Title, Author and Price using the Java Collection Framework. **Example:** Store and display book records using an ArrayList or LinkedList.
Program 5 - GUI Event Handling using Swing
GUI event handling using Swing: Develop a Swing application with two buttons, ”Alpha” and ”Beta” and display an appropriate message when either button is clicked using event handling. **Example:** Demonstrate Action Listener to respond to button click events.

Program 6 - Cookie Creation and Retrieval
Cookie creation and retrieval: Create a cookie named ”RNSIT” with the value”RN-SIT123” and retrieve the stored cookie information from the client browser. **Example:** Create and read browser cookies using Java web technologies.
Program 7 - Login Management using Cookies and Sessions in JSP
Login management using cookies and sessions in JSP: Develop a JSP application that stores the username in a cookie and maintains the login session using HTTP Session. **Example:** Display a personalized welcome message after successful login using cookies and sessions.
Program 8 - Form Handling using Spring Boot and Spring MVC
Form handling using Spring Boot and Spring MVC: Develop a Spring Boot application that accepts user input through an HTML form and displays a personalized greeting using a Spring MVC controller. **Example**: Accept a username from a form and display a greeting on the next page.
Program 9 - Automated Student Result Computation
Automated student result computation: Develop a web application that accepts student details, calculates the total marks automatically and displays the final result. **Example:** Enter student marks through a web form and generate the result instantly.
Program 10 - Fetching Student Records from MySQL Database
Fetching student records from MySQL database: Develop a Java application that connects to a MySQL database, retrieves all student records and displays them on the console. **Example:** Fetch and display all student details using JDBC.
Program 11 - Secure Database Operations using JDBC Prepared Statement
Secure database operations using JDBC Prepared Statement: Develop a Java application that securely inserts and deletes book records using JDBC Prepared Statement. **Example:** Insert a new book record and delete an unavailable book from the database using Prepared Statement.
Program 12 - Student Database Management using JDBC INSERT and DELETE
Student database management using JDBC INSERT and DELETE: Develop a Java application that performs INSERT and DELETE operations on a student database using JDBC. **Example:** Add new student records and remove outdated records from the database.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Apply advanced Java concepts such as multithreading and synchronization..
CO2	Implement Java programming concepts using collections and custom sorting techniques.
CO3	Construct applications using the Java Collection Framework to manage user-defined objects.

CO4	Design GUI applications using Swing and event handling.
CO5	Develop database-driven and web applications using JDBC, JSP, Cookies, Ses-sions and Spring Boot.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
Alternate Assessment (B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE (A+B)	50	20	Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.
SEE (C)	50	20	SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in Components A, B and C.
The minimum marks to be secured in CIE are 20 out of allotted 50 marks to appear for SEE, with minimum 40% scored in every component of CIE.			

Textbooks:

1. Herbert Scheldt: JAVA the Complete Reference. 12th Edition, Tata McGraw-Hill. 2021.
2. Jim Keogh, The Complete Reference J2EE, 1st Edition Tata McGraw-Hill 2007.
3. Kathy Sierra, Bert Bates, Headfirst Java, O'Reilly, 2nd Edition.2005.

Reference Books:

1. Y. Daniel Liang: Introduction to JAVA Programming, 7th Edition, Pearson Edu-cation, 2007.
2. Stephanie Bod off et al: The J2EE Tutorial, 2nd Edition, Pearson Education,2004.
3. Uttam K Roy, Advanced JAVA programming, 1st Edition Oxford University press, 2015.

B.E. (Computer Science and Engineering and Allied Branches)
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

Course Title:	**INTRODUCTION TO UNIX**		
Course Code:	**25BCYAE406B**	**CIE MARKS**	**50**
L: T : P	**0:0:2**	**SEE MARKS**	**50**
Credits:	**1**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Explain the fundamental structure, architecture, components, and key features of the Unix/Linux operating system.
2. Demonstrate proficiency in using the Unix/Linux command-line interface (CLI) to navigate the file system, manage files and directories, and perform basic system operations.
3. Develop and execute shell scripts using variables, decision-making statements, loops, and Unix commands to automate routine tasks.
4. Apply Unix/Linux system administration utilities to manage processes, files, users, permissions, and monitor system resources effectively.

Program 1 – Unix Commands

A Linux System Administrator is assigned to perform a routine system inspection before carrying out maintenance activities on a server. Develop and execute a Bash script that displays the current working directory, lists all files including hidden files in the current directory, sorts the files in descending order based on their size, displays the current date and time in detailed format, and shows the name of the currently logged-in user. The script should produce clearly labeled output for each task and help the administrator quickly obtain essential system information in an organized manner.

Program 2 - File and Directory Operations

A project coordinator is responsible for organizing project documents in a Unix/Linux environment before the start of a new development cycle. Develop and execute a Bash script that creates a nested directory structure named project/year2025/month07/week1, generates multiple text files inside the week1 directory using a loop, moves and renames these files into a new directory named archive, creates a backup by copying the entire archive directory to a new location named backup, and finally deletes all .txt files from the archive directory using appropriate wildcard patterns. The script should automate the complete file management process efficiently and display suitable messages for each operation.

Program 3 - File Permissions and Ownership

An System Engineer is required to secure a newly created file by modifying its access permissions and ownership before making it available to other users. Develop and execute a Bash script that creates a sample file named example.txt, displays its original permissions using the long listing format, changes the file permissions using both symbolic and numeric modes, displays the updated permissions after each modification, and optionally changes the ownership of the file to another user if appropriate privileges are available. The script should clearly show the permission details before and after each operation and include suitable messages to explain each step.

Program 4 - Text Processing Utilities

A college administrator has been provided with a tabseparated file named students.txt containing student records such as Roll Number, Name, Department, and Marks. Develop and execute a Bash script to analyze and process the file using Unix/Linux text-processing utilities. The script should display the complete file contents, show the first three and last two records, extract only the student names, convert the names to uppercase, sort the records based on marks in ascending order, display the list of unique departments, print the names of students who scored more than 80 marks, and replace the department name ”Computer” with ”CSE”. Ensure that the output of each command is clearly labeled to make the generated report easy to understand.

The contents of students.txt are as follows:

Roll.No	Name	Department	Marks
1.	John	Computer	85
2.	Ravi	Electronics	78
3.	Asha	Computer	92
4.	Maya	Electronics	85
5.	Arjun	Mechanical	78.

Program 5 - Redirection and Pipelines

A Human Resources (HR) executive has been provided with a comma-separated file named data.txt containing employee records with their ID, Name, and Department. The executive needs to identify employees belonging to the Computer department and generate a formatted report using Unix/Linux command-line utilities. Develop and execute a Bash script that uses input/output redirection and pipelines to read the data from the file, extract only the records belonging to the Computer department, retrieve the employee names, convert them to uppercase, save the intermediate output to a file named filtered names txt, sort the names, count the occurrence of each name, and display the final report on the terminal. The script should demonstrate the use of grep, cut, tr, sort, uniq redirection, and pipes, ensuring that each stage of processing is clearly labeled. The contents of data txt are as follows:

ID	Name	Department
1.	John	Computer
2.	Ravi	Electronics
3.	Asha	Computer
4.	Maya	Electronics
5.	Arjun	Mechanical
6.	David	Computer

Program 6 - File Search and Log Analysis
An IT Infrastructure Engineer is assigned to inspect a server's file system to locate important files and identify errors recorded in log files before performing system maintenance. Develop and execute a Bash script that recursively searches the current directory to locate a specific file, report.txt, identifies files of specific types such as directories or symbolic links, and finds files larger than 1 MB. The script should also search all .log files for lines containing the keyword "ERROR", display the filename along with each matching line, and save the complete error report to a file named error report.txt. Use appropriate find options (-name, -type, -size, etc.) and grep options (-r, -i, -n, --include, etc.) to perform the required tasks. Ensure that the script contains meaningful comments and clearly labeled output for each operation.
Program 7 - Process Management
A Server Administrator is responsible for monitoring and managing system processes on a production server to ensure efficient resource utilization and proper job control. Develop and execute a Bash script that displays a snapshot of the currently running processes, identifies the top five memory-consuming processes by showing their PID, User, Command, starts a long-running process in the background, displays its job ID and PID, monitors it using appropriate commands, brings the background job to the foreground, suspends and resumes the process using Unix/Linux job control commands, and finally terminates the process gracefully. The script should include meaningful comments, clearly labeled output, and log each step to demonstrate effective process monitoring and management.
Program 8 - Process Monitoring and Job Control
A Linux System Administrator is testing Unix/Linux process monitoring and job control features to ensure that background tasks can be effectively managed during system maintenance. Develop and execute a Bash script that displays the current list of running processes, identifies the top five memory-consuming processes by showing their PID, USER, % MEM, and COMMAND, starts a dummy long-running process in the back-ground, displays the background job and its corresponding PID, brings the job to the foreground, suspends and resumes the process using appropriate job control commands, and finally terminates the process gracefully. The script should include meaningful comments and clearly labeled output to explain each stage of process monitoring and job control.
Program 9 - Shell Variables and Quoting
A DevOps engineer is preparing a training session to help new team members understand how the Unix/Linux shell handles variables, environment variables, command substitution, and different quoting mechanisms. Develop and execute a Bash script that creates and displays shell variables, performs simple arithmetic operations, converts a shell variable into an environment variable using the export command and verifies it using printenv or env, demonstrates the behavior of single quotes, double quotes, backticks ( ), and $ () for command substitution, and includes examples that clearly illustrate how the shell interprets quoted and unquoted text differently. The script should produce descriptive and well-labeled output for each demonstration.

Program 10 - Decision Making, Loops, and File Handling
A software trainee is developing a Unix/Linux shell script to practice decisionmaking, loops, and file handling as part of a programming assignment. Develop and execute a Bash script that accepts a number from the user, determines whether it is even or odd, checks whether it is divisible by 3, uses a loop to display all even numbers between 1 and 50, and traverses the current directory to list the names of all files with the .txt extension. The script should use appropriate conditional statements, looping constructs, and file-handling commands, while displaying clearly labeled and user-friendly output for each task.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Apply fundamental Unix/Linux commands for file, directory, and system operations.
CO2	Develop shell scripts using variables, control structures, loops, and command substitution to automate Unix operations.
CO3	Apply Unix utilities for process management, text processing, and file permission management.
CO4	Analyze system requirements to automate tasks and monitor system performance using Unix scheduling tools.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests(A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
Alternate Assessment (B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE Practical(A+B)	**50**	**20**	**Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.**
SEE (C)	**50**	**20**	**SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.**
CIE+SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components **A, B and C.**

- The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.

Text Books:

1. Sumitabha Das , "Your UNIX/Linux: The Ultimate Guide" , 3rd edition, McGraw-Hill 2012
2. Sumitabha Das, "Unix: Concepts and Applications", 4th Edition, McGraw Hill, 2006.
3. W. Richard Stevens and Stephen A. Rago, "Advanced Programming in the UNIX Environment", 3rd Edition, Addison-Wesley, 2013

Reference Books:

1. Behrouz A. Forouzan and Richard F. Gilberg, "Unix and Shell Programming", 1st Edition, Cengage Learning, 2003.
2. William E. Shotts Jr., "The Linux Command Line: A Complete Introduction", 2nd Edition, No Starch Press, 2019.
3. Free Software Foundation, "GNU Bash Reference Manual", Edition 5.2, GNU Project, 2022.

ESTD : 2001
An Institute with a Difference

B.E. (Computer Science and Engineering (Cyber Security)) Outcome Based Education (OBE) and Choice Based Credit System(CBCS) SEMESTER - IV			
Course Title:	**DEVOPS-2**		
Course Code:	**25BCYAE406C**	**CIE MARKS**	**50**
(L: T: P) + SL	**0:0:2**	**SEE MARKS**	**50**
Credits:	**1**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Design and implement CI/CD pipelines using Jenkins, GitHub, Docker to automate application build and deployment.
2. Deploy and manage containerized applications using Docker Compose.
3. Provision and configure cloud infrastructure using Terraform and Ansible for application deployment.
4. Implement centralized logging and real-time monitoring using the ELK stack, Prometheus, and Grafana.
5. Apply GitOps and SRE practices such as blue-green deployment, alerting, incident response and cloud security measures.

Sl. No.	Experiments
1.	**Setup Docker Compose with multi-container configuration and private registry** Install Docker and Docker Compose on your local or virtual machine. Create a docker-compose.yml file defining at least two services (e.g., a web app and a database). Build Docker images for these services and push them to a private Docker registry. Pull the images from the registry and use Docker Compose to deploy and run the containers. Verify inter-service communication. **Chapter-7: Deploying the Code (Advanced Docker) pg. 132** • https://docs.docker.com/compose/ • https://docs.docker.com/registry/
2.	**Create a multi-node Jenkins pipeline with GitHub + Docker** Install Jenkins and configure a master-agent setup (on separate nodes or containers). Install necessary plugins like GitHub Integration and Docker Pipeline. Create a multi-branch pipeline project that pulls source code from GitHub. Define the Jenkinsfile to build a Docker image and deploy a container. Trigger the build and verify the deployment. **Chapter-5, Chapter-6, Chapter-7: Pipeline Orchestration pg. 132,41-52,63-80** https://www.jenkins.io/doc/book/installing/ • https://plugins.jenkins.io/github/s.io/github/ • https://www.jenkins.io/doc/book/pipeline/

3.	**Use Terraform to automate infrastructure provisioning** Install Terraform and AWS CLI. Configure credentials for AWS access. Write Terraform configuration files to provision an EC2 instance, a security group, and a VPC. Initialize and apply the Terraform code. Verify the infrastructure creation on AWS Console. Destroy the infrastructure using Terraform when done. **Module 2: Chapter-11(AWS + Infrastructure as Code) pg.469-480** • https://registry.terraform.io/providers/hashicorp/aws/latest/docs • https://developer.hashicorp.com/terraform/tutorials/aws-get-started
4.	**Provision and Access AWS EC2 Instance for Ansible Automation** To provision an EC2 instance on AWS manually and access it using SSH in preparation for Ansible-based automation. Launch an EC2 instance manually or using Terraform. Set up an inventory file and create Ansible playbooks to install necessary packages (like Nginx or Node.js) on the instance. **Chapter-11: AWS, IAM, EC2.**
5.	**Deploy Application on EC2 using Ansible Playbook** Deploy a sample application and ensure it is running and accessible via the public IP. Use SSH key authentication for connection. **Chapter-5 (Ansible) pg. 494** • https://docs.ansible.com/ansible/latest/scenario_guides/guide_aws.html • https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-automate-initial-server-setup-on-ubuntu-22-04
6.	**Setup Elastic Search, Logstash, and Kibana (ELK stack) for centralized logging** Install Docker Compose and write a configuration to deploy Elasticsearch, Logstash, and Kibana containers. Configure Logstash to collect logs from a running application or file. Run the stack and access Kibana in the browser to view indexed logs. Analyze logs and create basic visualizations. **Chapter-8,Chapter-9: Monitoring Logging pg. 418-435,335** • https://www.elastic.co/elastic-stack/ • https://logz.io/blog/elk-stack-on-docker/
7.	**Add system health checks using Sensu or Prometheus** Install and configure either Sensu or Prometheus monitoring stack. Deploy node exporters or checks on the target systems. Write basic alert rules for CPU, memory, or service failures. Simulate a failure to verify that the alert triggers properly. View metrics and alerts via the dashboard. **Chapter-10: Monitoring with Sensu pg.439-465** • https://prometheus.io/docs/introduction/overview/ • https://docs.sensu.io/sensu-go/latest/
8.	**Implement blue-green deployment with Jenkins + Ansible** Create two environments (blue and green) for the same application version using separate ports or instances. Write an Ansible playbook to deploy to both environments.

