B.N.M. Institute of Technology

An Autonomous Institution under VTU

Dept. of Computer Science and Engineering

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

Semester: III

Course: Fourier Transform, Fundamentals of logic and Linear Algebra

Course Code: 23MAC131 (Common to CSE, ISE, AIML)

L:T:P:J	**2:2:0:0**	**CIA**	**: 50**
Credits:	**03**	**SEA**	**: 50**
Hours:	**40**	**SEA Duration**	**: 03 Hours**

Course Learning Objectives: The students will be able to

1 Have an insight into Fourier series, Fourier transforms.

2 Develop knowledge of Fundamentals of logic and Relations, Vector Spaces & Linear Transformation arising in engineering

Module-1: Fourier Series & Fourier Transforms	**No. of hours**	**BLL, CO**
Examples from Engineering that require Fourier series and Fourier Transforms. **Fourier series**: Periodic functions, Introduction to Fourier Series, Dirichlet's condition. Problems on Fourier series over $(-l, l)$. **Fourier Transforms:** Introduction to infinite Fourier transform, Fourier sine and cosine transform and properties, problems on infinite Fourier transform, Discrete & Fast Fourier transform. *Experiential Learning component: Finding the Fourier series and Fourier Transform of a function*	**L : 04** **T : 04**	**Apply**
Module-2: Fundamentals of logic and Relations		
Examples from Engineering that require Fundamentals of logic and Relations. **Fundamentals of logic:** Basic connectives and truth tables, logic equivalence - the laws of logic, logical implication- rules of inference **Relations:** First order linear recurrence relation, second order linear homogenous recurrence relation with constant coefficients. *Experiential Learning component: Finding the solution of recurrence relation*	**L : 04** **T : 04**	**Apply**
Module-3: Vector Spaces		
Examples from Engineering that require vector spaces Recap of system of linear homogenous and non-homogeneous equation and solution sets. Vector spaces, subspaces, linearly independent and dependent, Linear span of a set, Basis and dimension, coordinate vectors. *Experiential Learning component: Problems on linearly independent and dependent vectors, basis and dimension of a vector space.*	**L : 04** **T : 04**	**Apply**
Module-4:Linear Transformation		
Examples from Engineering that require linear transformation. Linear transformations, algebra of linear transformations, representation of transformations by matrices, Non-singular linear transformation, Inverse of a linear transformation, Range space, Null space and problems on Rank-nullity theorem. *Experiential Learning component: Problems on Inverse of a linear transformation and Rank-nullity theorem*	**L : 04** **T : 04**	**Apply**
Module-5: Inner Product Spaces		

Examples from Engineering that require Inner product spaces. Inner products Inner product spaces, Orthogonal set, orthogonal projections, orthonormal bases, Gram-Schmidt process, QR-factorization, