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 COMPUTER SCIENCE & ENGINEERING

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 2025 – 26)

About RNSIT

RNS Institute of Technology (RNSIT), established in 2001, was founded by Dr. R. N. Shetty, a renowned industrialist and philanthropist, with the vision of imparting quality education in Engineering, Technology, and Management.

RNSIT is one of the premier engineering institutions in the country and is steadily advancing towards a global presence through excellence in academics and placements.

The institute features a lush green campus, state-of-the-art infrastructure, well-equipped laboratories, and a vibrant academic environment. With numerous initiatives, consistent VTU ranks, outstanding placements, cultural events, and sports activities, RNSIT has become a preferred destination for engineering aspirants. The institution hosts over 4000 students and more than 300 faculty members.

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 careers by:

- Attracting quality students and preparing them with a strong foundation in fundamentals to achieve excellence in various walks of life.
- Imparting value-based, need-based, choice-based, and skill-based professional education to shape disciplined, world-class professionals with social responsibility.
- Promoting excellence in teaching, research, and consultancy to enhance academic engagement among faculty and students.
- Exposing students to emerging frontiers of knowledge, preparing them for industry, entrepreneurship, higher studies, and research & development.
- Providing freedom of action and choice for all stakeholders with improved visibility and opportunities.

About Computer Science and Engineering

The Department of Computer Science & Engineering (CSE) at RNS Institute of Technology (RNSIT) is where academic excellence and innovation converge. For over 24 years, the department has proudly shaped future engineers by providing quality education, cutting-edge research opportunities, and a strong foundation for successful careers.

Accredited by the NBA, the department is committed to maintaining state-of-the-art laboratories and strong industry partnerships, ensuring a holistic learning experience. As an Autonomous Institution affiliated to VTU, the department offers an undergraduate program (established in 2001) with an intake of 720 students and houses a VTU-recognized R&D center (established in 2009).

Currently, 52 scholars are pursuing research, and 30 scholars have been awarded their degrees. The department has established Centers of Excellence in Machine Learning, FitterLife, and Agile Labs to advance in specialized fields. It has organized over 50 FDPs, 4 International Conferences, 3 National Conferences, and 400+ invited talks, benefiting both faculty and students.

The department strives to realize the vision of the institution by promoting excellence in teaching and providing a strong foundation in the fundamentals of computer science.

Vision

Preparing Better Computer Professionals for a Real World

Mission

- Imparting solid foundations and applied aspects in both Computer Science theory and programming practices.
- Providing training and encouraging R&D and consultancy services in frontier areas of Computer Science and Engineering with a global outlook.
- Fostering the highest ideals of ethics, values, and creating awareness of the role of computing in the global environment.
- Educating and preparing graduates who are highly sought after, productive, and well-respected for their work culture.
- Supporting and encouraging lifelong learning.

Program Educational Objectives (PEO)

PEO1: Graduates will build successful careers in computing by designing, analyzing, implementing, and evaluating reliable hardware/software systems to solve real-world problems.

PEO2: Graduates will practice engineering with ethics, effective communication, teamwork, leadership, and entrepreneurial skills while contributing responsibly to organizations and society.

PEO3: Graduates will pursue higher education, research, certifications, and continuous professional development to adapt to evolving technologies.

Program Outcomes (PO)

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.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis and interpretation of data to provide valid conclusions. (WK8)

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering and IT tools, including prediction and modelling, recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5 and WK7)

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national and international laws. (WK9)

PO8: Individual and Collaborative Teamwork: Function effectively as an individual, and as a member or leader in diverse and multidisciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, and make effective presentations considering cultural, language and learning differences.

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

PO11: Life-Long Learning: Recognize the need for and have the preparation and ability for (i) independent and life-long learning, (ii) adaptability to new and emerging technologies, and (iii) critical thinking in the broadest context of technological change. (WK8)

Program Specific Outcomes (PSO)

PSO1: Apply core computer science principles to design and develop hardware and software solutions with effective communication and teamwork.

PSO2: Develop innovative, SDG-aligned solutions using emerging technologies with research and lifelong learning.

B.E. in Computer Science and Engineering: Autonomous Scheme and Syllabus

3rd Semester

SL. NO	COURSE	COURSE CODE	COURSE TITLE	TEACHING DEPARTMENT AND QUESTION PAPER SETTING BOARD (PSB)	TEACHING & LEARNING SCHEME						EXAMINATION			
					CI		LI	TW & SL	# TH/S	DURATION IN HOURS	CIE MARKS	SEE MARKS	TOTAL MARKS	CREDITS
					L	T	P							
1	ASC	25BCSTH301	STATISTICS AND PROBABILITY DISTRIBUTIONS	TD/PSB: MATHEMATICS / CS ALLIED	45	30	0	45	120	3	50	50	100	**4**
2	IPCC	25BCSIC302	OBJECT ORIENTED PROGRAMMING USING JAVA	CSE	45	0	30	45	120	3	50	50	100	**4**
3	PCC	25BCSTH303	LOGIC DESIGN AND COMPUTER ORGANIZATION	CSE	45	30	0	45	120	3	50	50	100	**4**
4	PCC	25BCSTH304	DATA STRUCTURES AND APPLICATIONS	CSE	45	0	0	45	90	3	50	50	100	**3**
5	PCC	25BCSTH305	OPERATING SYSTEMS	CSE	45	0	0	45	90	3	50	50	100	**3**
6	PCCL	25BCSLB306	DATA STRUCTURES LABORATORY	CSE	0	0	30	00	30	3	50	50	100	**1**
7	AEC	25BCSAE307X	(BUCKET LIST B)	CSE	0	0	30	00	30	3	50	50	100	**1**
8	SDC	25BCBPR308	COMMUNITY PROJECT (PROJECT-BASED LEARNING) / SOCIETAL PROJECT	CSE	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	–	100	—	100	**PP**
TOTAL									**688**		**600**	**400**	**1000**	**21**

 # TH/S - TOTAL HOURS PER SEMESTER, TW & SL - TERM WORK & SELF LEARNING

B.E. in Computer Science and Engineering: Autonomous Scheme and Syllabus

3rd Semester

SL. NO	COURSE	COURSE CODE	COURSE TITLE	TEACHING DEPARTMENT AND QUESTION PAPER SETTING BOARD (PSB)	TEACHING & LEARNING SCHEME						EXAMINATION			
					CI		LI	TW & SL	# TH/S	DURATION IN HOURS	CIE MARKS	SEE MARKS	TOTAL MARKS	CREDITS
					L	T	P							
11	SKLAB	25BCSSL00X	SKILL LAB(BUCKET LIST SDL)	CSE	0	0	30	0	30	3	50	50	100	**1**

RNS INSTITUTE OF TECHNOLOGY ESTD:2001 An Institute with a Difference

AEC — 25BCSAE307X — BUCKET LIST B	
25BCSAE307A	PROJECT MANAGEMENT WITH GIT
25BCSAE307B	NETLIFY
25BCSAE307C	CODEPEN
25BCSAE307D	REPLIT
25BCSAE307E	OPEN REFINE
25BCSAE307F	DOCKER PLAYGROUND
25BCSAE307G	KAGGLE
25BCSAE307H	BURP SUITE COMMUNITY EDITION
25BCSAE307I	POSTMAN

SKILL LAB COURSE — 25BCSSL00X — BUCKET LIST SDL	
25BCSSL001	ROBOTICS PROGRAMMING AND AUTONOMOUS SYSTEMS
25BCSSL002	IOT WITH ESP32 BOARD
25BCSSL003	DRONE PROGRAMMING AND AUTONOMOUS SYSTEMS
25BCSSL004	IMPACT INCUBATOR: FIELD TO FUNCTION
25BCSSL005	AWS ACADEMY CLOUD FOUNDATIONS
25BCSSL006	SALESFORCE - AI AGENT FORCE
25BCSSL007	SALESFORCE - ADMINISTRATOR

B.E. in Computer Science and Engineering: Autonomous Scheme and Syllabus
4th Semester

SL. NO	COURSE	COURSE CODE	COURSE TITLE	TEACHING DEPARTMENT AND QUESTION PAPER SETTING BOARD (PSB)	TEACHING & LEARNING SCHEME						EXAMINATION			
					CI		LI	TW & SL	# TH/S	DURATION IN HOURS	CIE MARKS	SEE MARKS	TOTAL MARKS	CREDITS
					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	45	0	30	45	120	3	50	50	100	**4**
3	PCC	25BCSTH403	DESIGN AND ANALYSIS OF ALGORITHMS	CSE	45	30	0	45	120	3	50	50	100	**4**
4	PCC	25BCSTH404	ARTIFICIAL INTELLIGENCE & MACHINE LEARNING	CSE	45	30	0	45	120	3	50	50	100	**4**
5	PCCL	25BCSLB405	DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY	CSE	0	0	30	00	30	3	50	50	100	**1**
6	AEC	25BCSAE406X	(BUCKET LIST C)	CSE	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	RESPECTIVE ENGINEERING DEPT.	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	Campus: PHYSICAL EDUCATION DIRECTOR										
		25BYOGA409	YOGA	Campus: YOGA TEACHER										
		25BMUSK409	MUSIC	Campus: MUSIC TEACHER										
TOTAL									**660**		**500**	**400**	**900**	**21**
10	NCMC	25BMADP410	ADDITIONAL MATHEMATICS-II	TD - MATHS DEPT	14	0	0	14	28	–	100	—	100	**PP**
TOTAL									**686**		**600**	**400**	**1000**	**21**

 # TH/S - TOTAL HOURS PER SEMESTER, TW & SL - TERM WORK & SELF LEARNING ** the course 1BXXL406x - Ability Enhancement Course can be offered either as a single compulsory course. Alternatively, the Ability Enhancement Course Laboratory course can be offered with multiple elective options under the course codes 1BXXL406x (where x = A, B, C, D).

B.E. in Computer Science and Engineering: Autonomous Scheme and Syllabus

4th Semester

SL. NO	COURSE	COURSE CODE	COURSE TITLE	TEACHING DEPARTMENT AND QUESTION PAPER SETTING BOARD (PSB)	TEACHING & LEARNING SCHEME						EXAMINATION			
					CI		LI	TW & SL	# TH/S	DURATION IN HOURS	CIE MARKS	SEE MARKS	TOTAL MARKS	CREDITS
					L	T	P							
11	SKLAB	25BCSSL00X	SKILL LAB(BUCKET LIST SDL)	CSE	0	0	30	0	30	3	50	50	100	**1**

RNS INSTITUTE OF TECHNOLOGY
ESTD:2001
An Institute with a Difference

AEC - 25BCSAE406X - BUCKET LIST C	
25BCSAE406A	INTRODUCTION TO UNIX
25BCSAE406B	MOBILE APP DEVELOPMENT
25BCSAE406C	ADVANCED JAVA
25BCSAE406D	NMAP
25BCSAE406E	GOOGLE COLAB
25BCSAE406F	ANACONDA NAVIGATOR
25BCSAE406G	TRYHACKME
25BCSAE406H	WIRESHARK
25BCSAE406I	TABLEAU PUBLIC
25BCSAE406J	ORANGE DATA MINING

TH/S- Total Hours per Semester, TW & SL- Term Work & Self Learning. ** The course 1BXXL406x – Ability Enhancement Course Laboratory can be offered either as a single compulsory course. Alternatively, the Ability Enhancement Course Laboratory course can be offered with multiple elective options under the course codes 1BXXL406x (where x = A, B, C, D).

Contents

SEMESTER – III

1 STATISTICS AND PROBABILITY DISTRIBUTIONS 3

2 OBJECT ORIENTED PROGRAMMING USING JAVA 7

3 LOGIC DESIGN AND COMPUTER ORGANIZATION 12

4 DATA STRUCTURES AND APPLICATIONS 17

5 OPERATING SYSTEMS 21

6 DATA STRUCTURES LABORATORY 25

7 PROJECT MANAGEMENT WITH GIT 29

8 NETLIFY 32

9 CODEPEN 38

10 REPLIT 42

11 OPEN REFINE 47

12 DOCKER PLAYGROUND 52

13 KAGGLE 56

14 BURP SUITE COMMUNITY EDITION 59

15 POSTMAN 65

16 COMMUNITY PROJECT / SOCIETAL PROJECT (PROJECT-BASED LEARNING) 69

17 PHYSICAL EDUCATION 82

18 YOGA 85

19 MUSIC 88

20 ADDITIONAL MATHEMATICS-I 91

SEMESTER – IV

21 DISCRETE MATHEMATICS AND GRAPH THEORY 93
22 DATABASE MANAGEMENT SYSTEMS 96
23 DESIGN AND ANALYSIS OF ALGORITHMS 103
24 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING 106
25 DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY 109
26 INTRODUCTION TO UNIX 113
27 MOBILE APP DEVELOPMENT 119
28 ADVANCED JAVA 127
29 NMAP 131
30 GOOGLE COLAB 134
31 ANACONDA NAVIGATOR 138
32 TRYHACKME 142
33 WIRESHARK 146
34 TABLEAU PUBLIC 150
35 ORANGE DATA MINING 157
36 BIOLOGY FOR ENGINEERS 161
37 NATIONAL SERVICE SCHEME 164
38 PHYSICAL EDUCATION 166
39 YOGA 169
40 MUSIC 172
41 ADDITIONAL MATHEMATICS-II 175

III SEMESTER SUBJECTS

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	STATISTICS AND PROBABILITY DISTRIBUTIONS (CSE ALLIED BRANCHES)		
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. Analyse 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 by the method of least squares—fitting 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.

Textbook 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.

Textbook 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.

Textbook 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). **Textbook 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. **Textbook 2: 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 chainsand apply stochastic modelling methods for decision-making and forecasting.

ASSESSMENT AND EVALUATION REGULATIONS OF ASC

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 Theory (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. 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:

1. `http://academicearth.org/`
2. VTU e-Shikshana Program
3. VTU EDUSAT Program

4. https://nptel.ac.in/courses/111105160
5. https://nptel.ac.in/courses/127106019
6. https://sl1nk.com/rb8fcyk
7. https://sl1nk.com/ooolb6f

Activity-Based Learning:

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

B.E. (Computer Science and Engineering)
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)+(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. 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, `this` keyword, 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: 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.

Interfaces: Defining interfaces, Implementing interfaces, Extending interfaces.

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`, User-defined exception subclasses. **Multithreaded Programming:** Main thread, Creating threads, Multiple threads, Using `isAlive()` and `join()`, Synchronization. **Textbook 1 : Chapter 10 (10.1, 10.2, 10.4, 10.5, 10.7 - 10.9, 10.11), Chapter 11 (11.2 - 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`, `StringBuilder`. **Textbook 1 : Chapter 18 (18.1 - 18.10, 18.12, 18.13)**	
Module-5	**9 Hours**
The Collections Framework: Collections overview, The `Collection` interface, List and Set interfaces, Collection classes: `ArrayList`, `LinkedList`, `HashSet`, `LinkedHashSet`, `TreeSet`, Accessing a collection via an `Iterator`, The `RandomAccess` interface, 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 COMPONENTS 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.

Program 12: Managing Unique Student Records Using Java Collections (HashSet)
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 with appropriate data members and methods to address specified scenarios.
CO2	Apply the concept of inheritance to model and solve real-world programming problems.
CO3	Construct Java programs using multithreading and exception handling techniques to manage concurrency and runtime exceptions.
CO4	Demonstrate the use of string handling methods to manipulate and process textual data in Java.
CO5	Implement the Java Collections Framework using Collection interfaces, Lists, Sets, Maps, Iterators, Comparators and user-defined objects for efficient data management.

ASSESSMENT AND EVALUATION REGULATIONS OF IPCC

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 method 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, 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:

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`

Self Study Topics:

1. SOLID Principles (GeeksforGeeks)

B.E. (Computer Science and Engineering)
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

After completing this course, the learner will be able 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.

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 structures, Performance. **Machine Instructions and Programs:** Memory Locations 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 operational 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 AND EVALUATION REGULATIONS OF PCC

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 the two Internal Assessment Tests shall be scaled down to 20 marks.
Continuous and Comprehensive Evaluation (B)	20	-	Any one assessment method as per 24RNBE4.2, scaled to 20 marks.
Total CIE Theory (A+B)	50	20	Marks of tests and assignments awarded for a maximum of 30.
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. Vranesicand 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

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.

Activity Based Learning:

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

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 and Algorithms: Basic terminology, classification of data structures, operations on data structures.

Basic Concepts: Pointers and dynamic memory allocation.

Arrays and Structures: Arrays, dynamically allocated arrays, structures and unions, polynomials, sparse matrices, representation of multidimensional arrays, strings.

Searching and Sorting: Insertion sort, radix sort.

Textbook 2: Chapter 2 (2.1 - 2.3), Chapter 14 (14.8, 14.12)

Textbook 1: Chapter 1 (1.2), Chapter 2 (2.1 - 2.7)

Module-2	9 Hours

Stacks and Queues: Stacks, stacks using dynamic arrays, queues, circular queues using dynamic arrays, evaluation of expressions, multiple stacks and queues.

Stacks: Applications of stacks.

Textbook 1: Chapter 3 (3.1 - 3.4, 3.6, 3.7)

Textbook 2: Chapter 7 (7.7.4)

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**
Graphs: Graph abstract data types, elementary graph operations. **Hashing:** Introduction, static hashing, dynamic hashing. **Multiway Search Trees:** B-trees, B+ trees. **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	Identify fundamental data structures and describe operations using arrays, pointers, and strings.
CO2	Illustrate how stacks and queues evaluate expressions and solve linear data structure problems.
CO3	Construct memory-efficient programs using singly and doubly linked lists.
CO4	Examine tree-based structures to model hierarchical relationships.
CO5	Compare graphs, balanced trees, and hashing techniques to address real-world computational problems.

ASSESSMENT AND EVALUATION REGULATIONS OF PCC

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 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, McGraw 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

RNS
INSTITUTE OF TECHNOLOGY
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	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 apply concepts of processes, threads, and CPU scheduling, along with inter-process 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 operating system resources.

Module-1	9 Hours

Introduction to Operating Systems: What operating systems do, Computer system organization, Computer system architecture, Operating system structure, Operating system operations.

System Structures: 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
Synchronization: Background, The critical section problem, Peterson's Solution, Synchronization hardware, Semaphores, Classic 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 Strategies: 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: 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:** Overview of mass storage structure, Disk structure, Disk attachment, Disk scheduling. **System 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 AND EVALUATION REGULATIONS OF PCC

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 one assessment method as per 24RNBE4.2, scaled to 10 marks.
Total CIE Theory (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 and H. Bos, Modern Operating Systems, 4th Edition, Prentice-Hall of India, New Delhi, 2014.

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)	(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. Analyze 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, Semesterand 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, Salaryand 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.

e. 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, Preorderand 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:
1. Store vehicle registration numbers in the hash table using linear probing and quadratic probing for collision handling. 2. 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 AND EVALUATION REGULATIONS OF PCCL

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.			

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

Course Title	PROJECT MANAGEMENT WITH GIT		
Course Code	25BCSAE307A	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. 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 branchesand 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, dateand 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 AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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. Version Control with Git, 3rd Edition, by Prem Kumar Ponuthorai, Jon Loeliger. Released October 2022. Publisher(s): O'Reilly Media, Inc.
2. Pro Git book, written by Scott Chacon and Ben Straub, published by Apress. https://git-scm.com/book/en/v2
3. https://infyspringboard.onwingspan.com/.../overview
4. https://infyspringboard.onwingspan.com/.../overview

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

Course Title	NETLIFY		
Course Code	25BCSAE307B	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. Understand the Netlify platform its core concepts, pricing plansand the modern JAM-stack architecture it supports.
2. Deploy static & dynamic websites on Netlify using drag-and-drop, Git-based Continuous Deployment (CD), & the Netlify CLI.
3. Leverage Netlify's serverless and edge computing primitives including Serverless Functions, Edge Functions, Blobsand Netlify DB to build full-stack applications without managing servers.
4. Configure and manage builds, custom domains, environment variables, redirectsand forms using Netlify's platform features and the netlify.toml configuration file.
5. Explore Netlify's AI capabilities including the AI Gateway and Agent Runners to build and deploy AI-powered web applications with minimal configuration.

Program 1 - Deploy a Static Website on Netlify using Drag-and-Drop and Git-based Continuous Deployment.

Example 1 – Drag and Drop: Build a simple personal portfolio using HTML, CSSand JavaScript. Drag and drop the project folder into the Netlify Deploy Zone. Note the auto-generated *.netlify.app URL. Share the live URL with peers.
Example 2 – Git-based CD: Push the same project to a GitHub repository. Connect the repository to Netlify via Add new site ¿ Import an existing project. Observe how every git push triggers an automatic build and deploy—enabling Continuous Deployment with zero extra effort.

Program 2 - Demonstrate the Netlify CLI for Local Development and Manual Deploys.

Install the Netlify CLI globally using: npm install -g netlify-cli. Use netlify login to authenticate with your Netlify account. Run netlify dev to spin up a local development server that mirrors the production Netlify environment, including environment variables and function routing.

Example 1: Run netlify deploy –dir=dist to create a draft (preview) deploy of a Vite project. Observe the unique deploy preview URL generated.
Example 2: Run netlify deploy –prod to push changes to the production URL. Demonstrate how the CLI allows rollback by listing deploys with netlify sites:list.

Program 3 - Configure Builds using netlify.toml — Build Settings, Redirectsand Environment Variables.

Demonstrate how to create a netlify.toml file in the project root to configure build commands, publish directoriesand redirect rules — without touching the Netlify dashboard.
Example 1: Write a netlify.toml with [build] settings specifying command = 'npm run build' and publish = 'dist'. Show how this auto-configures Netlify's build pipeline.
Example 2: Add a [[redirects]] block to redirect /old-page to /new-page with a 301 status. Deploy and verify the redirect works on the live site.
Example 3: Set environment variables via the Netlify UI (Project configuration Environment variables) and access them during the build using process.env.VARIABLE_NAME. Demonstrate scoping variables to Build, Functions, or Runtime contexts.

Program 4 - Demonstrate Deploy Previews, Branch Deploysand Rollbacks.

One of Netlify's most powerful team collaboration features is Deploy Previews — every pull request or branch gets its own unique live preview URL before changes are merged to production.
Example 1 – Deploy Preview: Create a new feature branch in Git. Push a change (e.g., update the homepage heading). Open a Pull Request on GitHub. Show how Netlify automatically posts a Deploy Preview URL in the PR comments — allowing code review with a real live site.
Example 2 – Branch Deploy: Configure a staging branch in Netlify dashboard (Site configuration Branch deploys). Push changes to the staging branch. Show the staging.projectname.netlify.app URL live alongside the main production URL.
Example 3 – Rollback: From the Netlify dashboard, navigate to Deploys. Click on any previous successful deploy and select 'Publish deploy' to instantly revert the live site — demonstrating zero-downtime rollback.

Program 5 - Build and Deploy a Serverless Function on Netlify.

Netlify Functions allow you to run backend server-side code without managing a dedicated server. Functions are version-controlled, automatically deployed alongside your siteand accessible via built-in API routes.

Example 1 – Hello World Function: Create a file netlify/functions/hello.mjs. Write a default export handler function that returns a JSON Response with message: 'Hello from Netlify Functions!'. Run locally using netlify dev and visit /.netlify/functions/hello. Deploy to production and test the live endpoint.

Example 2 – Contact Form Handler: Build a function that receives a POST request from an HTML form, extracts the submitted data (name, email, message)and returns a success JSON response. Demonstrate how this eliminates the need for a separate Express/Node.js server.

Program 6 - Demonstrate Netlify Forms – Serverless Form Handling without a Backend.

Netlify Forms allows HTML forms to be processed and stored by Netlify without writing any backend code or API calls. Just add the `netlify` attribute to a `<form>` tag and Netlify handles submissions automatically.
Example 1 – Basic Contact Form: Create an HTML page with a `<form netlify>` containing fields for name, emailand message. Deploy the site and verify submissions from the Netlify dashboard.
Example 2 – Email Notifications: Configure email notifications under Site Configuration > Forms > Form Notifications and verify email alerts on form submission.
Example 3 – Spam Filtering: Enable the built-in honeypot field and demonstrate automatic spam filtering.

Program 7 - Configure Custom Domains, HTTPSand Netlify DNS.

Configure custom domains, SSL certificatesand Netlify DNS services for secure website hosting.
Example 1 – Add a Custom Domain: Add a custom domain from Site Configuration > Domain Management and update DNS records at the registrar. Verify automatic SSL certificate provisioning.
Example 2 – Netlify DNS: Configure Netlify DNS by updating nameservers and manage DNS records from the Netlify dashboard.

Example 3 – Branch-specific Subdomains: Configure automatic deploy subdomains such as `staging.myportfolio.com` for branch deployments.

Program 8 - Demonstrate Edge Functions for Personalized and Low-Latency Experiences.

Edge Functions run code at the CDN edge — at the network node geographically closest to each user — making them ideal for low-latency personalization, authenticationand content modification. They use a Deno-based runtime and are written in TypeScript or JavaScript.
Example 1 – Geolocation-based Greeting: Create an edge function that reads the context.geo object (which contains the user's country, cityand timezone) and returns a personalized greeting: 'Welcome from Bengaluru, India!' Test locally using netlify dev and verify the geo data.

Example 2 – Request Header Modification: Write an edge function that intercepts every request, adds a custom response header (e.g., X-Edge-Powered: true)and calls context.next() to pass the request to the origin. Deploy and verify the custom header appears in browser DevTools > Network > Response Headers.
Example 3 – A/B Testing: Demonstrate splitting traffic between two versions of a landing page using an edge function that randomly sets a cookie and serves different content — without a full page reload or server round-trip.

Program 9 - Use Netlify Blobs (Key-Value Storage) and Netlify DB for Data Persistence.

Netlify provides two built-in data storage primitives: Netlify Blobs (a simple key-value/object store) and Netlify DB (a production-grade SQL database powered by Neon Postgres), both provisioned instantly with zero external configuration.
Example 1 – Netlify Blobs:In a Serverless Function, use the netlify/blobs package:import getStore from 'netlify/blobs';.Create a store called 'user-uploads'.Use await store.set(key, fileData) to save a file upload and await store.get(key) to retrieve it. Show the entry in the Netlify dashboard under Data & Storage Blobs.
Example 2 – Netlify DB: Use the Netlify dashboard to provision a Netlify DB instance (Data & Storage Database). Connect to it using a standard PostgreSQL client or Drizzle ORM. Create a simple todos table, insert a row from a Serverless Functionand query it back, demonstrating full-stack data persistence with zero server management.

Program 10 - Demonstrate Netlify Image CDN for On-Demand Image Optimization.

Netlify Image CDN transforms images on-demand through resizing, format conversionand quality adjustment without impacting build times. Transformed images are cached at the CDN edge for fast delivery to users globally.
Example 1 – Basic Transform:Given an image at /images/photo.jpg, request a resized version using the URL /.netlify/images?url=/images/photo.jpg&w=400&h=300&fit=cover. Compare the original and optimized file sizes using browser DevTools.
Example 2 – Format Conversion: Request the same image in WebP format using /.netlify/images?url=/images/photo.jpg&fm=webp. Demonstrate how Netlify automatically negotiates the image format using the Accept header, serving WebP to supported browsers and JPEG to others without any code changes.
Example 3 – Remote Image Optimization: Configure allowed remote domains in netlify.toml under [images] remote_images. Optimize a remote image URL (for example, a product image hosted on a CDN) through Netlify Image CDN to reduce bandwidth usage and improve Core Web Vitals scores.

COURSE OUTCOMES

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

CO1	Deploy static and dynamic websites to Netlify using drag-and-drop, Git-based Continuous Deploymentand the Netlify CLI choosing the most appropriate method for a given use case.
CO2	Configure build pipelines, environment variables, custom domains, HTTPS, redirectsand forms using netlify.toml and the Netlify platform dashboard.
CO3	Build and deploy serverless backend logic using Netlify Serverless Functions and Edge Functions eliminating the need for a dedicated backend server.
CO4	Integrate persistent data storage into web applications using Netlify Blobs and Netlify DB and optimize media delivery using Netlify Image CDN.
CO5	Construct and deploy AI-powered web applications using Netlify AI Gateway and Agent Runners, leveraging platform-managed credentials for OpenAI, Anthropicand Google Gemini.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	**Maximum Marks**	**Minimum Passing Marks**	**Evaluation Details**
CIE – 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 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://docs.netlify.com/
2. https://docs.netlify.com/start/overview/
3. https://l1nq.com/hbh7uau
4. https://docs.netlify.com/build/configure-builds/
5. https://docs.netlify.com/build/environment-variables/overview/
6. https://docs.netlify.com/build/functions/overview/
7. https://docs.netlify.com/build/edge-functions/overview/
8. https://docs.netlify.com/manage/forms/setup/
9. https://docs.netlify.com/manage/domains/get-started-with-domains/
10. https://docs.netlify.com/deploy/compare-preview-options/
11. https://docs.netlify.com/build/image-cdn/overview/
12. https://docs.netlify.com/build/data-and-storage/netlify-blobs/
13. https://docs.netlify.com/build/data-and-storage/netlify-db/
14. https://docs.netlify.com/build/ai-gateway/overview/
15. https://docs.netlify.com/build/ai-gateway/quickstart-for-ai-gateway/
16. https://docs.netlify.com/build/build-with-ai/agent-runners/overview/
17. https://docs.netlify.com/build/build-with-ai/netlify-mcp-server/
18. https://docs.netlify.com/manage/monitoring/overview/
19. https://docs.netlify.com/build/frameworks/overview/
20. https://developers.netlify.com/videos/

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

Course Title	CODEPEN		
Course Code	25BCSAE307C	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. Understand the fundamentals of HTML, CSS and JavaScript for web development.
2. Design and develop responsive and visually appealing user interfaces.
3. Implement client-side scripting for dynamic and interactive web pages.
4. Analyze the design and implementation requirements of real time web applications and reusable UI Components.
5. Apply concepts of validation, user interaction and modern web design practices.

Program 1 - Responsive Personal Profile Webpage Using HTML and CSS

A front-end developer is assigned to create a responsive personal profile webpage using HTML and CSS. The webpage should display the user's name and profile image at the top. It must include a structured list of skills. A contact section should be provided for user details. The layout should use cards to organize content neatly. Hover effects should be added to improve user interaction. The overall design should be clean, structured and responsive across devices.

Program 2 - Automatic Image Slider Using HTML, CSSand JavaScript

A web developer is assigned to create an image slider webpage using HTML, CSS and JavaScript. The webpage should display one image at a time for better focus. It must include Next and Previous buttons for manual navigation. Images should automatically change every 3 seconds. Basic CSS styling should be applied to improve visual appearance. The interface should update dynamically using JavaScript. The overall design should be simple, interactive and user-friendly.

Program 3 - Student Login Form with Validation Using HTML, CSSand JavaScript

A web developer is assigned to create a student login form using HTML, CSS and JavaScript. The form should include username and password fields for user input. It must provide a submit button with proper styling. Validation should be implemented for empty fields and password length. A password visibility toggle should be included for better usability. Error and success messages must be displayed based on user input. The overall design should be user-friendly and secure.

Program 4 - Click Counter Application with Custom Step Values
A web developer is assigned to create a click counter application using HTML, CSS and JavaScript. The application should allow incrementing and decrementing counter values. It must include a reset button to restore the count to default. Users should be able to provide custom step values for increment and decrement. The system must prevent negative values from being displayed. The user interface should update dynamically using DOM manipulation. The overall design should be interactive and easy to use.
Program 5 - Random Background Colour Changer with HEX Display
A web developer is assigned to create a background colour changer webpage using HTML, CSS and JavaScript. The application should change the background colour on button click. It must generate random HEX colour values dynamically. The current colour code should be displayed and allow users to copy it. Smooth CSS transitions should be applied for better visual effect. The user interface must update dynamically using DOM manipulation. The overall design should be interactive and visually appealing.
Program 6 - To-Do List Application with Local Storage
A web developer is assigned to create a to-do list application using HTML, CSS and JavaScript. The application should allow users to add and display tasks dynamically. It must enable marking tasks as completed for better tracking. Users should have options to edit and delete tasks as needed. All tasks should be stored in local storage for persistence. The user interface must update dynamically using DOM manipulation. The overall design should be interactive and user-friendly.
Program 7 - Dynamic Random Quote Generator with Auto-Refresh
A web developer is assigned to create a dynamic quote generator webpage using HTML, CSS and JavaScript. The application should display one quote at a time on the screen. It must provide a button to generate a random quote based on user interaction. The quotes should also change automatically at fixed time intervals. Each quote must display the author's name clearly. Proper CSS styling should be applied to enhance the visual appearance. The overall design should be interactive and user-friendly.
Program 8 - Advanced Calculator with Calculation History
A web developer is assigned to create an advanced calculator application using HTML, CSS and JavaScript. The application should accept two numbers as input from the user. It must perform arithmetic operations such as addition, subtraction, multiplication, division and modulus. The results should be displayed dynamically on the webpage. The system should also maintain a history of all calculations performed. Invalid inputs must be handled properly using JavaScript validation. The overall design should be interactive, accurate and user-friendly.
Program 9 - Feedback Form with Star Rating System
A web developer is assigned to create a feedback form with rating system using HTML, CSS and JavaScript. The application should collect user name and feedback message along with a star rating (1–5). Submitted feedback must be displayed dynamically on the

webpage. Input fields should be cleared after submission. Proper CSS styling should be applied for better user interface. The design must be interactive and user-friendly.
Program 10 - Interactive Online Quiz Platform with Timer and Leaderboard
A web developer is assigned to create an interactive online quiz platform using HTML, CSS and JavaScript. The application should support multiple categories and difficulty levels, display one question at a time with a countdown timer, automatically submit answers when time expires and calculate the score, show performance analysis with correct and incorrect answers, maintain a leaderboard using local storage and include a progress bar to track quiz completion. The design must be interactive, responsive and user-friendly.

COURSE OUTCOMES

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

CO1	Develop responsive web pages using HTML and CSS with proper structure and styling.
CO2	Implement interactive features using JavaScript and DOM manipulation.
CO3	Design and build web applications such as forms, counters, to-do lists and calculators.
CO4	Apply input validation, error handling and data storage techniques in web applications.
CO5	Create advanced UI components and user-friendly interfaces using modern design practices.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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. Jon Duckett, HTML and CSS: Design and Build Websites, 1st Edition, John Wiley & Sons, Inc., 2011.
2. Jon Duckett, JavaScript and JQuery: Interactive Front-End Web Development, 1st Edition, John Wiley & Sons, Inc., 2014.

Web Links:

1. https://www.coursera.org/learn/introduction-to-front-end-development
2. https://www.coursera.org/professional-certificates/meta-front-end-developer

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

Course Title	REPLIT		
Course Code	25BCSAE307D	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. Use cloud-based development platforms like Replit to create, execute and test programs without requiring local setup.
2. Apply AI-assisted tools in Replit for generating, modifying and improving program code.
3. Develop basic web applications using HTML, CSS and JavaScript with focus on structure, responsiveness and user interaction.
4. Utilize Replit features such as collaboration, debugging, testing and sharing for effective application development.

Program 1 - Exploring Replit Workspace

Demonstrate the usage of Replit (cloud-based development platform) which enables coding without installing any software, providing features such as editor, console and live preview.

a. Create a new project using “Create something new” option

b. Write a simple program to display student details

c. Execute the program using the Run button

d. Identify and explain editor, consoleand preview sections

Program 2 - AI-Based Code Generation using Replit AI

Demonstrate the use of Replit AI for generating code automatically from natural language prompts, reducing manual coding effort.

a. Give a prompt to generate a simple calculator program

b. Execute the generated code

c. Verify correctness of output

d. Modify the prompt and observe changes in code

Program 3 - Modifying AI-Generated Code
Demonstrate how AI-generated code can be customized and improved for real-world usage. a. Take an AI-generated program b. Add user input functionality c. Improve output formatting d. Test the program with different inputs
Program 4 - Personal Portfolio Webpage Development
Demonstrate the development of a basic personal portfolio using Replit's web environment. a. Create an HTML page with personal details b. Add headings, paragraphs and basic styling c. Use Replit preview to view the webpage d. Modify design using simple CSS
Program 5 - Login Interface Design
Demonstrate the design of a simple user interface for authentication systems. a. Create username and password input fields b. Add a login button c. Design layout using CSS (alignment, spacing) d. Improve readability and structure
Program 6 - Login Interface Design
To-Do List Application Demonstrate basic application development for managing tasks using Replit's web environment. a. Create input field for tasks b. Add button to insert tasks c. Display tasks dynamically using JavaScript d. (Optional) Add delete or clear functionality
Program 7 - Responsive Web Design using Replit
Demonstrate how web applications can be adapted for different screen sizes using Replit's live preview feature. a. Modify webpage layout using basic CSS b. Adjust width, alignment and text size c. Use media queries for simple responsiveness

d. Test layout using Replit preview resize e. Ensure mobile-friendly display
Program 8 - Project Presentation Webpage using Replit
Demonstrate how structured project information can be presented using web technologies in Replit. a. Organize content into sections (title, objective, features, conclusion) b. Use headings, paragraphs and lists c. Apply simple CSS styling for clarity d. Use Replit preview to view and refine layout e. Ensure clear and readable presentation
Program 9 - Webpage Enhancement using Animation in Replit
Demonstrate how user interaction can be improved by applying simple CSS animations using Replit's web development environment. a. Add hover effects using CSS b. Apply transitions or basic animations c. Modify colors, size, or position on interaction d. Test animation behavior using Replit preview e. Ensure smooth and responsive user experience
Program 10 - Collaboration and Project Sharing
Demonstrate collaborative development using Replit's sharing features. a. Share project link with peers b. Make changes collaboratively c. View updates in real-time d. Present the working application
Program 11 - AI-Assisted Code Debugging and Improvement using Replit
Demonstrate the use of Replit AI to debug, analyze and improve existing code within the Replit workspace, highlighting real-time assistance during development. a. Write or use an existing program with logical or syntax errors b. Use Replit AI to identify and suggest corrections c. Apply the suggested fixes within the editor d. Execute the program and verify correct output in the console e. Compare the original and improved code for clarity and efficiency

Program 12 - Application Testing and Sharing using Replit
Demonstrate how applications developed in Replit can be executed, tested and shared using its cloud-based features, ensuring correctness and accessibility. a. Write a simple program (e.g., calculator or student information display) b. Execute the program using the Run feature c. Test the program with different inputs and verify output d. Identify and correct minor errors if present e. Generate and share the project link for access and demonstration

COURSE OUTCOMES

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

CO1	Understand and demonstrate the use of Replit for program development and execution in a cloud environment.
CO2	Apply AI-assisted techniques to generate, modifyand debug programs effectively.
CO3	Design and develop basic web applications with appropriate structure, styling and interactivity.
CO4	Analyze program behavior and improve functionality through testing and debugging.
CO5	Utilize Replit features to collaborate, share and present working applications effectively.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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.			

Reference Links:

1. https://www.geeksforgeeks.org/.../introduction-to-replit
2. https://www.youtube.com/watch?v=St95nPOwsa8...
3. https://docs.replit.com/getting-started/intro-replit

RNS 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	OPEN REFINE		
Course Code	25BCSAE307E	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. Use Open Refine for Data Cleaning and Transformation.
2. Understand Data Exploration Techniques.
3. Apply GREL for Data Manipulation.
4. Perform Data Reconciliation.
5. Develop Reproducible Data Wrangling Workflows.

Program 1 - Installing OpenRefine and Exploring Dataset Using Facets

i) Install OpenRefine and demonstrate how to import a dataset (CSV/Excel/JSON). Explore the dataset by identifying rows and columns.

ii))Using OpenRefine, apply different types of facets (Text, Numeric, Timeline) to explore and filter the dataset.

Program 2 - Dataset Exploration and Email Domain Analysis Using Facets

i)Explore the dataset using facets and filtersand identify inconsistencies or errors present in the data.

ii)Email Domain Analysis (Faceting)

A college collects student email IDs for communication purposes, but the administration wants to analyze which email service providers are most commonly used. The dataset contains various email IDs such as Gmail, Yahoo and Outlook. Due to the large volume of data, manual analysis is difficult. Using OpenRefine, the domain part of each email is extracted using transformation functions. Then, a text facet is applied to group similar domains together. This allows the system to count the frequency of each domain efficiently. The result helps the college understand communication trends. It also assists in optimizing official communication channels.

Program 3 - Data Cleaning and Date Standardization Using OpenRefine

i)Clean the dataset by removing duplicate rows, handling missing values and standardizing text formats case conversion, trimming spaces).

ii)Date Standardization

A company gathers data from multiple sources where dates are stored in different formats such as DD/MM/YYYY, YYYY-MM-DD and text formats. This inconsistency creates problems during data analysis and reporting. Using OpenRefine, all date values are converted into a standard format. Transformation functions are applied to parse and reformat dates. This ensures uniformity across the dataset. The cleaned data becomes easier to sort and analyze. It also avoids errors in time-based calculations. Standardization improves overall data quality and reliability.

Program 4 - Clustering Techniques and Removing Extra Spaces

i) Apply clustering techniques to detect and merge similar or inconsistent data entries.

ii) Removing Extra Spaces

In a student database, names are entered with inconsistent spacing such as leading and trailing spaces. This results in duplicate entries and incorrect grouping during analysis. OpenRefine is used to clean the data by applying trimming operations. The tool removes unnecessary spaces from each record. After cleaning, all names appear in a consistent format. This improves accuracy in searching and filtering operations. It also helps in eliminating duplicate records. Overall, the dataset becomes cleaner and more reliable.

Program 5 - Splitting and Merging Columns Using Delimiters

i)Demonstrate how to split and merge columns using appropriate delimiters.

ii)Name Splitting

An organization stores employee names in a single column as full names. However, for processing and reporting, it requires first and last names separately. OpenRefine is used to split the column based on a space delimiter. This operation creates multiple columns for structured data storage. Each part of the name is separated accurately. This makes it easier to sort and filter records. It also supports better database management. Structured data improves system efficiency and usability.

Program 6 - GREL Expressions for Data Transformation and Marks Filtering

i)Use GREL expressions to transform data, such as converting text case, trimming values and modifying entries.

ii)Student Marks Filtering

A teacher wants to analyze student performance and identify those who have passed the exam. The dataset contains marks of all students. Instead of checking each record manually, OpenRefine is used to apply numeric filtering. A numeric facet is created to filter marks greater than a passing threshold. This quickly separates passed and failed students. The filtered data can be used for further analysis. It helps in preparing result summaries efficiently. The process saves time and reduces manual effort.

Program 7 - Advanced GREL Functions and Name Merging

i)Apply advanced GREL functions to perform operations like splitting values, calculating length and conditional transformations.

ii)Name Merging

In a database system, first names and last names are stored in separate columns. However, for generating reports or displaying user details, full names are required. OpenRefine is used to merge these columns into a single column. A transformation expression combines both values with proper spacing. This creates a new column containing full names. The merged data improves readability and presentation. It is useful for generating reports and documents. The operation simplifies data handling tasks.

Program 8 - Creating New Columns Using GREL Expressions

i)Create new columns using GREL expressions based on existing column values.

A dataset contains columns Name, Marks and Department, but does not include derived information required for analysis.

Question:

How will you use GREL in OpenRefine to:

a. Create a new column Final Marks by adding bonus marks.

b. Generate a Result column based on marks.

c. Construct an Email ID column using Name and Department.

Explain the steps and justify how creating new columns improves data usability.

Program 9 - Data Reconciliation and Text-to-Numeric Type Conversion

i)Demonstrate the process of data reconciliation by linking the dataset with external data sources.

ii)Data Type Conversion

A sales dataset contains numerical values stored as text, which prevents performing arithmetic operations.

Write a procedure using OpenRefine and GREL expressions to convert text-based numeric values into numeric format. Further, demonstrate how to perform calculations such as total sales using the converted data.

Program 10 - Reusable Project Templates and Keyword Filtering in OpenRefine

i)Create a reusable OpenRefine project template and demonstrate how it can be applied to multiple datasets.

ii)Keyword Filtering

A researcher is working with a large dataset containing various topics. They need to extract only records related to a specific keyword, such as AI. Manually searching through the dataset is time-consuming. OpenRefine is used to apply a text filter on the column. The keyword is entered and only relevant records are displayed. This helps in focusing on required data. It improves efficiency in data analysis. The process saves time and enhances productivity.

COURSE OUTCOMES

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

CO1	Apply OpenRefine to import, explore and analyze datasetsusing facets.
CO2	Perform data cleaning and standardization including handling missing values, removing duplicates, formatting textand date standardization.
CO3	Implement data transformation techniques using GREL, including column splitting/merging, filtering, advanced expressions and creation of new columns.
CO4	Demonstrate advanced data processing and integration, including data reconciliation, data type conversion, workflow reuse and efficient data extraction.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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.			

Textbooks:

1. https://ds-tutorials.github.io/openrefine-tutorial/
2. https://odl.ischool.uw.edu/openrefine_tutorial/
3. https://meta.wikimedia.org/wiki/...Tutorial.pdf

Web Links:

1. https://www.getty.edu/research/...openrefine_tutorial.pdf

RNS
INSTITUTE OF TECHNOLOGY
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	DOCKER PLAYGROUND		
Course Code	25BCSAE307F	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. Understand containerization concepts and the role of Docker in modern software development.
2. Install, configure and use Docker for creating and managing containers.
3. Build, deploy and manage containerized applications using Docker tools.
4. Work with Docker images, volumes, networks and Docker Compose.
5. Develop practical skills in deploying scalable applications using containers.

Program 1 - Docker Installation & Verification and Basic Container Operations

i) Install Docker on Linux or Windows and verify the installation.

ii) Run a basic Docker container using the hello-world image.

iii) Execute Docker commands such as docker run hello-world and docker info.

iv) Launch an Ubuntu container in interactive mode.

v) Execute commands inside a running container using docker exec.

vi) List available containers using docker ps -a.

Program 2 - Working with Docker Images and Creating a Simple Dockerfile

i) Pull, list and remove Docker images.

ii) Compare image sizes.

iii) Explore image layers using docker history.

iv) Create a Dockerfile for a simple Python application.

v) Build and run the Docker image.

vi) Understand Docker image layers and caching.

Program 3 - Container Lifecycle Management and Introduction to Volumes

i) Start, stop, restartand remove Docker containers.

ii) Inspect container metadata using Docker commands.

iii) Create named Docker volumes. iv) Persist data across container restarts. v) Compare bind mounts and Docker volumes.
Program 4 - Docker Networking
i) Create a custom bridge network. ii) Enable communication between multiple containers. iii) Demonstrate web server and database communication using Docker networking.
Program 5 - Multi-Container Setup Using Docker Compose
i) Create a Docker Compose configuration file (docker-compose.yml). ii) Deploy an application consisting of a backend service and a database. iii) Use environment variables in Docker Compose.
Program 6 - Building Multi-Stage Dockerfiles
i) Optimize Docker image size using multi-stage builds. ii) Separate build and runtime stages. iii) Build a multi-stage Dockerfile for a Node.js or Java application.
Program 7 - Docker Hub and Image Versioning
i) Tag Docker images with version numbers. ii) Push Docker images to Docker Hub. iii) Implement semantic versioning for Docker images.
Program 8 - Container Orchestration Basics Using Docker Swarm
i) Initialize a Docker Swarm cluster. ii) Deploy replicated services. iii) Scale Docker services dynamically.
Program 9 - Reverse Proxy Using Nginx
i) Deploy multiple backend containers. ii) Configure Nginx as a reverse proxy. iii) Implement load balancing across backend containers.
Program 10 - CI/CD Integration with Docker
i) Integrate Docker with a CI/CD pipeline. ii) Automatically build Docker images. iii) Push Docker images to a container registry using GitHub Actions or Jenkins.

Program 11 - Monitoring and Logging
i) Explore monitoring tools such as Prometheus and Grafana. ii) Monitor container performance. iii) Implement centralized logging using Docker logs and ELK Stack basics.
Program 12 - Microservices, Secure and Production-Ready Deployment
i) Build a multi-service architecture consisting of frontend, backend and database services. ii) Deploy the application using Docker Compose. iii) Implement Docker security best practices, including non-root containers, secrets management and environment isolation. iv) Optimize Docker images for production deployment.

COURSE OUTCOMES

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

CO1	Understand and apply Docker concepts for containerization.
CO2	Build and manage Docker images and containers effectively.
CO3	Develop and deploy applications using Docker and Docker Compose.
CO4	Implement container networking and storage solutions.
CO5	Apply Docker in real-world development and deployment scenarios.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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.			

Textbook:

1. Nigel Poulton – Docker Deep Dive
2. Karl Matthias & Sean Kane – Docker: Up & Running

Web Links:

1. https://docs.docker.com
2. https://hub.docker.com
3. https://play-with-docker.com

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

Course Title	KAGGLE		
Course Code	25BCSAE307G	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. Understand and utilize the Kaggle platform for accessing datasets and executing notebooks in a cloud-based environment.
2. Apply data handling techniques to import, explore and summarize datasets using Python libraries.
3. Perform data preprocessing and cleaning by handling missing values, duplicates and inconsistencies in datasets.
4. Analyze and visualize data using appropriate charts and graphical representations to derive meaningful insights.
5. Build and evaluate basic machine learning models and document the complete data analysis workflow using Kaggle notebooks.

Program 1 - Introduction to Kaggle

a. Create Kaggle account

b. Explore datasets

c. Run an existing notebook

c. Understand notebook outputs

Program 2 - Data Loading and Exploration

a. Import dataset

b. Display rows and columns

c. Use head(), tail(), info() and describe()

d. Generate summary statistics

Program 3 - Data Cleaning

a. Identify missing values

b. Handle missing values

c. Remove duplicates

d. Verify cleaned data

Program 4 - Data Visualization
a. Create a bar chart b. Plot a histogram c. Generate a pie chart d. Interpret the visualizations
Program 5 - Data Analysis
a. Filter data b. Perform GroupBy operations c. Conduct correlation analysis d. Generate insights
Program 6 - Basic Machine Learning
a. Perform train-test split b. Build a Logistic Regression model c. Evaluate the model using accuracy score d. Generate and interpret the confusion matrix
Program 7 - Mini Project
a. Select a Kaggle dataset b. Perform data cleaning c. Create visualizations d. Build a simple prediction model e. Document findings

COURSE OUTCOMES

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

CO1	Demonstrate the use of the Kaggle platform for accessing datasets, executing notebooks and managing cloud-based data analysis tasks.
CO2	Apply data handling techniques using Pandas to import, explore and summarize datasets.
CO3	Perform data preprocessing operations such as handling missing values, removing duplicates and preparing data for analysis.
CO4	Create and interpret visualizations to identify trends, patterns and relationships in datasets.
CO5	Develop and evaluate a basic machine learning model using Kaggle datasets and document the findings through a mini project.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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.			