	Use Jenkins to automate deployment and switch traffic from blue to green using a load balancer (like Nginx or HAProxy). Confirm zero-downtime deployment. **Chapter-7: Deployment Strategies** • https://martinfowler.com/bliki/BlueGreenDeployment.html • https://www.jenkins.io/doc/tutorials/build-a-java-app-with-maven/
9.	**Simulate incident response and create alerts with log data** Deploy an application and generate abnormal behavior (e.g., failed login attempts, high response time). Ensure logs are captured via ELK or Prometheus. Configure alert rules based on error logs or performance metrics. Simulate an incident, receive alerts and prepare a brief incident response note. **Chapter-8: Monitoring, Chapter-9: Issue Tracking. Pg. 137-196** • https://prometheus.io/docs/alerting/latest/overview/ • https://www.elastic.co/docs/explore-analyze/alerts-cases
10.	**Apply security practices (IAM, SSH keys, access logs) in cloud deployments** Create IAM roles and policies for least privilege access. Launch EC2 instances with SSH key-pair for authentication. Enable CloudTrail and CloudWatch for logging API and system activities. Review logs to identify potential security events and ensure compliance with basic cloud security hygiene. **Chapter-11: IAM, Security, and Cloud Governance pg. 469-494** • https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html • https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html
Capstone Project – Semester 4 Deploy a containerized microservice app to AWS with full CI/CD pipeline, automated provisioning (Terraform), centralized logging (ELK), real-time monitoring (InfluxDB + Grafana), and role-based access control.	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Build automated CI/CD pipelines using Jenkins, GitHub and Docker.
CO2	Deploy and manage containerized applications with Docker Compose.
CO3	Provision and manage cloud infrastructure using Terraform and Ansible.
CO4	Implement centralized logging and monitoring using ELK, Prometheus and Grafana.
CO5	Apply GitOps and SRE practices including blue-green deployments, alerting and incident response.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE Practical (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
CIEPractical Test(B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE Practical (A+B)	**50**	**20**	Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.
SEE (C)	**50**	**20**	SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	**100**	**40**	Final marks shall be the sum of marks scored in Components A, B and C.
The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

Textbook:

1. Learning DevOps – Continuously Deliver Better Software by Mikael Krief.

eBook link:

https://github.com/rootusercop/Free-DevOps-Books-1/blob/master/book/Learning%20DevOps%20-%20Continuously%20Deliver%20Better%20Software.pdf

E- Learning Resources:

1. CICD-DevOps Automation and DevSecOps: https://elearn.nptel.ac.in/shop/iit workshops/completed/cicd-devops-automation-and- devsecops-automation/

2. DevOps Tutorial by 3RI Technologies: https://www.youtube.com/playlist?list=PLAkvN1xnxGzGwWfep3tvUzuX11pBEGO ex

Practical Based learning Mini Project:

- Project Based Learning

B.E. (Computer Science and Engineering -Cyber Security) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER - IV			
Course Title:	**INTRODUCTION TO CYBER SECURITY TOOLS**		
Course Code:	**25BCYAE406D**	**CIE MARKS**	**50**
(L: T: P) + SL	**0:0:2**	**SEE MARKS**	**50**
Credits:	**1**	**EXAM HOURS**	**3**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand the fundamentals of cybersecurity tools and Linux environments.
2. Apply network scanning, traffic analysis, and vulnerability assessment techniques.
3. Explore web security, phishing, malware analysis, and cryptographic tools.
4. Develop basic incident investigation, reporting, and response skills.

Sl. No.	Experiments
1.	Install **Kali Linux** and explore basic Linux commands, package management, and commonly used cybersecurity tools.
2.	Perform **web server vulnerability assessment** using **Nikto**. Identify insecure configurations, outdated software, and common web server vulnerabilities.
3.	Conduct phishing awareness simulations using Social-Engineer Toolkit (SET) or GoPhish.
4.	Extract and analyze **digital file metadata** using **ExifTool**. Examine timestamps, GPS coordinates, device information, and other metadata for forensic investigations.
5.	Perform SQL Injection testing in a controlled environment using DVWA and Burp Suite.
6.	Conduct a ransomware tabletop exercise based on an insider threat scenario.
7.	Perform **network discovery** using **Netdiscover**. Identify active hosts, IP addresses, MAC addresses, and network topology within a local network.
8.	Perform Cryptanalysis of symmetric and asymmetric ciphers using Cryptool.
9.	Analyze suspicious URLs and IP addresses using VirusTotal and online sandbox tools.
10.	Prepare an Incident Response Report and Documentation for a simulated cyber attack scenario.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Demonstrate the use of Linux-based cybersecurity tools for system administration, reconnaissance, and network analysis.
CO2	Apply cybersecurity tools and techniques to identify vulnerabilities, analyze network traffic, and assess common web-based attacks.
CO3	Utilize cryptographic and threat intelligence tools to analyze ciphers, suspicious URLs, IP addresses, and phishing attacks.
CO4	Investigate cyber security incidents, evaluate attack scenarios, and prepare incident response reports and documentation.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Marks awarded for Weekly Conduction of Experiments and Submission of Laboratory records, scaled to 30 marks.
Alternate Assessment (B)	20	-	Average of marks of Two tests, each conducted for 100 marks covering all experiments, scaled to 20.
Total CIE (A+B)	50	**20**	Sum of Marks of Experiments, Record and Test shall be CIE marks awarded for a maximum of 50.
SEE (C)	50	**20**	SEE is a lab exam, conducted for 100 marks and scaled down to 50 marks. Exam duration shall be for 3 hours.
CIE + SEE (A+B+C)	100	**40**	Final marks shall be the sum of marks scored in Components **A, B and C.**
● The minimum marks to be secured in CIE are 20 out of allotted 50 marks to appear for SEE, with minimum 40 % scored in every component of CIE.			