Textbooks:

1. Python for Data Analysis: Data Wrangling with Pandas, NumPy and Jupyter by Wes McKinney, 3rd Edition, 2022.

Reference Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, 2nd Edition, by Aurelien Geron.
2. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking by Tom Fawcett.

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

Course Title	BURP SUITE COMMUNITY EDITION		
Course Code	25BCSAE307H	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. Understand Web Application Security and OWASP Top 10.
2. Learn Burp Suite tools (Proxy, Repeater, Intruder, Decoder).
3. Perform basic vulnerability testing (SQLi, XSS).
4. Analyze HTTP/HTTPS traffic.

Installation and Setup

Installation and proxy configuration

A junior security analyst is assigned to set up a web application testing environment using Burp Suite Community Edition. The tool should be downloaded and installed on the system. A new project should be created and launched successfully. The browser must be configured to route traffic through Burp Suite's proxy listener. The CA certificate should be exported and installed in the browser to intercept HTTPS traffic. The Intercept tab should be enabled to confirm that requests are being captured. The overall setup should be functional, correctly configured and ready for web traffic analysis.

Program 1 - Demonstration of Intercepting HTTP requests

A security analyst is assigned to investigate unusual login activity on a web application by intercepting HTTP requests using Burp Suite. The browser proxy should be active and routing traffic through Burp Suite. A login attempt should be made on the target web application. The HTTP request containing the login credentials should be captured in the Intercept tab. The request headers, parameters and body should be examined carefully. The captured request should be forwarded and the server response should be observed. The overall process should demonstrate clear understanding of HTTP request structure and interception workflow.

Program 2 - Demonstration of Target mapping and scope

A security analyst is assigned to define the scope of a client's web application before beginning a security test using Burp Suite.

The target domain provided by the client should be added to Burp Suite's Target tab. The scope should be configured to include only the specified domain and exclude all others. The browser should be used to navigate the target application so that Burp Suite maps its structure. The Site Map should display all discovered pages, directories and endpoints. Out-of-scope traffic should be filtered and verified as excluded. The overall configuration should ensure focused, organized and scope-compliant testing.
Program 3 - Repeater for Input Validation
A security analyst is assigned to test a feedback form on a web application for inconsistent input validation behavior using Burp Suite's Repeater tool. A form submission request should be captured using the Proxy Intercept tab. The captured request should be sent to the Repeater tool for controlled retesting. Input field values should be manually modified to include edge cases such as empty fields, special characters and oversized inputs. Each modified request should be sent and the server response should be carefully observed. Differences in server behavior for different inputs should be documented. The overall process should demonstrate how Repeater is used to identify weak or inconsistent input handling on the server side.
Program 4 - Intruder for Brute Force Testing
A security analyst is assigned to test a login form for weak password policies using Burp Suite's Intruder tool. A login request should be captured using the Proxy and sent to the Intruder tool. The attack type should be set to Sniper and the password field should be marked as the payload position. A small wordlist of common passwords should be loaded as the payload. The attack should be launched and all responses should be collected. Response lengths and status codes should be compared to identify a successful login attempt. The overall process should demonstrate how Intruder is used to detect weak credential policies through automated request manipulation.
Program 5 - SQL Injection testing
A security analyst is assigned to test a login form for SQL injection vulnerabilities using Burp Suite. A login request should be captured using the Proxy and forwarded to the Repeater tool. Common SQL injection payloads should be inserted into the username and password fields. Each modified request should be submitted and the server response should be analyzed. Indicators of vulnerability such as authentication bypass, error messages, or abnormal responses should be identified. Multiple payloads should be tested to confirm the presence or absence of the vulnerability. The overall process should demonstrate how SQL injection can be detected and confirmed through manual testing using Burp Suite.

Program 6 - XSS testing

A security analyst is assigned to identify Cross-Site Scripting vulnerabilities in a web application where users have reported unusual pop-ups using Burp Suite. Input fields and URL parameters on the target application should be identified as potential injection points. XSS payloads such as `<script>alert('XSS')</script>` should be inserted into these fields using the Repeater tool. Each request should be submitted and the server response should be inspected for reflected or stored payload content. Successful execution of the payload in the browser should be observed and noted. Both reflected and stored XSS scenarios should be attempted wherever applicable. The overall process should demonstrate how Burp Suite is used to detect and confirm XSS vulnerabilities through systematic payload testing.

Program 7 - Authentication and Session testing

A security analyst is assigned to test session management on a web application where users have reported session hijacking issues using Burp Suite. A valid login should be performed and the session token or cookie should be captured in the Proxy.

The session token should be examined for predictability, encoding and security attributes such as HTTP Only and Secure flags. The Repeater tool should be used to replay requests with modified or expired session tokens to test server-side validation. Attempts to access protected pages without a valid session should be made and the responses analyzed. The behavior of the application after logout should be verified to check if the session is properly invalidated. The overall process should demonstrate how Burp Suite is used to evaluate the strength and security of session management mechanisms.

Program 8 - Decoder for Encoded Traffic

A security analyst is assigned to decode and analyze suspicious encoded data found within intercepted HTTP requests using Burp Suite's Decoder tool. A request containing encoded parameter values should be captured using the Proxy Intercept tab. The encoded value should be sent to the Decoder tool for analysis. The data should be decoded using appropriate formats such as URL encoding, Base64 and HTML encoding. The decoded content should be read and examined for sensitive or suspicious information. Re-encoding should also be performed to understand how the data would appear when transmitted. The overall process should demonstrate how the Decoder tool is used to interpret and manipulate encoded data found in web application traffic.

Program 9 - Demonstrating the use of Comparer tool
A security analyst is assigned to investigate two similar HTTP responses that behave differently on a web application using Burp Suite's Comparer tool. Two HTTP responses should be captured for the same request made under different conditions such as valid and invalid credentials. Both responses should be sent to the Comparer tool from the Proxy or Repeater. A word-level or byte-level comparison should be performed between the two responses. Differences in response content, headers, length, or status codes should be identified and highlighted. The significance of each difference in the context of application behavior should be analyzed. The overall process should demonstrate how the Comparer tool is used to detect subtle but meaningful differences that reveal application logic flaws.
Program 10 - Vulnerability assessment
A security analyst is assigned to perform a focused security audit of a web application by identifying multiple vulnerability types using Burp Suite. The target application should be added to scope and fully mapped using the Site Map. At least three types of vulnerabilities such as SQL injection, XSS and insecure input handling should be tested across different pages and parameters. Each vulnerability test should be carried out using the appropriate Burp Suite tool such as Proxy, Repeater, or Intruder. All findings should be recorded with the affected URL, parameter, payload used and observed response. A brief summary report of identified vulnerabilities should be prepared with severity observations. The overall process should demonstrate how Burp Suite is used to conduct a structured and documented web application security assessment.
Program 11 - Input validation testing
A security analyst is assigned to test the robustness of input sanitization on a web application that behaves unpredictably with certain user inputs using Burp Suite. Input fields across multiple pages of the application should be identified and cataloged. A variety of test inputs including special characters, excessively long strings, null values and script tags should be submitted using the Repeater tool. Server responses for each input type should be captured and compared. Inputs that cause unexpected behavior such as errors, crashes, or unfiltered output should be flagged. Observations should be documented indicating whether the application accepts, rejects, or mishandles each input type. The overall process should demonstrate how Burp Suite is used to systematically evaluate whether a web application properly sanitizes and validates user-supplied data.

Program 12 - Simulated Penetration Testing on DVWA/OWASP Juice Shop
A security analyst is assigned to perform a simulated penetration test on a deliberately vulnerable web application such as DVWA or OWASP Juice Shop using Burp Suite. The vulnerable application should be set up and confirmed accessible through the configured browser proxy. At least four attack scenarios should be attempted including SQL injection, XSS, brute force login and session token analysis. Each attack should be performed using the relevant Burp Suite tools covered across previous programs. All steps, payloads and observations for each attack should be documented systematically. A consolidated summary of all vulnerabilities discovered during the session should be prepared. The overall process should demonstrate how Burp Suite is used as a complete penetration testing tool in a controlled and ethical lab environment.

COURSE OUTCOMES

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

CO1	Understand web vulnerabilities.
CO2	Apply Burp Suite tools for web application security testing.
CO3	Apply penetration testing techniques on web applications.
CO4	Analyze SQL Injection (SQLi) and Cross-Site Scripting (XSS) vulnerabilities.
CO5	Apply vulnerability reporting techniques to document security findings.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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.			

Textbook:

1. The Web Application Hacker's Handbook

Reference:

1. OWASP
2. PortSwigger Web Security Academy

RNS 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	POSTMAN		
Course Code	25BCSAE307I	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. Understand API fundamentals including HTTP methods, request-response architecture and status codes.
2. Use Postman to create and organize API requests.
3. Validate API responses using Postman.
4. Automate API testing using JavaScript scripts.

Program 1 - Demonstrate the API Fundamentals
To understand API concepts, HTTP methods, request-response structure and status codes. An API enables communication between systems. HTTP methods like GET, POST, PUT, DELETE define operations and responses include status codes and data. As an example, send a GET request and identify the HTTP method, status code and response structure.
Program 2 - Demonstrate the Postman Interface Effectively
Postman interface allows request creation, execution and organization using collections and folders.As an example, create and organize API requests using the Postman interface and execute them successfully.
Program 3 - Demonstrate API Testing
API testing ensures correctness by validating status codes, headers and response data. As an example, write and execute test scripts to validate the status code, response time and response data of a given API.
Program 4 - Demonstrate API Validation using Scripts
Comprehensive API validation ensures an API works correctly by checking its status code, response content, headers, performance and data structure using automated test scripts. As an example, perform comprehensive API validation by verifying the status code, response body, headers, response time and JSON structure using test scripts.

Program 5 - Demonstrate Automation of API Testing using Scripts in Postman
Automation in Postman enables efficient and repeatable API testing using Pre-request Scripts and Test Scripts. As an example, execute an API request and automate validation of response parameters and dynamic data handling using scripts.
Program 6 - Demonstrate Working with Variables and Environments
Use environment, global and local variables to store and manage dynamic data, enabling flexible and reusable API test execution. As an example, create and use environment, global and local variables to manage dynamic data and execute API requests dynamically.
Program 7 - Demonstrate Debugging API Responses using Postman Console
Debug API responses, identify errors and monitor performance using the Postman Console and test scripts. As an example, execute API requests and perform debugging, error detection and performance monitoring using test scripts and console logs.
Program 8 - API Chaining using Variables and Data Passing
Implement API chaining to pass data between requests and verify its correctness using test scripts. As an example, implement API chaining and validate that the data passed between requests is correctly handled using test scripts.
Program 9 - Advanced API Chaining using Dynamic IDs and Multiple Requests
Illustrate API chaining by extracting data from one API response, using it in another request and validating the result with test scripts. As an example, extract data from one API response, use it in another request and validate the output using test scripts.
Program 10 - Demonstrate Basic Security Testing of APIs using Postman
Test an API with authentication by validating both successful access with valid credentials and failure responses for unauthorized requests using Postman. As an example, test an API using authentication and verify both authorized and unauthorized access using Postman.
Program 11 - Demonstrate Performance Analysis and Error Troubleshooting using Postman Console
Debug API responses, identify errors and monitor performance using the Postman Console and test scripts. As an example, execute API requests and perform debugging, error detection and performance monitoring using test scripts and console logs.
Program 12 - Demonstrate Debugging API Responses using Postman Console
Debug API responses, identify errors and monitor performance using the Postman Console and test scripts. As an example, a company has developed an e-commerce application where users are facing issues such as failed logins, slow API responses and incorrect order

details. As an API tester, use Postman to debug the APIs by analyzing responses in the Postman Console. Write appropriate test scripts to validate authentication, check response accuracy, identify errors and monitor performance.

COURSE OUTCOMES

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

CO1	Understand API fundamentals including HTTP methods, request-response architecture and status codes.
CO2	Apply Postman to organize API requests and responses for testing.
CO3	Design API test cases for functionality and performance testing.
CO4	Apply Postman scripting features to automate API testing and validation.
CO5	Develop end-to-end API testing solutions including debugging, data handling and basic security testing.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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.			

Textbook:

1. https://www.oreilly.com/library/view/.../9781800569201

Web Links:

1. https://learning.postman.com/
2. https://learning.postman.com/docs/
3. https://www.postman.com/.../postman-api-fundamentals-student-expert/
4. https://www.researchgate.net/.../API_Testing_Using_Postman_Tool
5. https://jsonplaceholder.typicode.com/
6. https://public-apis.io/

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

Course Title	COMMUNITY PROJECT / SOCIETAL PROJECT (PROJECT-BASED LEARNING)		
Course Code	25BCBPR308	CIE MARKS	50
(L : T : P)	(0:0:0)+(30 Hours/Sem)	SEE MARKS	50
Credits	1	EXAM HOURS	3

COURSE LEARNING OBJECTIVES

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.

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

Restrictions and professional conduct:

Societal projects should be academic, socially responsible, technically focused, and politically neutral. Students and faculty must maintain professionalism, ethical conduct, and respect while working with communities and stakeholders and avoid any political, ideological, commercial, or disruptive activities. Projects should also consider practical constraints such as cost, resources, safety, social impact, and stakeholder cooperation.

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

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.	BroadAreas/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	BasicCrop 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	2,3,15
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. 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 geotagged 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 onsoil 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 locationbased 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 Too	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 skilltraining 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 unorganizedsector 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

Teaching Learning Process:
The suggested TLP for Community / Societal Project.

The following structured sequence of activities is to be carried out during the semester.

Week / Phase	**Teaching–Learning Activity**	**Faculty Role**	**Student Activity**	**Output / Evidence**
1	Course introduction, objectives, ethics, conduct, and assessment briefing	Explain course scope, rules, rubrics, and expected deliverables	Understand course requirements and form teams	Team list, course plan
2	Community problem identification and topic selection	Guide Topic Framing and Feasibility	Identify local issues through discussion and observation	Shortlisted project topic
3	Stakeholder interaction and preliminary survey	Demonstrate survey methods and stakeholder engagement	Collect initial data from community/users	Survey notes, interview record
4	Problem statement and objective formulation	Review and refine problem statement	Prepare problem statement, scope, and objectives	Approved problem statement
5	Literature/case study review and baseline analysis	Suggest relevant references and examples	Study existing solutions and compare approaches	Baseline analysis summary
6	Project planning and role assignment	Help teams organize tasks, timeline, and responsibilities	Plan activities, divide roles, and prepare schedule	Work plan / Gantt chart
7	Feasibility analysis and solution brainstorming	Check technical feasibility and practicality	Generate possible engineering solutions	Solution alternatives sheet

8	Selection of final solution and design approach	Review final concept and design logic	Finalize concept, methods, and expected outcome	Concept note / design outline
9	Data processing / calculations / sizing / mapping	Support Computations and technical review	Analyze data, perform calculations, prepare maps/chart	Technical analysis sheet
10	Prototype / model / audit / DPR / app development	Monitor progress and correct technical issues	Develop the selected output	Working model/draft DPR/map/app
11	Testing, refinement, and implementation planning	Evaluate technical correctness and improvements	Test prototype or validate proposal with users	Test results / revised output
12	Field implementation / demonstration / awareness activity	Facilitate community interaction and documentation	Conduct implementation or demonstration	Photos, attendance, implementation record
13	Documentation of outcomes, feedback, and impact	Review report structure and completeness	Compile findings, stakeholder feedback, and observations	Draft report
14	Presentation preparation and viva practice	Conduct mock presentation and feedback	Prepare slides/poster and rehearse	Presentation deck
15	Final presentation, submission, and reflection	Evaluate performance using rubrics	Present project, submit report, reflect on learning	Final report, presentation, viva

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 engagement; comprehensive 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 omissions; 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/environment	16	Clearly articulated with strong evidence and context. (16)	Clear with reasonable evidence. (12)	Problem stated but evidence/context weak. (8)	Poorly defined or irrelevant. (4)
Formation of problem statement; engineering approach, innovation & feasibility	20	Precise demonstrates engineering depth, novelty and feasibility. (20)	Technically sound and feasible with minor limitations. (15)	Basic; limited technical depth or Feasibility concerns. (10)	No viable engineering approach or infeasible plan. (5)

Interaction with stakeholders; data collection, analysis & interpretation	16	Excellent engagement; rigorous data collection and insightful analysis. (16)	Good engagement and acceptable analysis. (12)	Superficial data collection and limited analysis. (8)	Inadequate or no data; poor analysis. (4)
Project documentation; clarity, organization & problem-solving demonstrated	10	Comprehensive, technically rigorous, and systematic. (20)	Complete and technically correct with minor gaps (15)	Lacks technical depth or has notable omissions. (10)	Missing key elements or technically incorrect. (5)
Communication & presentation skills	12	Presents expertly, defends work persuasively and answers all queries correctly. (12)	Good presentation and satisfactory answers to most queries. (9)	Good presentation and satisfactory answers to most queries. (9)	Poor presentation and unable to defend work. (3)
Contribution & benefit to society (impact, sustainability)	16	Clear, evidence based impact assessment with scaling Recommendations. (16)	Reasonable impact discussion and Sustainability considerations. (12)	Limited evaluation of impact or sustainability. (8)	Limited evaluation of impact or sustainability (8)

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) $\geq$ 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.

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.

Web links 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.

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.

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 a) Holding the Discus b) Initial Stance Primary Swing c) Rotation d) 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	Exhibit basic skills and techniques in hockey, including passing, trapping, dribbling, tackling, penalty strokes/corners and understand rules and duties of officials.

ASSESSMENT AND EVALUATION REGULATIONS OF NCMC

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

Course Title	YOGA		
Course Code	25BYOGA309	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 and Spiritual objectives of Yoga.
2. Explain the meaning of Yoga.
3. Discuss the importance of 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 OUTCOMES

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, precautions and awareness of their benefits.
CO5	Practice Pranayama techniques (Kapalabhati, Ujjayi, Sheetali) to improve breathing control, concentration and overall health.

ASSESSMENT AND EVALUATION REGULATIONS OF NCMC

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. 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 by T.K.V. Desikachar.
2. Asana Pranayama Mudra Bandha by Swami Satyananda Saraswati.

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

Course Title	**MUSIC**		
Course Code	**25BMUSK309**	**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 origin, evolutionand 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 awarenessand 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 and Mela.	
Module-2	**3 Hours**
Compositions: Introduction to the few types of compositions in Carnatic Music—Geethe, Swarajathi, Varna and Madhyama Kala Kriti.	
Module-3	**3 Hours**
Compositions: Introduction to the few types of compositions 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 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 AND EVALUATION REGULATIONS OF NCMC

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.			

Textbooks:

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.

RNS
INSTITUTE OF TECHNOLOGY
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	**ADDITIONAL MATHEMATICS-I**		
Course Code	**25BMADP310**	**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 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	**9 Hours**
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:** Scalars and vectors. Addition, subtraction and multiplication of vectors. Dot and Cross products, Problems.	
Module-2	**9 Hours**
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	**9 Hours**
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	**9 Hours**
Integral Calculus: Review of elementary integral 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	**9 Hours**
Ordinary Differential Equations (ODEs): Introduction-solutions of first order and first-degree differential equations: Exact differential equations, Linear differential equations, Equations reducible to exact differential equations 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 the 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 AND EVALUATION REGULATIONS OF NCMC

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 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: B. S. Grewal, Khanna Publishers, 43rd Edition, 2015.

Reference Books:

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

IV SEMESTER SUBJECTS

B.E. (Computer Science and Engineering)
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 and solve discrete mathematical problems.
2. Analyze counting techniques and the principle of counting to solve combinatorial problems.
3. Familiarize with the fundamental concepts of the principle of inclusion and exclusion to build a strong foundation in discrete mathematics.
4. Solve engineering problems using graph-theoretic 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. **Textbook 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. **Textbook 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. **Textbook 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. **Textbook 1: Chapter 8 (8.1 – 8.4)**	

Module-5	9 Hours
Introduction to Graph Theory: Definitions and Examples, vertex degree, Complete graph, Regular graph, Peterson graph, bipartite graph, complete bipartite graph, Subgraphs, 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 AND EVALUATION REGULATIONS OF ASC

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.
CIE: Theory + Practical (A + B)	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)	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. 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 Edition, PRISM Books Pvt. Ltd., Bengaluru, 2012.
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.

RNS
INSTITUTE OF TECHNOLOGY
ESTD : 2001
An Institute with a Difference

B.E. (Computer Science and Engineering)
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.

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.).

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 Normal Forms, Boyce-Codd Normal Form. **Relational Database Design Algorithms:** Functional Dependencies: Inference Rules, Equivalence and Minimal Cover, Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, About Nulls, Dangling Tuples. **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)**

ESTD:2001 An Institute with a Difference

PRACTICAL COMPONENT OF IPCC

Sl. No.	EXPERIMENTS
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 an 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 salesmen 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. e. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

3	Consider the following schema for a 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 the rating of all movies directed by 'Steven Spielberg' to 5.
4	Consider the following schema for a 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 the 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 FinalIA ¡ 12 then CAT = 'Weak'. Give these details only for 8th semester A, B and C section students.

5	Consider the following schema for a 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 salaryand 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 and perform basic CRUD (Create, Read, Update and Delete) operations. Download MongoDB Community Server from: 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 AND EVALUATION REGULATIONS OF IPCC

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 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 IPCC 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, McGraw Hill, 2013.

Web Links:

1. `https://www.youtube.com/playlist?...PLJwqWlxkbMhdAM5pv3Yfx5CWkm83tZy0s`
2. `https://www.youtube.com/playlist?...PLxCzCOWd7aiFAN6I8CuViBuCdJgiOkT2Y`
3. `https://www.youtube.com/watch?v=c5HAwKX-suM`
4. `https://www.geeksforgeeks.org/dbms/dbms/`

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.

Mini Project:

1. `https://www.hackerrank.com/domains/sql`
2. Case Study based Flip Class

B.E. (Computer Science and Engineering)
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 analysing 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:** Finding the Maximum and Minimum. **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: Balanced Search Trees (AVL trees,2-3 trees), Heaps and Heapsort **Greedy Technique:** Prim's Algorithm, Kruskals Algorithm, Dijkstra's Algorithm, Huffman Trees, Knapsack Problem, Job sequencing with deadlines.	
Textbook 1: Chapter-6 (6.3-6.4) Chapter-9 (9.1-9.4) **Text book 2: Chapter-4 (4.2,4.4)**	
Module-4	**9 Hours**
Dynamic Programming: The Knapsack Problem and Memory functions, Warshall's Algorithm and Floyd's Algorithms, Multistage Graph, 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
Limitations of Algorithm Power: P, NP, and NP-Complete Problems. **Coping with the Limitations of Algorithm Power:** Backtracking: n-Queens Problem, Hamiltonian Circuit Problem, Subset-Sum Problem, Graph Coloring. **Branch-and-Bound:** Assignment Problem, Knapsack Problem, Traveling Salesman Problem. **Approximation Algorithms for NP-Hard Problems:** Approximation Algorithms for Traveling Salesman Problem, Approximation Algorithms for Knapsack Problem. **Textbook 1: Chapter-11 (11.3), Chapter-12 (12.1 - 12.3)** **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 to determine the efficiency of algorithms.
CO2	Apply decrease-and-conquer, divide-and-conquer, transform-and-conquer, and greedy strategies to design and implement efficient algorithms for computational problems.
CO3	Apply dynamic programming, backtracking, and branch-and-bound techniques to solve optimization and combinatorial problems.
CO4	Analyze the performance and suitability of different algorithmic paradigms by comparing their time and space complexities for various problem domains.
CO5	Analyze computational complexity classes (P, NP, NP-Complete) and evaluate appropriate exact and approximation algorithms for solving computationally hard problems.

ESTD:2001 An Institute with a Difference

ASSESSMENT AND EVALUATION REGULATIONS OF PCC

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 a minimum of 40% scored in every component of CIE.			

Textbooks:

1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson, 2021.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, 2nd Edition, Universities Press (widely cited Indian edition/reprint), 2019.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.

Activity Based Learning (Suggested Activities in Class):

1. Hackathons

Self-Study:

1. Probabilistic Analysis and Randomized Algorithms
2. Maximum Flow

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

Course Title	ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING		
Course Code	25BCSTH404	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. Explain the basics of Artificial Intelligence & Machine Learning algorithms.
2. Identify the problems applicable for Artificial Intelligence and Machine Learning techniques.
3. Discuss knowledge representation issues and different kind of learning algorithms.
4. Demonstrate application of algorithms to real-world problems.

Module-1	9 Hours
Introduction: What is AI? The state of the art. **Intelligent agents:** Agents and environment, Good Behavior: The Concept of Rationality, The nature of Environments, The structure of Agents. **Textbook 1: Chapter 1(1.1, 1.4), Chapter 2 (2.1 - 2.4)**	
Module-2	**9 Hours**
Problem Solving by Searching: Problem-solving agents, Example Problems, Search algorithms, Uninformed Search Strategies, Informed search Strategies. **Adversarial search:** Game Theory, Optimal Decision in games. **Textbook 1: Chapter 3 (3.1 - 3.5), Chapter 6 (6.1 - 6.2)**	
Module-3	**9 Hours**
Introduction: Need for Machine Learning, Machine Learning Explained, Machine Learning in Relation to other Fields, Types of Machine Learning, Challenges of Machine Learning, Machine Learning Process, Machine Learning Applications. **Similarity-based Learning:** Nearest-Neighbor Learning, Weighted K-Nearest-Neighbor Algorithm, Nearest Centroid Classifier, Locally Weighted Regression (LWR). **Regression Analysis:** Introduction to Regression, Introduction to Linearity correlation and Causation, Introduction to Linear Regression, Multiple Linear Regression, Polynomial Regression, Logistic Regression. **Textbook 2: Chapter 1, Chapter 4 (4.2 - 4.5), Chapter 5 (5.1 - 5.3, 5.5 - 5.7)**	
Module-4	**9 Hours**
Bayesian Learning: Introduction to Probability-based Learning, Fundamentals of Bayes Theorem, Classification Using Bayes Model, Naïve Bayes Algorithm for Continuous Attributes. **Artificial Neural Networks:** Introduction, Biological Neurons, Artificial Neurons, Perceptron and Learning Theory, Types of Artificial Neural Networks, Popular Applications of Artificial Neural Networks, Advantages and Disadvantages of ANN, Challenges of ANN. **Textbook 2: Chapter 8 (8.1 - 8.4), Chapter 10 (10.1 - 10.5, 10.9 - 10.11)**	
Module-5	**9 Hours**

Decision Tree Learning: Introduction to Decision Tree Learning Model, Decision Tree Induction Algorithms. **Reinforcement Learning:** Overview of Reinforcement Learning, Scope of Reinforcement Learning, Reinforcement Learning as Machine Learning, Components of Reinforcement Learning, Model Free Methods, Q-Learning. **Textbook 2: Chapter 6 (6.1 - 6.2), Chapter 14 (14.1 - 14.4, 14.8, 14.9)**

COURSE OUTCOMES

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

CO1	Understand how intelligent agents work in real-world applications.
CO2	Utilize search algorithms to construct effective AI agent solutions.
CO3	Examine machine learning techniques to address real-world problems.
CO4	Implement Bayesian learning methods for solving classification tasks.
CO5	Demonstrate machine learning applications using neural networks and reinforcement learning approaches.

ASSESSMENT AND EVALUATION REGULATIONS OF PCC

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 maximum 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 3 hours.
CIE + SEE (A+B+C)	100	40	Final marks shall be the sum of marks scored in components A, Band C.
The minimum marks to be secured in CIE is 20 (40%) out of allotted 50 marks to appear for SEE.			

Textbooks:

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson Education, 2023.
2. S Sridhar, M Vijayalakshmi, Machine Learning, Oxford University Press, 2021, First Edition.

Reference Books:

1. Aurelien Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools and Techniques to Build Intelligent Systems, 3rd Edition, Shroff / O'Reilly Media, 2022.
2. Tom M. Mitchell, Machine Learning, 1st Edition, McGraw Hill Education, 2017.
3. Ethem Alpaydin, Introduction to Machine Learning, 2nd Edition, PHI Learning Pvt. Ltd., 2014.
4. George F. Luger, Structures and Strategies for Complex Problem Solving, 5th Edition, Pearson Education, 2011.
5. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, 2nd Edition, Springer New York, NY, 2009.

Web Links:

1. https://nptel.ac.in/courses/106106139
2. https://nptel.ac.in/courses/106102220
3. https://www.coursera.org/.../machine-learning-introduction

Self Study Topics:

1. Compare Uninformed vs Informed Search.
2. Implement BFS & DFS in Python.
3. Explore A* Search with heuristics.
4. Learn Bayes' theorem applications in classification.
5. Naive Bayes text classification example.
6. Neural Networks and Backpropagation demo.
7. Q-Learning on a grid world.
8. Temporal Difference vs Monte Carlo Method.
9. Utility-based Agents vs Goal-based Agents.

B.E. (Computer Science and Engineering)
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 - Bellman-Ford Algorithm – Dynamic Routing in Unstable Networks
Objective: Implement a C/C++ program to find the shortest paths from a single source vertex to all other vertices in a directed graph that may contain negative edge weights using the Bellman-Ford algorithm, including negative cycle detection.

Application: Optimizing autonomous delivery or logistics routes where certain pathways pass through energy-harvesting zones (recharging grids represented as negative weights), while ensuring the system flags infinite battery-draining loops (negative cycles).
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 AND EVALUATION REGULATIONS OF PCCL

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/

Activity Based Learning:

1. Simulation of Algorithms
2. Leet Coding

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

Course Title	INTRODUCTION TO UNIX		
Course Code	25BCSAE406A	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 tab-separated 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
101	John	Computer	85
102	Ravi	Electronics	78
103	Asha	Computer	92
104	Maya	Electronics	85
105	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
101	John	Computer
102	Ravi	Electronics
103	Asha	Computer
104	Maya	Electronics
105	Arjun	Mechanical
106	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 background, 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 decision-making, 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 AND EVALUATION REGULATIONS OF AEC

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. 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.

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

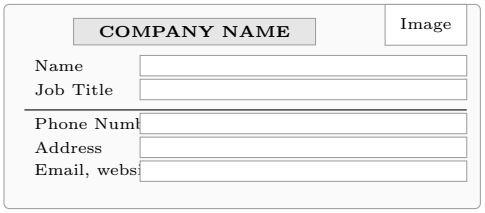
Course Title	MOBILE APP DEVELOPMENT		
Course Code	25BCSAE406B	CIE MARKS	50
L : T : P	(0:0:2)	SEE MARKS	50
Credits	1	EXAM HOURS	3

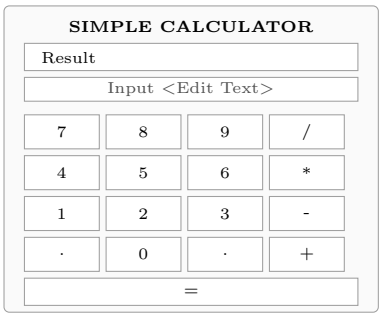
COURSE LEARNING OBJECTIVES

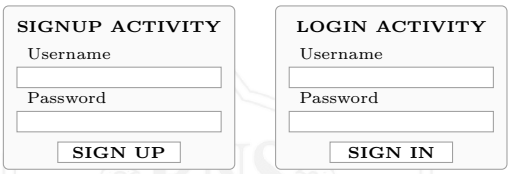
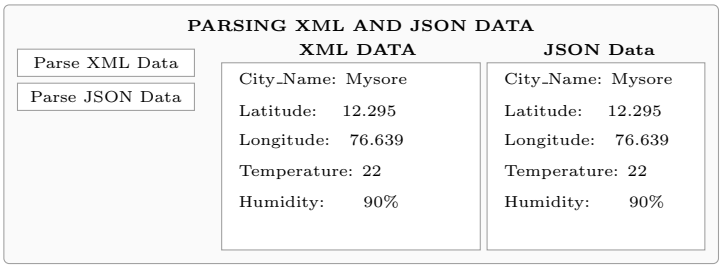
The purpose of the course is to facilitate the learner to:

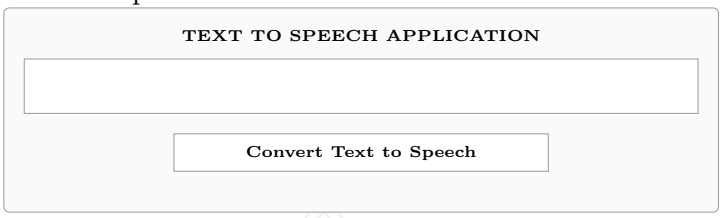
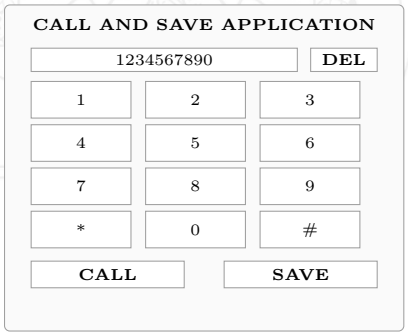
1. Develop basic Android applications by applying fundamental programming concepts and techniques.
2. Configure and use Android Studio IDE for designing, coding and testing mobile applications.
3. Design and implement user interfaces using Android UI components such as layouts, widgets and event handlers.
4. Create and manage local databases using SQLite for data storage and retrieval within Android apps.
5. Demonstrate data sharing techniques by implementing services, content providers and inter-process communication in Android.

EXPERIMENTS	
PART A	
1	A company wants to create a digital visiting card for its employees to be displayed in its Android application. The card should present the company's branding along with the employee's professional details in a clear and organized format. Design an Android application that displays the visiting card as per the given requirements.
2	A tuition center wants a simple Android application that allows students to perform basic arithmetic calculations during practice sessions. Design an Android application using TextView, EditText, and Button controls to create a calculator that supports Addition, Subtraction, Multiplication, and Division operations.