Text Books:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Olson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa (John) Wu and J. David Irwin, CRC Press.

Reference Books:

1. The Basics of Hacking and Penetration Testing, Patrick Engebretson, Syngress.
2. Kali Linux Revealed, Ra+phael Hertzog and Jim O'Gorman, Offensive Security.
3 Network Security Essentials: Applications and Standards, William Stallings, PearsonEducation.
4 Incident Response and Computer Forensics, Jason T. Luttgens, Matthew Pepe, and Kevin Mandia,

McGraw-Hill.

Web Links:

1. https://www.kali.org
2. https://nmap.org
3. https://www.wireshark.org
4. https://owasp.org
5. https://www.virustotal.com
6. https://portswigger.net/web-security

ESTD : 2001
An Institute with a Difference

B.E. (Common to all branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
Course Title:	**BIOLOGY FOR ENGINEERS**		
Course Code:	**25BFETH407**	**CIE MARKS**	**50**
(L: T: P) + SL	**2:0:0**	**SEE MARKS**	**50**
Total hours of pedagogy:	**30**	**TOTAL MARKS**	**100**
Credits:	**2**	**EXAM HOURS**	**2**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Familiarize the students with the basic biological concepts and their engineering applications.
2. Enable the students with an understanding of bio design principles to create novel devices and structures.
3. Provide the students with an appreciation of how biological systems can be re-designed as substitute products for natural systems.
4. Motivate the students to develop interdisciplinary vision of biological engineering.

Module-1	**6 Hours**
Introduction to fundamentals of life: Structure and functions of a cell. Biomole cules: Properties and functions of Carbohydrates, Nucleic acids, Proteins, Enzymes and Lipids,.	
Module-2	**6 Hours**
Introduction to applications of biomolecules: Carbohydrates in cellulose-based water filters production, PHA and PLA in bioplastics, Nucleic acids in vaccines and diagnosis, Proteins in food production, Enzymes in biosensors fabrication, Genome & protein data types.	
Module-3	**6 Hours**
Introduction to adaptations of anatomical principles for bioengineering design - Brain as a CPU system. Eye as a Camera system. Heart as a pump system. Lungs as purification system. Kidney as a filtration system.	
Module-4	**6 Hours**
Introduction to nature inspired biomaterials and mechanisms - Echolocation, Photosynthesis, Bird flying, Lotus leaf effect, Shark skin, Kingfisher beak. Artificial blood - haemoglobin-based oxygen carriers (HBOCs).	
Module-5	**6 Hours**
Introduction to trends in bioengineering, Bioprinting techniques and materials. Electrical tongue and electrical nose in food science, DNA origami and Biocomputing, Bioimaging and Artificial Intelligence for disease diagnosis. Bio-concrete.	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Corroborate the concepts of biomimetics for specific requirements.
CO2	Understand the basic biological concepts for relevant industrial applications & adaptations.
CO3	Evaluate the principles of design and development for exploring engineering knowledge in biological stream.
CO4	Explore innovative biobased solutions for socially relevant problems.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	30	-	Average of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Continuous and Comprehensive Evaluation (B)	20	-	Poster presentation to satisfy SDG goals & coursera certification (10 + 10).
Total CIE (A+B)	50	20	CIE Evaluation is done for a maximum of 50 marks.
SEE (C)	50	18	Question papers having 50 MCQs with each question carrying 01 mark, and Exam duration shall be for 2 hours.
CIE + SEE **(A+B+C)**	100	40	Final marks shall be the sum of marks scored in Components **A, B and C.**
The minimum marks to be secured in CIE is 20 out of 50 allotted marks to appear for SEE, with minimum 40 % scored in every component of CIE.			

Textbooks:

1. Basuchandra's Biology for Engineers, Chandra Shekara. B.M, Basavaraju B.C, 2025.
2. Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.

Reference Books:

1. Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022

2. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W.
3. Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.
4. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
5. Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
6. Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
7. Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
8. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
9. Bioremediation of heavy metals: bacterial participation, by C R Sunil kumar, N Geetha, A C Udaya shankar, Lambert Academic Publishing, 2019.

ESTD : 2001
An Institute with a Difference

B.E. (Common to All Branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
Course Title:	**NATIONAL SERVICE SCHEME**		
Course Code:	**25BNSSP409**	**CIE MARKS**	**100**
(L: T: P) + SL	**(0:0:0)+(30 Hours/Sem)**	**SEE MARKS**	**0**
Credits:	**0**	**EXAM HOURS**	**-**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Identify the needs and problems of the community and participate in problem-solving activities.
2. Develop a sense of social and civic responsibility and utilize knowledge to find practical solutions to individual and community problems.
3. Develop the competencies required for group living and sharing responsibilities, and gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.

Module-1
Environment Issues and Role of NSS
Module-2
Disaster Management and Role of NSS
Module-3
Activities -1 (Choose from Activity list)
Module-4
Activities -2 (Choose from Activity list)
Module-5
Activities -3 (Choose from Activity list)

ACTIVITY LIST	
Helping local schools achieve good results and enhance their enrolment in Higher/Technical/Vocational Education.	Preparing an actionable business proposal for enhancing the village income.
Reduction in energy consumption	Preparing an actionable business proposal for enhancing the village's income.
Creating awareness about rainwater harvesting i urban and rural areas.	Developing a Sustainable Water Management system.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Understand the motto and concepts of NSS.
CO2	Develop overall personality of volunteers and make them as leaders and responsible citizens of our nation.
CO3	Analyze the environmental and societal problems/issues and will be able to design solutions for the same.
CO4	Plan and implement government initiatives or self-driven projects effectively in real-life community settings.
CO5	Communicate effectively with the community and with the officials.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE - IA Tests (A)	60	-	Sum of Two Internal Assessment Tests (MCQs) each con-ducted for 30 marks.
Continuous and Compre-hensive Evaluation (B)	40	-	Any two Assessment methods as per 24RNBE4.2, scaled to 40 marks.
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks.
The minimum marks to be secured in CIE is 40 (40%) out of allotted 100 marks to Pass.			

Reference Books:

1. NSS Course Manual, Published by NSS Cell, VTU Belagavi.
2. Government of Karnataka, NSS cell, activities reports and its manual.
3. Government of India, NSS cell, Activities reports and its manual.

Weblinks:

1. https://nss.gov.in/
2. https://pledge.mygov.in/

B.E. (Common to All Branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
Course Title:	**PHYSICAL EDUCATION**		
Course Code:	**25BPEDU409**	**CIE MARKS**	**100**
(L: T: P) + SL	**(0:0:0)+(30 Hours/Sem)**	**SEE MARKS**	**---**
Credits:	**---**	**EXAM HOURS**	**---**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand the fundamental concepts of Physical Education, games and sports skills, Health, Nutrition and Fitness.
2. Familiarization of health-related Exercises, Sports for overall growth and development.
3. Participate in the competition at regional/state / national / international levels.
4. Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

Module-1: Title: Athletics - Triple Jumps **3 Hours**

Relays

1. Relay Race: Starting, Baton Holding/Carrying, Baton Exchange in between zone and Finishing.

Module-2: Title: Athletics - Triple Jumps **3 Hours**

Triple Jump

A) Approach Run
B) Take-off
C) Step
D) Flight in the Hop
E) Jump and Landing

Module-3: Title: Athletics – Throws **3 Hours**

Javelin Throw

A) Grip
B) Carry and Recovery (3/5 Impulse stride)
C) Release

Module-4: Title: CRICKET **3 Hours**

A. Fundamental skills

1. Batting - Forward Defense Stroke, Backward Defense Stroke, Off Drive, On Drive, Straight Drive, Cover Drive, Square Cut.
2. Bowling - Out-swing, In-swing, Off Break, Leg Break and Googly.
3. Fielding: Catching - The High Catch, The Skim Catch, The Close Catch and throwing at

the stumps from different angles. Long Barrier and Throw, Short Throw, Long Throw, Throwing on the Turn. 4. Wicket Keeping. **B. Rules and their interpretation and duties of the officials**
Module-5: Title: BASEBALL **3 Hours**
A. Fundamental skills 1. Player Stances – walking, extending walking, L stance, cat stance. 2. Grip – standard grip, choke grip. 3. Batting – swing and bunt. Pitching. 4. Baseball: slider, fast pitch, curve ball, drop ball, rise ball, change up, knuckle ball, screw ball. **B. Rules and their interpretation and duties of the officials**

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Demonstrate the techniques of relay races, including starting, baton holding, exchange within the zone and finishing, along with basic skills of javelin throw.
CO2	Apply the fundamental techniques of triple jump, including approach run, take-off, hop, step, jump, flight and landing.
CO3	Develop and perform javelin throw techniques, including grip, carry, impulse stride, release and recovery with proper coordination.
CO4	Exhibit fundamental skills of cricket, including batting, bowling, fielding, wicket keeping and understand rules and duties of officials.
CO5	Demonstrate basic skills and techniques in baseball, including player stances, grip, batting, pitching and understand rules and officiating.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	—	Sum of Two Internal Assess-ment Tests (MCQs) each con-ducted for 30 marks.
Project (B)	40	—	———
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks.
• The minimum marks to be secured in CIE is 40 (40%) out of 100 marks allotted to Pass.			

Textbooks:

1. Health and Physical Education by V.K. Sharma.
2. Introduction to Physical Education, Fitnessand Sport by Daryl Siedentop.
3. Physical Education Activities, Sports and Games: The Origin and Background, Rules of Play, Teaching Hints and Fundamentals by Louis E. Means.

Reference Books:

1. Saraswati Health and Physical Education Textbook.
2. Foundations of Physical Education, Exercise Science and Sport by Deborah A. Wuest.