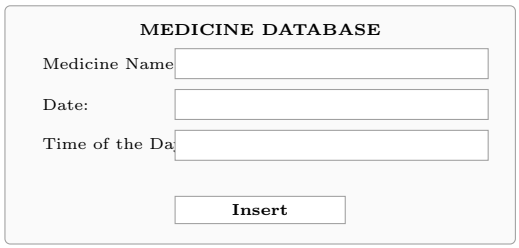


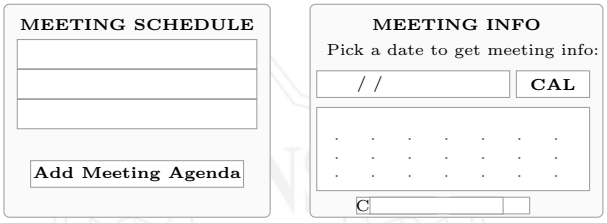
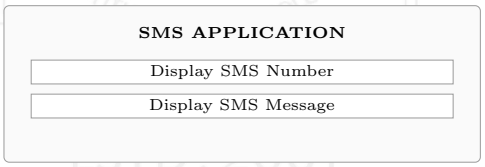
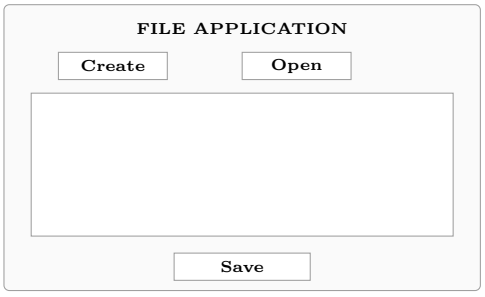


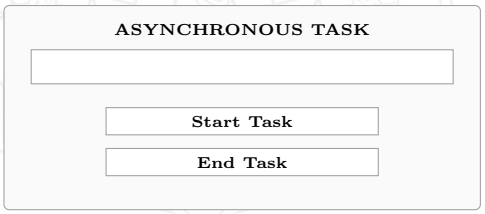
3	A company is developing an Android application that requires users to create an account before accessing its services. Design an application with Sign Up and Login activities, where the password is validated based on security requirements. After successful registration, allow users to log in using the created credentials. If the login is successful, display a "Successful Login" message; otherwise, display appropriate error messages. Restrict users to two login attempts, after which the Sign In button should be disabled. Use a Bundle to transfer the registered user information between activities. 
4	A mobile personalization company wants to develop an Android application that automatically changes the device wallpaper to enhance the user experience. Design an application that allows the user to set wallpapers, where clicking a button starts changing the wallpaper randomly every 30 seconds. 
5	A fitness application requires a simple counting feature to help users keep track of repetitions during workouts. Design an Android application with START and STOP buttons, where pressing START begins counting from 1 and continuously updates the count in a TextView until the STOP button is pressed. 
6	A weather monitoring company wants an Android application to compare weather data stored in different file formats. Design an application that reads weather information from both XML and JSON files. When the respective buttons are clicked, the application should parse the files and display the City Name, Latitude, Longitude, Temperature, and Humidity side by side for easy comparison. 

7	An educational mobile application aims to help users listen to written content for improved accessibility and learning. Design an Android application with an EditText for user input and a "Convert Text to Speech" button that converts the entered text into spoken audio using the device's Text-to-Speech feature. 
8	A mobile service provider wants to develop a simple Android dialer application that enables users to quickly make phone calls and save important contact numbers. Design an application with a phone dialer interface containing CALL and SAVE buttons, where the CALL button initiates a phone call to the entered number and the SAVE button stores the number in the device's contacts. 

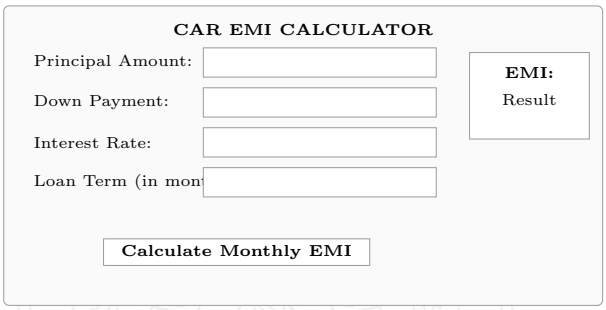
PART B

1	A healthcare center wants to develop an Android application that helps patients remember to take their medicines on time. Design an application that allows users to enter the Medicine Name, Date, and Time of the Day (Morning, Afternoon, Evening, or Night), store the details in an SQLite database, and automatically trigger an alarm at the scheduled time to display the medicine name as a reminder. 

2	A company wants to develop an Android-based meeting management system to help employees organize and access meeting schedules efficiently. Design two Android applications using a Content Provider: one application should allow users to enter and store the Meeting Date, Time, and Meeting Agenda in an SQLite database, while the second application should retrieve and display the Meeting Agenda for a selected date using a Date Picker. If no meeting is scheduled for the selected date, display a toast message stating "No Meeting on this Date.". 
3	A mobile communication company wants to develop an Android application that notifies users whenever they receive a new SMS. Design an application that displays a notification for incoming SMS messages, and when the user taps the notification, the sender's phone number and the message content should be displayed on the screen. Use the Android emulator to simulate sending an SMS to the application. 
4	A note-taking application allows users to create, edit, and manage text files on their device. Design an Android application with a TextBox and Create, Save, and Open buttons. The application should allow users to create a text file, save updates to the same file, and open previously stored content. If the user attempts to save the text before creating a file, display a toast message stating "First Create a File.". 

5	A music streaming company wants to develop a simple Android audio player that provides users with essential playback controls. Design an application that allows users to Play, Pause, Forward, and Backward an audio track. The application should also include a SeekBar that enables users to move to any position in the audio by dragging the progress indicator. 
6	A digital advertising company wants to display promotional messages as a scrolling banner in its Android application without affecting the app's responsiveness. Design an application that demonstrates the use of Asynchronous Tasks by scrolling the banner message "Demonstration of Asynchronous Task" from right to left when the Start Task button is pressed, and stopping the scrolling when the Stop Task button is pressed. 
7	A document editing application requires a feature that allows users to quickly copy and paste text between input fields. Design an Android application using the Clipboard Framework with two EditText controls and Copy and Paste buttons to demonstrate text copying and pasting functionality. 

8	A car dealership wants to develop an Android application that helps customers estimate their monthly car loan repayments before purchasing a vehicle. Design an application that uses an AIDL service to calculate the Car Loan EMI after deducting the down payment from the principal amount. The application should accept the Principal Amount, Down Payment, Interest Rate, and Loan Term (in months) as inputs, and display the calculated Monthly EMI when the "Calculate Monthly EMI" button is clicked. Additionally, evaluate how the EMI changes for different loan terms and interest rates.



ESTD:2001
An Institute with a Difference

COURSE OUTCOMES

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

CO1	Analyze the architecture and components of the Android operating system for mobile application development.
CO2	Develop Android applications using Android Studio by applying appropriate development tools and project structures.
CO3	Analyze the integration of user interface elements and layouts to enhance application usability.
CO4	Implement application navigation and communication using activities, intent
CO5	Execute the testing procedures on Android applications using emulators and real devices.

Descriptions:

1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/Kotlin to execute these programs. Diagrams given are for representational purposes only, students are expected to improvise on them.
3. Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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. Google Developer Training Team android Developer Fundamentals Course – Concept Reference, Google Developer Training, 1st Edition, 2017.
   `https://google-developer-training.github.io/...course-concepts/`
2. Erik Hellma android Programming – Pushing the Limits, Wiley India Pvt. Ltd., 1st Edition, 2014.
3. Pierre-Olivier Laurence, Amanda Hinchman-Dominguez, Mike Dunn, Programming Android with Kotlin, O'Reilly SPD Publishers, 1st Edition, 2021.

Reference Books:

1. Bill Phillips, Chris Stewart, Kristin Marsicanoandroid Programming: The Big Nerd Ranch Guide, Big Nerd Ranch Guides, 3rd Edition, 2017.
2. Ian F. Darwin android Cookbook: Problems and Solutions for Android Developers, O'Reilly Media, 2nd Edition, 2017.
3. Neil Smyth android Studio 4.2 Development Essentials – Java Edition, Payload Media, 1st Edition, 2021.

ESTD:2001
An Institute with a Difference

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

Course Title	ADVANCED JAVA		
Course Code	25BCSAE406C	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. 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 "RNSIT123" 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, Sessions and Spring Boot.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

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 Education, 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.

RNS
INSTITUTE OF TECHNOLOGY
ESTD:2001
An Institute with a Difference

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

Course Title	NMAP		
Course Code	25BCSAE406D	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 how to utilise the nmap scanner to discover live hosts and open ports.
2. Learn about various nmap port scans such as TCP, SYN, UDP, null, FIN and Xmas.
3. Learn advanced options of nmap such as fragmentation.
4. Learn OS and service detection and capabilities of nmap scripts.
5. Learn to save nmap results for future usage.

Program - 1
Host discovery: Perform live host discovery using ARP scan, ICMP scan and TCP/UDP scan in nmap. Example Perform live host discovery on a subnet to find live hosts.
Program - 2
Port scanning basics: Understand the basics of port scanning in nmap and recognize open, closed and filtered port states using nmap. Example: Find out the open and closed ports after discovering a live host.
Program - 3
Version detection: Use nmap to identify specific service software and version numbers. Example: Find out the service and version running on a specific port.
Program - 4
Timing and performance: Control scan speed using timing templates in nmap (-T0 to T5). Example: Perform slower scans to avoid raising alarm.
Program - 5
Output formats: Save results of nmap in normal, XML and grepable formats. Example: Save results of a massive scan using the above formats.

Program - 6
TCP connect scans: Use nmap to perform TCP connect scan using the full three-way handshake. Example: Perform a TCP scan using a non-root account and understand the results.
Program - 7
SYN scans: Perform stealthy half-open scanning using nmap.
Example: Perform a SYN scan using a root account to try and bypass application level logging.
Program - 8
UDP scans: Use nmap to scan for connectionless services like DNS and DHCP. Example: Run UDP scan to search for systems running hidden DNS service.
Program - 9
Specialized scans: Use nmap to perform Null, FIN and Xmas scans and how they interact with different OS network stacks. Example: Upon encountering a firewall that restricts all SYN packets, run the scans above to reveal open ports.
Program - 10
Firewall and IDS evasion: Use nmap facilities of packet fragmentation, MTU manipulation and source port modification to bypass security controls. Example: Use fragmentation to prevent a firewall from detecting the signature of a port scan.
Program - 11
OS detection: Use TCP/IP fingerprinting in nmap to determine target operating system. Example: Use unusual packets to analyze how an OS handles them, in the process fingerprinting the OS.
Program - 12
Nmap scripting engine (NSE): Understand the basics of scripting in nmap using Lua. Example: Use various scripts to check the vulerabilities of a target system.

COURSE OUTCOMES

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

CO1	Explain the purpose of Nmap in discovering live hosts and open ports.
CO2	Perform TCP, SYN, UDP, NULL, FIN and Xmas port scans using Nmap.
CO3	Apply advanced Nmap options such as packet fragmentation during network scanning.
CO4	Analyze operating system and service detection results obtained using Nmap scripts.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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.			

Reference Links:

1. `https://nmap.org/docs.html`
2. `https://tryhackme.com/module/nmap`

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

Course Title	GOOGLE COLAB		
Course Code	25BCSAE406E	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. Understand and utilize the features of Google Colab to write, execute and manage Python programs in a cloud-based environment.
2. Apply fundamental programming concepts to perform data processing tasks such as input handling, file operations and string manipulation.
3. Analyse and manipulate structured data using appropriate tools and techniques, including CSV handling and data filtering.
4. Visualize data using graphical techniques and interpret patterns to support decision-making.
5. Implement data pre-processing, basic predictive analysis and classification techniques to solve real-world problems.

Prerequisites

1. Basic knowledge of programming concepts (variables, loops, functions) using languages such as C, Python, or Java.
2. Fundamentals of database systems including basic SQL operations like SELECT, INSERT and filtering conditions.

Program 1 - Student Performance Analysis

A college maintains records of students' marks in multiple subjects. The administration wants to analyse overall performance to identify academic trends. Develop a program to accept marks of N students across multiple subjects and compute total and average marks for each student. Further, calculate the class average, identify the topper and list all students whose performance is below the class average.

Program 2 - Text Data Analysis

A company collects customer feedback in textual format and wants to extract meaningful insights. Write a program to read a given text file, process the contents and compute the frequency of each word. The system should also identify and display the most frequently occurring word and list words whose frequency exceeds a specified threshold.

Program 3 - CSV-Based Data Query System
A college stores student records in a CSV file instead of a traditional database. The management requires a system to perform query-like operations on this data. Develop a program to read the CSV file and implement functionalities similar to SQL operations such as filtering records based on conditions (e.g., marks greater than a specified value), sorting records in ascending or descending order and displaying a limited number of top records.
Program 4 - Data Visualization and Interpretation
An academic coordinator wants to visually analyze student performance for better decision-making. Using a given dataset, develop a program to generate different types of visualizations such as line graphs, bar charts and pie charts. Based on the generated graphs, provide a brief interpretation highlighting key trends and comparisons observed in the data.
Program 5 - Attendance Monitoring System
Maintaining attendance is crucial in academic institutions. Design a system that stores attendance details of students and calculates their attendance percentage. The program should identify students whose attendance falls below a specified threshold (e.g., 75%) and generate a report listing such students along with their attendance percentages.
Program 6 - Mini Database Management System
A small organization wants to manage employee records without using a full-fledged database system. Develop a program to simulate basic database operations such as inserting new records, deleting existing records, updating record details and searching for records based on a unique identifier. Ensure the system maintains data consistency during these operations.
Program 7 - Data Cleaning and Pre-processing
Real-world datasets often contain inconsistencies such as missing values and duplicate entries. Given a dataset with such issues, develop a program to identify missing values and replace them using appropriate statistical measures (mean or median). Additionally, remove duplicate records and normalize the data to ensure consistency for further analysis.
Program 8 - Correlation Analysis
An educational researcher wants to study the relationship between study hours and student performance. Develop a program to compute the correlation between different variables in a dataset. Visualize the relationship using appropriate plots and interpret whether the variables are positively correlated, negatively correlated, or independent.

Program 9 - Predictive Analysis
An institution aims to predict student performance based on study-related factors. Develop a program that analyses the relationship between input variables such as study hours, attendance, or internal assessment scores and the final performance of students. The program should use suitable techniques to make predictions for new inputs and provide meaningful insights about expected outcomes. Additionally, evaluate the effectiveness of the predictions by comparing them with actual values.
Program 10 - Classification of Student Performance
To automate result processing, a system is required to classify students based on their academic performance. Develop a classification model that categorizes students into groups such as "Pass" or "Fail" based on input features like marks and attendance. Train the model using sample data and test its performance on unseen inputs.

COURSE OUTCOMES

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

CO1	Apply fundamental Python programming concepts to develop solutions for data processing and manipulation.
CO2	Analyse structured and unstructured data using appropriate techniques such as file handling, data filtering and aggregation.
CO3	Develop data-driven applications by integrating data visualization and database-like operations.
CO4	Evaluate datasets through preprocessing, cleaning and interpretation to extract meaningful insights.
CO5	Develop basic predictive and classification models for decision making.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPyand Jupyter, 3rd Edition, O'Reilly Media, 2022.

Reference Books:

1. E. Balagurusamy, Programming in Python, 1st Edition, McGraw Hill Education, 2016.

Self Study topics:

1. Introduction to cloud-based development environments and comparison with traditional IDEs using Google Colab.
2. Working with different data file formats such as CSV, Exceland JSON for real-world applications.
3. Exploratory Data Analysis (EDA) techniques for understanding and summarizing datasets.
4. Data visualization techniques using advanced plots such as histogram, scatter plotand box plot.
5. Overview of machine learning concepts including supervised and unsupervised learning methods.

Activity Based Learning:

1. Dataset Exploration and Insight Generation: Students are provided with a real-world dataset and asked to explore its structure, identify important featuresand present key observations. They should summarize patterns, trendsand anomalies based on their analysis.
2. Data Cleaning and Pre-processing Activity: Students work on a dataset containing missing values and inconsistencies. They are required to clean the data by handling null values, removing duplicatesand preparing it for further analysis.
3. Visualization and Interpretation Task: Students create various visualizations such as bar charts, line graphs, or pie charts and interpret the results. They must explain trends and provide meaningful conclusions based on the visual output

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

Course Title	**ANACONDA NAVIGATOR**		
Course Code	**25BCSAE406F**	**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. Understand and navigate the Anaconda Navigator interface effectively.
2. Manage environments and packages efficiently using Conda.
3. Integrate and operate development tools such as Jupiter Notebook and Spyder IDE within Anaconda.
4. Ensure reproducibility and perform dependency management for data science and programming projects.

Program 1 - Installation and Initial Setup of Anaconda Navigator

Install Anaconda Distribution on the system and configure Anaconda Navigator. Verify installation paths, default Python version and startup settings.

Use Anaconda Navigator to launch Spyder and develop an application to enter marks, calculate total, percentage, grade and result status.

Program 2 - Exploring Anaconda Navigator Interface

Use Anaconda Navigator to explore the Home, Environments, Learning and Community tabs. Launch Spyder IDE and write a Python script to display the names of these interface sections.

Program 3 - Launching Integrated Applications

Use Anaconda Navigator to launch integrated applications such as Jupyter Notebook, Spyder IDE and JupyterLab. Open Spyder IDE.

Write a Python script to display the names of the applications launched through Anaconda Navigator.

Program 4 - Creating and Managing Virtual Environments

Use Anaconda Navigator to launch Spyder IDE.

Write a Python script to perform addition, subtraction, multiplication and division of two numbers and display the results.

Program 5 - Environment Management Using Conda Commands
Use Anaconda Navigator to create a new environment and launch Spyder IDE from that environment. Write a Python script to read a string from an input file, display the content in the Anaconda console and store the result in an output file.
Program 6 - Advanced Multi-Version Environment Setup
Use Anaconda Navigator to create multiple virtual environments with different Python versions and launch Spyder IDE from each environment. Write a Python script to display the Python version, environment name and current date/time in the Anaconda console for comparison.
Program 7 - Package Installation Using Navigator GUI
A data analyst needs libraries for numerical computation and visualization. Use Anaconda Navigator to install packages such as NumPy and Matplotlib using the Navigator GUI. Launch Spyder IDE. Write a Python script to create an array of student marks, calculate the average and display a bar chart of the marks in the Anaconda console/output window.
Program 8 - Package Management Using Conda and Pip
A startup developer needs to compare two package installation methods for a text processing project. Use Anaconda Navigator to open Anaconda Prompt and install one package using Conda and another using Pip. Launch Spyder IDE. Write a Python script to count the number of words and vowels in a given paragraph, then display the results in the Anaconda console.
Program 9 - Dependency Conflict Resolution
A software team faces errors due to incompatible library versions in a project environment. Use Anaconda Navigator to identify and resolve the dependency conflict by updating or modifying the required packages. After resolving the issue, launch Spyder IDE. Write a Python script to display the project title, number of team members, date of start, final execution status and names of people involved in sorted order in the Anaconda console.
Program 10 - Running Projects Across Multiple Environments
Create separate projects in Jupyter Notebook and Spyder under different environments. A student development team is working on projects that require different Python environments. Use Anaconda Navigator to create and run the projects across multiple environments with the required packages. After successful execution, launch Spyder IDE.
Write a Python script to display student name, USN, branch, project title, environment used and execution status in the Anaconda console. Notebook and IDE Configuration through Navigator, Environment Export, Import & Recovery: A healthcare analyst needs to maintain daily in-patient and out-patient records using properly configured development tools. Use Anaconda Navigator to configure Jupyter Notebook and Spyder IDE, then export, import and recover the project environment for reliable usage across systems. After setup, write a Python script to display the hospital name, date, number of in-patients, number of out-patients, total patients and status report in the Anaconda console.

COURSE OUTCOMES

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

CO1	Explain the architecture, features, and applications of Anaconda Navigator.
CO2	Apply Anaconda Navigator to create and manage Python development environments.
CO3	Demonstrate the use of Anaconda tools for developing, executing, and debugging Python applications.
CO4	Analyze and resolve issues related to package management and environment configuration.
CO5	Evaluate and maintain reproducible Anaconda environments for collaborative software development.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	**Maximum Marks**	**Minimum Passing Marks**	**Evaluation Details**
CIE – 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 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. Official Anaconda Documentation, Anaconda Inc. `https://docs.anaconda.com/`
2. Wes McKinney, Python for Data Analysis, O'Reilly Media, 3rd Edition, 2022.

Reference Books:

1. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 1st Edition, 2016.
2. Charles R. Severance, Python for Everybody: Exploring Data Using Python 3, CreateSpace Independent Publishing Platform, 1st Edition, 2016.

Web Links:

1. `https://www.anaconda.com/products/distribution`

2. https://docs.anaconda.com/navigator/
3. https://docs.conda.io/projects/conda/en/latest/
4. Anaconda Official Channel
5. Corey Schafer – Python Environment Tutorials
6. freeCodeCamp – Anaconda & Jupyter Tutorials
7. Simplilearn / Edureka – Conda Package Management

RNS
INSTITUTE OF TECHNOLOGY
ESTD : 2001
An Institute with a Difference

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

Course Title	TRYHACKME		
Course Code	25BCSAE406G	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. Understand fundamental cyber security concepts including reconnaissance, CIA triad, vulnerabilities, authentication and system defense mechanisms.
2. Apply security tools and techniques such as Nmap, Hydra, Wireshark, log analyzers, firewall utilities and TryHackMe labs for practical cyber security learning.
3. Analyze common cyber-attacks including password attacks, privilege escalation, malware persistence, SQL injection and network scanning.
4. Investigate security incidents using logs, system monitoring and attack indicators to identify threats and recommend mitigation measures.
5. Design and implement secure system configurations, layered defense strategies and best practices for protecting computing environments.

Program 1 - Think Like an Attacker (Reconnaissance)
Perform reconnaissance on a Windows/Linux system using ipconfig, ifconfig, whoami and netstat. Compare the outputs of both operating systems and identify which information is most useful for an attacker. Based on discovered open ports and users, recommend security hardening controls.
Program 2 - Why Security is a Concern (CIA Triad)
Create a confidential file, change permissions, modify contents and stop a running web service. Explain how each action affects confidentiality, integrity and availability. Relate the same concept to a ransomware attack scenario.
Program 3 - Vulnerability Identification
Scan a machine using nmap localhost, identify open services, gather OS details using systeminfo or uname and map possible vulnerabilities for discovered services. Recommend a patching strategy.
Program 4 - Password Attack Simulation
Simulate a brute-force login attack using Hydra against SSH. Compare brute-force, dictionary and credential stuffing attacks. Design a secure authentication policy for organizations.
Program 5 - Privilege Escalation
Use sudo -l to identify privileged commands available to a user. Explain how attackers escalate privileges using misconfigured sudo rights. Recommend controls to prevent privilege abuse.
Program 6 - Malware Simulation (Persistence)
Create a cron-based startup task that runs on reboot. Compare Linux cron persistence with Windows registry startup persistence. Explain how defenders detect and remove persistence mechanisms.
Program 7 - Network Attack (Scanning & Sniffing)

Use Nmap to scan a target machine and Wireshark to capture packets. Analyze captured HTTP traffic for sensitive information and recommend secure network controls.
Program 8 - Web Attack (SQL Injection)
Test a vulnerable login form using SQL injection payloads. Compare error-based, union-based and blind SQL injection methods. Recommend secure coding practices to prevent database compromise.
Program 9 - Log Analysis & Detection
Analyze /var/log/auth.log for brute-force attempts, suspicious login timings and repeated failed logins. Explain how SIEM tools correlate such events and suggest response actions.
Program 10 - Defense & Mitigation
Harden a Linux server using firewall rules, package updates and secure SSH configuration. Compare preventive, detective and corrective controls. Prepare a baseline security checklist.

COURSE OUTCOMES

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

CO1	Apply cyber security tools to perform reconnaissance, vulnerability assessment and network analysis.
CO2	Analyze cyber threats such as brute-force attacks, malware persistence, privilege escalation and web application attacks.
CO3	Apply defensive mechanisms including firewalls, authentication controls, system hardening and secure configurations.
CO4	Investigate logs and security events to detect attacks and recommend corrective actions.
CO5	Develop practical cyber security skills through hands-on labs using TryHackMe and attack–defense simulations.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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. Dafydd Stuttard and Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd Edition, John Wiley & Sons, Inc., 2011.
2. Gordon Fyodor Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, 1st Edition, Insecure.Com LLC, 2009.

Reference Books

1. Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008.
2. OccupyTheWeb, Linux Basics for Hackers: Getting Started with Networking, Scriptingand Security in Kali, 1st Edition, No Starch Press, 2018.
3. Don Murdoch, Blue Team Handbook: Incident Response Edition, 1st Edition, CreateSpace Independent Publishing Platform, 2014.
4. David Kennedy, Jim O'Gorman, Devon Kearnsand Mati Aharoni, Metasploit: The Penetration Tester's Guide, 1st Edition, No Starch Press, 2011.

Web Links:

1. TryHackMe Learning Platform
   Covers Nmap, SQLi, privilege escalation, Hydra, SOC labs.
2. OWASP SQL Injection Guide
   Useful for Experiment 8 (Web Attacks).

3. OWASP Top 10 2025 Injection Category
   Useful for secure coding and web vulnerabilities.

4. OWASP Top 10 2021 Injection
   Security awareness reference.

RNS
INSTITUTE OF TECHNOLOGY
ESTD:2001
An Institute with a Difference

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

Course Title	WIRESHARK		
Course Code	25BCSAE406H	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. Capture and analyze real-time network traffic using Wireshark.
2. Understand the structure and behavior of network protocols through packet-level inspection.
3. Apply filters to extract meaningful information from captured packets.
4. Analyze application layer protocols such as HTTP, DNS and FTP.
5. Perform basic network troubleshooting and identify security threats.

Program 1 - Scenario: Network Traffic Capture and Interface Analysis
A network administrator needs to monitor real-time traffic and understand packet structure for analysis. Install Wireshark and perform live packet capture. Save the captured packets in .pcap format. Study Wireshark interface and identify packet list pane, packet details pane and packet bytes pane.
Program 2 - Scenario: Website Connectivity Issue Analysis
A user reports that a website is not loading, indicating a possible issue in DNS resolution or TCP connection establishment. Capture traffic using Wireshark. Analyze DNS query/response, SYN, SYN-ACK, ACK packets. Identify DNS resolution is happening and TCP handshake is successful. Apply filter and check for missing responses and error codes.
Program 3 - Scenario: TCP Connection Behavior Analysis
A network engineer needs to analyze the communication between a client and server to understand connection establishment and termination. Identify: 3-way handshake
Connection termination
Extract sequence numbers
Program 4 - Scenario: HTTP Communication Analysis
A user accesses a website and there is a need to examine HTTP request and response behavior for proper data exchange. Capture HTTP packets. Identify: Request method (GET/POST) Response status code
Program 5 - Scenario: Detection of High Traffic / Potential DDoS Activity

A system experiences unusually high traffic from a single source, indicating a possible DDoS attack or abnormal network behavior. Apply filter for ip address, identify packet rate and repeated requests.
Program 6 - Scenario: Detection of Unencrypted Data Exposure
A user reports that sensitive information may be exposed during network communication. Task: Check HTTP packets: Look for Plain text credentials and Unencrypted data. Compare secure vs non-secure traffic (HTTP vs HTTPS).
Program 7 - Scenario: ARP Spoofing Detection and Connectivity Issue Analysis
A network user reports intermittent connectivity issues and suspects possible malicious activity within the local network. Apply filters and identify Multiple MACs for same IP.
Program 8 - Scenario: File Transfer and Download Analysis
A file has been downloaded from a remote server and there is a need to analyze the communication and data transfer characteristics. Identify: Server IP Protocol used Packet size
Program 9 - Scenario: Network Performance and Retransmission Analysis
The network shows signs of slow performance, requiring analysis of retransmissions and packet loss. Capture network traffic while downloading a large file or streaming video.
Apply filter: tcp.analysis.retransmission
Identify: Retransmitted packets Duplicate ACKs Packet loss indicators
Program 10 - Scenario: Suspicious Traffic / Attack Pattern Analysis
Unusual traffic patterns are observed in the network, indicating potential malicious activity such as SYN flood or repeated requests. Capture network traffic while downloading a large file or streaming video. Apply filter: tcp.flags.syn == 1 and tcp.flags.ack == 0 Analyze: High number of SYN packets (possible SYN flood) Repeated requests from a single IP Unusual traffic spikes

COURSE OUTCOMES

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

CO1	Analyze network packets captured using Wireshark across different protocol layers.
CO2	Diagnose network connectivity issues using DNS resolution, TCP handshake and HTTP communication analysis.
CO3	Apply filtering techniques to isolate and examine specific network traffic.
CO4	Analyze network behavior by identifying retransmissions, packet loss and data transfer characteristics.
CO5	Detect network security threats such as sniffing, ARP spoofing and DDoS/suspicious traffic patterns.

RNS
INSTITUTE OF TECHNOLOGY
ESTD:2001
An Institute with a Difference

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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. Vinit Jain, Wireshark Fundamentals: A Network Engineer's Handbook to Analyzing Network Traffic, 1st Edition, Apress, 2022.

Reference Link:

1. `https://www.wireshark.org/docs/wsug_html_chunked/`

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

Course Title	TABLEAU PUBLIC		
Course Code	25BCSAE406I	CIE MARKS	50
(L : T : P)	(0:0:2)	SEE MARKS	50
Credits	1	EXAM HOURS	3

COURSE LEARNING OBJECTIVES

The purpose of this course is to enable the learner to:

1. Apply Python programming and libraries (NumPy, Pandas) to perform data manipulation, array operations and matrix computations.
2. Analyze and process datasets by performing data cleaning, transformation and statistical computations on structured data.
3. Visualize data effectively using tools like Matplotlib to represent trends, patterns and relationships through various charts.
4. Utilize data analytics and visualization tools (e.g., Tableau) to explore real-world datasets and generate meaningful insights.

Program 1 - NumPy Array Operations and Statistical Analysis

You are working as a data analyst at TrendMart Retail Pvt. Ltd., a company that monitors daily sales performance to make business decisions. The management has provided you with a dataset containing daily sales figures for a week and expects a detailed analysis to understand sales trends and patterns.

i) Develop a Python program to analyze the given sales data by performing the following tasks:
   a) Calculate the square and square root of each sales value to study growth patterns
   b) Raise each value to a specified power to simulate projected growth
   c) Compute the mean and median to understand average performance
   d) Determine the total sales, minimum and maximum values
   e) Calculate the variance and standard deviation to measure consistency in sales
   f) Sort the sales data in ascending or descending order for reporting
   g) Identify unique sales values to detect duplicates or anomalies

ii) As part of deeper analysis, you are required to extract meaningful insights from specific portions of the dataset. Develop a Python program to:
   i) Generate an array (list) of sales data
   ii) Demonstrate indexing and slicing techniques to retrieve specific subsets of data (e.g., first three days, last two days, alternate days)

Program 2 - Matrix and Multi-Dimensional Array Operations using NumPy

You are working as a data analyst in a scientific research organization, where experimental data is often represented in the form of matrices and multi-dimensional arrays.

i) You are provided with a dataset representing measurements collected from different experiments, stored in a 2-D matrix. Develop a Python program to analyze and manipulate this data by performing the following tasks:
   a) Determine the number of elements in each dimension to understand the dataset size
   b) Identify the data type of the matrix for proper processing
   c) Determine the dimensions of the matrix
   d) Compute the inverse and transpose of the matrix for further mathematical analysis
   e) Calculate the eigenvalues and determinant to study matrix properties
   f) Generate an identity matrix for comparison and validation
   g) Create matrices filled with ones and zeros for initialization purposes
   h) Demonstrate the use of the empty() method for memory allocation
   i) Convert the matrix into float type for precise computations

ii) In another experiment, the data is stored as a 3-D array representing multiple layers of observations over time. Develop a Python program to:
   i) Extract the inner matrix (specific layer) for detailed analysis
   ii) Sort the elements row-wise and column-wise to organize the data for reporting.

Program 3 - DataFrame Manipulation and Data Analysis using Pandas

You are working as a data analyst in a retail company that maintains large datasets of sales and customer transactions stored in CSV files. Your task is to analyze and preprocess the data for generating business insights. Develop a Python program that performs the following operations:

i) Read the given CSV file and convert it into a DataFrame for analysis
ii) Compute the summary statistics to understand overall data distribution
iii) Reshape the DataFrame to organize the data in a more meaningful structure
iv) Use the pivot_table() method to generate summarized insights (e.g., sales by category or region)
v) Identify duplicate records based on specific column labels, excluding the last occurrence, to clean the dataset.
vi) Replace values in the DataFrame based on given conditions or labels to correct inconsistencies.
vii) Merge multiple DataFrames based on a common column (e.g., customer ID or product ID) to consolidate information.

Program 4 - DataFrame Modification and Record Management using Pandas

You are working as a data analyst at MediCare Health Services, a company that manages patient and hospital records stored in CSV files. Your responsibility is to maintain and update the dataset for accurate reporting and analysis. Develop a Python program to perform the following operations:

i) Read the given CSV file and convert it into a DataFrame for analysis
ii) Compute the summary statistics to understand patient data trends and distributions
iii) Insert a new row and a new column to include additional patient details (e.g., new admission record, health parameter)
iv) Delete an existing row and a column to remove outdated or incorrect records.

Program 5 - Line Chart Visualization using Matplotlib

The following information is given that contains yearly sales of Smart phones in two Companies for the past six years.

Year	Sales_Company_A	Sales_Company_B
2018	650	780
2019	500	400
2020	400	550
2021	800	650
2022	880	850
2023	900	1000

i) Develop a Python program that utilizes the Matplotlib library to visualize the yearly sales trends of both companies using Line plots
ii) Display the values on the graph and save the chart in jpg format.