ESTD : 2001
An Institute with a Difference

B.E. (Common to All Branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER– IV			
Course Title:	**YOGA**		
Course Code:	**25BYOGA409**	**CIE MARKS**	**100**
(L: T: P) + SL	**(0:0:0)+(30 Hours/Sem)**	**SEE MARKS**	**---**
Credits:	**---**	**EXAM HOURS**	**---**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Physical, Mental & spiritual Objectives of Yoga
2. Explain the meaning of Yoga.
3. Discuss the importance of the Yoga and its scope.
4. Describe the aims and objectives of Yoga.
5. Clarify the misconceptions on Yoga

Module-1: Title: Suryanamaskara **3 Hours**

Suryanamaskara - Revision of practice 12 count 8 rounds

1. Suryanamaskar prayer.
2. Meaning, need & Importance
3. Benefits of Suryanamaskara

Module-2: Title: Asanas **3 Hours**

Need & importance of different Asana, Asana its meaning by name, technique, precau-tionary measures and benefits of each asana.

Sitting:

1. Bakasana
2. Hanumanasana
3. Yogamudra in Vajrasana

Standing:

1. Vatayanasana
2. Garudasana

Module-3: Title: Asanas **3 Hours**

Need, importance of Asana. Different types, Asana by name, technique, precautionary measures and benefits of each asana

Prone line:

1. Bhujangasana
2. Shalabh asana

Supine line:

1. Utthita dvipadasana,

2. Ardha halasana

Balancing:

1. Sheershasana

Module-4: Title: Pranayama	**3 Hours**
Kapalabhati - Revision of Kapalabhati practice 80 strokes/min 3 rounds	
Module-5: Title: Pranayama	**3 Hours**

Pranayam

1. Bhastrika
2. Bhramari

Meaning, Need, importance of Pranayama. Different types. Meaning by name, technique, precautionary measures and benefits of each Pranayama

COURSE OUTCOMES

After successfully completing the course, the student will be able to:

CO1	Understand the fundamental concepts of Yoga
CO2	improves of health and related aspects, overall growth and development
CO3	Participate in the competition at regional/state / national / international levels.
CO4	Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	----	Sum of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Quiz (B)	40	----	Quiz for a maximum of 40 marks
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks

- The minimum marks to be secured in CIE is 40 (40%) out of 100 marks allotted to Pass.

Textbooks:

1. The Yoga Sutras of Patanjali by Sri Swami Satchidananda.
2. Light on Yoga by B.K.S. Iyengar.

Reference Books:

1. The Heart of Yoga — T.K.V Desikachar.
2. "Asana Pranayama Mudra Bandha" by Swami Satyananda Saraswati.

ESTD : 2001
An Institute with a Difference

B.E. (Common to All Branches) Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
Course Title:	**MUSIC**		
Course Code:	**25BMUSK409**	**CIE MARKS**	**100**
(L: T: P) + SL	**(0:0:0)+(30 Hours/Sem)**	**SEE MARKS**	**---**
Credits:	**---**	**EXAM HOURS**	**---**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Understand the significance of music in daily life and gain an overview of Indian folk, contemporary, and Western music traditions.
2. Acquire basic knowledge of introductory Carnatic music exercises and compositions,
3. Compare the fundamental features of Carnatic and Hindustani music.
4. Recognize the classification of wind instruments and appreciate the contributions of eminent composers to Indian classical music.

Module-1	**3 Hours**
Importance of music in daily life ,Indian folk, and contemporary music.Introduction to Western music.	
Module-2	**3 Hours**
Compositions: Introduction to the few types of composition in Carnatic Music- Nottu Swara, Sarale and janti varase, Alakara.	
Module-3	**3 Hours**
Comparative Study of Carnatic and Hindustani Music based on Geographical origin and development, Musical structure and style ,Raga and Tala systems.	
Module-4	**3 Hours**
Music Instruments: Classification and construction of wind instrurnents in both Carnatic and Hindustani Music.	
Module-5	**3 Hours**
Composers: Biography and contributions of of the following eminent personalities Muttuswami Dikshitaru , Shyama Shastri, Mysore Vasudevacharya, Pt. Bheemasen Joshi.	

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Explain the origin, evolution and fundamental concepts of Indian classical music.
CO2	DCompare and distinguish the characteristics of selected Carnatic and Hindustani music compositions.
CO3	Classify the major categories of Indian musical instruments and describe their construction and applications in Carnatic and Hindustani music.
CO4	Examine the contributions of eminent Composers/Musicians and appreciate their role in preserving and promoting Indian classical music.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – IA Tests (A)	60	-	Sum of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Project (B)	40	-	Submission of report on a topic related to music.
Total CIE (A+B)	**100**	**40**	CIE Evaluation is done for a maximum of 100 marks.
• The minimum marks to be secured in CIE is 40 (40%) out of 100 marks allotted to Pass.			

Text Books:

1. T Sachidevi and T Sharadha (Thirumalai Sisters), Karnataka Sangeetha Dharpana- Vol. 1 (English), Shreenivaas Prakaashana, 2018.
2. B. Chaitanya Deva, An Introduction to Indian Music, Publications Division, Government of India, New Delhi.

Reference Books:

1. Lakshminarayana Subramaniam, Viji Subramanaim, "Classical Music of India: A Practical Guige, Tranqueber 2018.
2. Vamanrao H. Deshpande, Indian Musical Traditions: An Aesthetic Study of the Gharanas in Hindustani Music, Popular Prakashan, Mumbai**.**

ESTD : 2001
An Institute with a Difference

B. E. Common to all Programmes Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
Course Title	**ADDITIONAL MATHEMATICS-II**		
Course Code	**1BMADP410**	**CIE Marks**	**100**
(L:T:P) + SL	**(1:0:0)+(14 Hours/Sem)**	**SEE Marks**	**--**
Credits	**-**	**Exam Hours**	**--**

COURSE LEARNING OBJECTIVES:

The purpose of the course is to facilitate the learner to:

1. Apply the fundamental concepts of linear algebra to solve systems of linear equations and analyze mathematical models.
2. Solve second- and higher-order ordinary differential equations using appropriate analytical methods in engineering contexts.
3. Develop the knowledge and problem-solving skills required to evaluate multiple integrals, apply change of order of integration.
4. Apply numerical methods to obtain approximate solutions to mathematical problems arising in science and engineering.

Module-1
Linear Algebra: Introduction - rank of matrix by elementary row operations - Echelon form. Consistency of system of linear equations - Gauss elimination method. Eigen values and Eigen vectors of a square matrix. Problems.
Module-2
Numerical Methods: Finite differences. Interpolation/extrapolation using Newton's forward and backward difference formulae (Statements only)-problems. Solution of polynomial and transcendental equations – Newton-Raphson and Regula-Falsi methods (only formulae)- Illustrative examples. Numerical integration: Simpson's one third rule (without proof) Problems.
Module-3
Higher order Ordinary Differential Equations: Linear differential equations of second and higher order with constant coefficients, homogeneous and non-homogeneous equations, inverse differential operatorsand particular integral restricted to $R(x) = e^{ax}$, sin $ax$and cos ax for $f(D)y = R(x)$.
Module-4
Double integrals: Definition, evaluation of rectangular and general regions, Change of order of integration.
Module-5
Triple integrals: Definition and evaluation over simple regions, Applications of triple integrals to the evaluation of volumes.

COURSE OUTCOMES:

After successfully completing the course, the student will be able to:

CO1	Solve systems of linear equations using matrix algebra.
CO2	Apply the knowledge of numerical methods in modelling and solving engineering problems.

CO3	Apply analytical methods to solve higher order differential equations.
CO4	Solve double integrals, including change of order of integration.
CO5	Solve triple integrals, volumes of geometrical regions.

ASSESSMENT DETAILS (BOTH CIE AND SEE)

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE - IA Tests (A)	60	-	Sum of Two Internal Assessment Tests (MCQs) each conducted for 30 marks.
Continuous and Compre-hensive Evaluation (B)	40	-	Any two Assessment methods as per 24RNBE4.2, scaled to 40 marks.
Total CIE (A+B)	100	40	CIE Evaluation is done for a maximum of 100 marks.
The minimum marks to be secured in CIE is 40 (40%) out of allotted 100 marks to Pass.			

Textbooks:

1. B. S. Grewal, Higher Engineering Mathematics, 43rd Edition, Khanna Publishers, 2015.

Reference Books:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 10th Edition, John Wiley & Sons, 2015.
2. N. P. Bali and Manish Goyal, Engineering Mathematics, 7th Edition, Laxmi Pub-lishers, 2007.
3. Rohit Khurana, Engineering Mathematics, Vol. I, 1st Edition, Cengage Learning, 2015.

Additional Mathematics I and II:

1. These courses are prescribed for III and IV semesters respectively to lateral entry Diploma holders admitted to III semester of B.E./B.Tech., programs. They shall attend the classes during the respective semesters to complete all the formalities of the course and appear for the Continuous Internal Evaluation (CIE). In case any student fails to register for the said course/fails to secure the minimum 40 % of the prescribed CIE marks, he/she shall be deemed to have secured an F grade. In such a case, the student has to fulfill the course requirements during subsequent semesters to earn the qualifying CIE marks. These courses are slated for CIE only and have no SEE.
2. Additional Mathematics I and II shall not be considered for vertical progression as well as for the

calculation of SGPA and CGPA, but completion of the courses shall be mandatory for the award of degree.

3. Successful completion of the courses Additional Mathematics I and II shall be indicated as satisfactory in the grade card. Non-completion of the courses Additional Mathematics I and II shall be indicated as Unsatisfactory.

ESTD : 2001
An Institute with a Difference



VISION

Building RNSIT into a World Class Institution.

MISSION

To impart high quality education in Engineering, Technology and Management with a difference, enabling students to excel in their career by

- Attracting quality students and preparing them with a strong foundation in fundamentals so as to achieve distinctions in various walks of life leading to outstanding contributions.
- Imparting value based, need based, choice based and skill based professional education to the aspiring youth and carving them into disciplined, World class professionals with social responsibility.
- Promoting excellence in Teaching, Research and Consultancy that galvanizes academic consciousness among Faculty and Students.
- Exposing Students to emerging frontiers of knowledge in various domains and make them suitable for Industry, Entrepreneurship, Higher studies, and Research & Development.
- Providing freedom of action and choice for all the Stakeholders with better visibility.

www.rnsit.ac.in
follow us on:
RR Nagar, Dr. Vishnuvardhan Road, Bengaluru - 560 098
Ph: (080) 28611880, 28611881