Program 6 - Bar Chart Visualization using Matplotlib

You are working as a data analyst at ReadMore Book Exhibitions, a company that organizes large-scale book fairs and tracks weekly sales data. The management wants to compare the performance of book sales across different weeks to understand customer trends and improve future planning. You are provided with a dataset in CSV format that contains the number of book copies sold during different weeks of the fair. Develop a Python program that:

i) Reads the given CSV dataset
ii) Uses the Matplotlib library to create a bar chart
iii) Compares the number of book copies sold in Week 1, Week 2and Week 5
iv) Clearly labels the chart for better interpretation
v) Saves the generated chart in JPG format for reporting purposes

Program 7 - Donut Chart Visualization using Matplotlib

The following information is given that contains Tax details in a Budget.

Income Form	Amount (in Paise)
Corporation Tax	21
Income Tax	16
Customs	4
Union Excise Duties	8
GST & other Taxes	19

Income Form	Amount (in Paise)
Non-Tax Revenue	9
Non-Debt Capital Receipts	3
Borrowings and other Liabilities	20

Develop a Python program that utilizes the Matplotlib library to create a Donut Chart to visualize the distribution of Income tax across different categories.

Program 8 - Scatter Plot Visualization using Matplotlib

You are working as a data analyst at BrightFuture Analytics, an educational analytics company that helps schools evaluate student performance. The company has provided you with a dataset in CSV format containing students' scores in five subjects: English, Math, Science, Social Science and Computer Science. Your task is to analyze the relationship between students' performance across different subjects. Develop a Python program that:

i) Reads the given CSV dataset and converts it into a suitable data structure.
ii) Uses the Matplotlib library to create a scatter plo.t
iii) Visualizes the relationship between the students' scores in any three selected subjects.
iv) Helps identify patterns, correlations, or trends in student performance.

Program 9 - Data Visualization using Tableau – Basic Charts

As a data analyst at a retail company, you are interested in comparing the Sales and Profit performance across different Regions and Subcategories in the Sample-Superstore dataset. Use Tableau Visualization tool to perform the following operations on the given data set.

i) Create a Column chart to find the sum of Sales for each Region and Category. Sort the details in ascending order of Sales.
ii) Create a Horizontal Bar chart to find the Profit of each Sub-Category and sort them in descending order of Profit.
iii) Create a Tree map to find the Sub-Category that has the highest Sales.
iv) Create a Scatter plot to find the Customers with the highest Profit and Sales.

Program 10 - Advanced Data Visualization using Tableau
As a data analyst at a retail company, you are tasked to analyze the Sales, Profit and Quantity distribution across different regions for each subcategory and segment in the Sample-Superstore dataset. Use Tableau Visualization tool to perform the following operations on the given dataset. i) Create a Pie chart to visually represent the proportion of Sales contributed by each Region and sort the details in descending order of Sales. ii) Create a Bubble chart to find the Sub-Category with the highest and the lowest Quantity. iii) Create a Heat map to represent the Profit in each Segment for each Sub-Category and sort the details in descending order of Profit. iv) Create a Box plot to identify the Technology that has the highest Profit.
Program 11 - Sales and Profit Analysis of Coffee Dataset using Tableau
You are provided with Coffee dataset which contains details about Coffee Sales and Profit. Use Tableau Visualization tool to perform the following operations on the given dataset. i) Display the top 5 States (each in different color) having the highest Sales ii) Display the Espresso Products in descending order of Sales iii) In California, display the Product with the highest and lowest Profit iv) Create a Heat map for Product Type, State and Profit. Identify which State has the lowest Profit for Espresso.
Program 12 - Interactive Data Analysis and Geographic Visualization using Tableau
You are provided with Coffee dataset which contains details about Coffee Sales and Profit. Use Tableau Visualization tool to perform the following operations on the given dataset. i) Display the States having the Sales between 50,000$ and 80,000$ ii) Determine the Average Profit for all the Products starting with "C" iii) In 2013, determine the State with the highest Profit in the West market iv) Display the bottom 5 States using a Map having the lowest Profit.

COURSE OUTCOMES

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

CO1	Apply NumPy array operations and indexing techniques for efficient numerical computations.
CO2	Analyze Pandas data structures to perform data selection, alignment and transformation.
CO3	Create data visualizations using Matplotlib and Pandas to present meaningful insights.
CO4	Apply Tableau functionalities to connect data sources, perform aggregation and visualize data using charts, maps and filters.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPyand Jupyter, 2nd Edition, O'Reilly Media, 2017.
2. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, 1st Edition, O'Reilly Media, 2016.
3. Phuong Vo T.H., Martin Czygan, Ashish Kumarand Kirthi Raman, Python: Data Analytics and Visualization, 1st Edition, Packt Publishing, 2017.
4. Davy Cielen, Arno D. B. Meysmanand Mohamed Ali, Introducing Data Science: Big Data, Machine Learningand More, Using Python Tools, 1st Edition, Manning Publications, 2016.

Web Links:

1. https://www.mygreatlearning.com/.../tableau-for-beginners
2. https://www.mygreatlearning.com/.../data-visualization-using-tableau

MOOCs:

1. https://www.coursera.org/.../data-analysis-visualization-foundation
2. https://www.coursera.org/learn/data-visualization

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

Course Title	ORANGE DATA MINING		
Course Code	25BCSAE406J	CIE MARKS	50
(L : T : P)	(0:0:2)	SEE MARKS	50
Credits	1	EXAM HOURS	3

COURSE LEARNING OBJECTIVES

The purpose of this course is to enable the learner to:

1. Understand the data mining concepts and techniques using a visual tool.
2. To enable students to perform data preprocessing and cleaning.
3. Develop skills in data visualization and exploration.
4. Apply classification and clustering algorithms on a given dataset.
5. Gain knowledge of association rule mining and outlier detection.

PART A

Program 1 - Exploratory Data Analysis using Orange Data Mining

A data analyst is assigned to explore a newly acquired dataset using the Orange Data Mining tool. The analyst must load the dataset using appropriate file handling components. The data should be displayed in a structured tabular format for initial inspection. Further, the analyst is required to visualize relationships between attributes using scatter plots to identify patterns and correlations. The overall objective is to perform an initial exploratory data analysis in a clear and structured manner.

Program 2 - Data Preprocessing and Normalization using Orange

A data engineer is tasked with preparing raw data for analysis using the Orange Data Mining tool. The dataset contains irrelevant attributes and missing values. The engineer must select only the necessary features, handle missing data appropriately and normalize the dataset to ensure consistency. The preprocessing steps should improve data quality and make it suitable for further analysis.

Program 3 - Statistical Data Visualization using Histograms and Box Plots in Orange

A data scientist is assigned to analyze the distribution and characteristics of a dataset using visualization techniques in Orange. The task involves generating histograms to study frequency distribution and box plots to examine median, spread and outliers. These visualizations help in understanding the statistical properties of the data. The system should allow clear interpretation of patterns and variability. The objective is to extract meaningful insights before applying machine learning algorithms.

Program 4 - Statistical Data Visualization using Histograms and Box Plots in Orange
A machine learning engineer is required to improve model efficiency through feature selection using the Orange Data Mining tool. The task involves ranking attributes based on their importance using appropriate techniques. The engineer must then select the most relevant features for further analysis. This process helps in reducing dimensionality and computational complexity. The objective is to enhance model performance and accuracy.
Program 5 - Decision Tree Classification and Performance Evaluation using Orange
A data analyst is assigned to build a predictive classification model using the Decision Tree algorithm in Orange. The dataset must be used to train the model effectively. The analyst should evaluate model performance using the Test & Score widget. The results must be analyzed to determine the accuracy and reliability of the model. The objective is to develop an interpretable and efficient classification system.
Program 6 - Naïve Bayes Classification and Model Comparison using Orange
A machine learning practitioner is tasked with implementing a Naïve Bayes classification model using the Orange Data Mining tool. The model must be trained on the dataset and evaluated using appropriate validation techniques. The practitioner should compare its performance with the Decision Tree model. This comparison helps in identifying the most suitable algorithm for the dataset. The objective is to ensure accurate and efficient classification.
Program 7 - Naïve Bayes Classification and Model Comparison using Orange
A data scientist is assigned to perform clustering analysis on an unlabeled dataset using the K-Means algorithm in Orange. The task involves grouping similar data points into clusters based on their characteristics. The clusters must be visualized using scatter plots for better interpretation. The system should clearly represent relationships among data points. The objective is to identify hidden patterns and groupings in the dataset.
Program 8 - Hierarchical Clustering and Dendrogram Analysis using Orange
A researcher is required to perform hierarchical clustering using the Orange Data Mining tool. The task involves generating a dendrogram to represent the hierarchical relationships among data points. The researcher must analyze cluster formations at different levels. This helps in understanding the structure and grouping of the dataset. The objective is to study data relationships in a multi-level manner.

Program 9 - Association Rule Mining using Orange Data Mining
A business analyst is tasked with discovering relationships within a transactional dataset using association rule mining in Orange. The task involves identifying frequent item sets and generating association rules. These rules must be interpreted to understand relationships between items. The analysis supports better decision-making and strategy formulation. The objective is to extract meaningful patterns from the dataset.
Program 10 - Outlier Detection and Analysis using Orange Data Mining
A data analyst is assigned to detect anomalies in a dataset using the Outliers widget in Orange. The task involves identifying data points that deviate significantly from normal patterns. These outliers must be analyzed to determine possible errors or rare events. The system should clearly highlight abnormal data points. The objective is to improve data quality and gain insights into unusual behaviors.

COURSE OUTCOMES

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

CO1	Apply data preprocessing techniques such as attribute selection, handling missing values and normalization.
CO2	Interpret data using visualization techniques such as Scatter Plot, Histogram and Box Plot.
CO3	Apply classification algorithms such as Decision Tree and Naive Bayes to build predictive models.
CO4	Analyze data using K-Means and Hierarchical Clustering techniques.
CO5	Evaluate data mining results using techniques such as feature selection, association rule mining and outlier detection.

ASSESSMENT AND EVALUATION REGULATIONS OF AEC

Assessment Type	Maximum Marks	Minimum Passing Marks	Evaluation Details
CIE – 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 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.			

Reference Links:

1. `https://orangedatamining.com/`
2. YouTube Tutorials – Orange Data Mining for Beginners
3. `https://www.youtube.com/watch?...HXjnDIgGDuI`
4. `https://www.youtube.com/watch?...2w9kJXGyiXk`
5. `https://www.youtube.com/watch?v=WBAA2l0HwXI`
6. `https://www.youtube.com/watch?v=ESAUErq0t7U`
7. `https://www.youtube.com/watch?v=f4QhPmWNzP0`
8. `https://www.youtube.com/watch?v=dJ5z2SRwzgs`
9. `http://youtube.com/watch?v=uzwqkEUEatk`
10. `https://orangedatamining.com/.../visualizations-101/?s=09`
11. `https://orangedatamining.com/.../interactive-data-visualization/`

B.E. (Computer Science and Engineering)
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)+(30 Hours/Sem)**	**SEE MARKS**	**50**
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 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. Biomolecules: 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, Tissue engineering, stem cells and their applications, Bioprinting techniques and materials. Electrical tongue and electrical nose in food science, DNA origami and Biocomputing, Bioimaging and Artificial Intelligence for disease diagnosis. Bioconcrete.	

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. adaptation.
CO3	Evaluate the principles of design and development, for exploring engineering knowledge in biological.
CO4	Explore innovative biobased solutions for socially relevant problems.

ASSESSMENT AND EVALUATION REGULATIONS OF BSC

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	-	Any two Assessment methods as per 24RNBE4.2 scaled to 20 marks.
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 before 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 (40%) out of alloated 50 marks to apper for SEE.			

Textbooks:

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

Reference Books:

1. Stuart Fox and Krista Rompolski, Human Physiology, 16th Edition, McGraw-Hill Education, 2022.
2. S. Thyagarajan, N. Selvamurugan, M. P. Rajesh, R. A. Nazeer, W. Thilagaraj, S. Barathi- and M. K. Jaganthan, Biology for Engineers, 1st Edition, Tata McGraw-Hill, 2012.
3. Arthur T. Johnson, Biology for Engineers, 1st Edition, CRC Press, 2011.
4. Leslie Cromwell, Biomedical Instrumentation and Measurements, 2nd Edition, Prentice Hall, 2011.
5. Sohini Singh and Tanu Allen, Biology for Engineers, 1st Edition, Vayu Education of India, 2014.
6. Yoseph Bar-Cohen, Biomimetics: Nature-Based Innovation, 1st Edition, CRC Press, 2012.
7. Dario Floreano and Claudio Mattiussi, Bio-Inspired Artificial Intelligence: Theories, Methodsand Technologies, 1st Edition, MIT Press, 2008.
8. C. R. Sunilkumar, N. Geethaand A. C. Udayashankar, Bioremediation of Heavy Metals: Bacterial Participation, 1st Edition, Lambert Academic Publishing, 2019.

B.E. (Computer Science and Engineering)
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	–
Credits	–	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 in 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 AND EVALUATION REGULATIONS OF NCMC

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 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.			

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.

Web Links:

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

B.E. (Computer Science and Engineering)
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 - Track Events - Relays **3 Hours**

Relays

1. Relay Race: Starting, Baton Holding/Carrying, Baton Exchange in between zoneand 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 zoneand finishing, along with basic skills of javelin throw.
CO2	Apply the fundamental techniques of triple jump, including approach run, take-off, hop, step, jump, flightand landing.
CO3	Develop and perform javelin throw techniques, including grip, carry, impulse stride, releaseand recovery with proper coordination.
CO4	Exhibit fundamental skills of cricket, including batting, bowling, fielding, wicket keepingand understand rules and duties of officials.
CO5	Demonstrate basic skills and techniques in baseball, including player stances, grip, batting, pitchingand understand rules and officiating.

ASSESSMENT AND EVALUATION REGULATIONS OF NCMC

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. (Computer Science and Engineering)
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 and Spiritual objectives of Yoga.
2. Explain the meaning of Yoga.
3. Discuss the importance of 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

a) Suryanamaskar prayer.

b) Meaning, need & Importance

c) Benefits of Suryanamaskara

Module-2: Title: Asanas	3 Hours

Need & importance of different Asana, Asana its meaning by name, technique, precautionary 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	Explain the meaning, objectives, scopeand importance of Yoga, including its physical, mentaland spiritual benefits and misconceptions.
CO2	Demonstrate Suryanamaskara (12 counts, 8 rounds) with correct technique, prayerand understanding of its need and benefits.
CO3	Perform sitting and standing Asanas (Bakasana, Hanumanasana, Yogamudra in Vajrasana, Vatayanasana, Garudasana) with proper technique, precautionsand benefits.
CO4	Execute prone, supineand balancing Asanas (Bhujangasana, Shalabhasana, Utthita Dvipadasana, Ardha Halasana, Sheershasana) with correct body alignment and safety measures.
CO5	Practice Pranayama techniques (Kapalabhati, Bhastrika, Bhramari) with proper method, breathing controland understanding of their importance and health benefits.

ASSESSMENT AND EVALUATION REGULATIONS OF NCMC

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. 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 by T.K.V. Desikachar.
2. Asana Pranayama Mudra Bandha by Swami Satyananda Saraswati.

RNS INSTITUTE OF TECHNOLOGY
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	**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 AND EVALUATION REGULATIONS OF NCMC

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.			

Textbooks:

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.

RNS
INSTITUTE OF TECHNOLOGY
ESTD:2001
An Institute with a Difference

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

Course Title	**ADDITIONAL MATHEMATICS-II**		
Course Code	**25BMADP410**	**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-order and higher-order ordinary differential equations using appropriate analytical methods in engineering applications.
3. Develop the knowledge and problem-solving skills required to evaluate multiple integrals and apply the 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 a matrix by elementary row operations, echelon form, consistency of systems of linear equations, Gauss elimination method, eigenvalues and eigenvectors of a square matrixand problems.
Module-2
Numerical Methods: Finite differences, interpolation/extrapolation using Newton's forward and backward difference formulae (statements only), solution of polynomial and transcendental equations using Newton–Raphson and Regula–Falsi methods (formulae only) with illustrative examples, numerical integration using Simpson's one-third rule (without proof)and 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 over rectangular and general regionsand change of order of integration.
Module-5
Triple Integrals: Definition, evaluation over simple regionsand applications of triple integrals to the evaluation of volumes.

COURSE OUTCOMES

At the end of 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 and evaluate volumes of geometrical regions.

ASSESSMENT AND EVALUATION REGULATIONS OF NCMC

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 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.			

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 Publishers, 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, for lateral entry Diploma holders admitted to the III semester of B.E./B.Tech. programmes. Students 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 a student fails to register for the course or fails to secure the minimum 40% of the prescribed CIE marks, he/she shall be deemed to have secured an F grade. In such cases, the student shall fulfill the course requirements during subsequent semesters to earn the qualifying CIE marks. These courses are prescribed for CIE only and have no Semester End Examination (SEE).
2. Additional Mathematics I and II shall not be considered for vertical progression or for the calculation of SGPA and CGPA. However, successful completion of these courses is mandatory for the award of the degree.
3. Successful completion of Additional Mathematics I and II shall be indicated as Satisfactory in the grade card. Non-completion of these courses shall be indicated as Unsatisfactory.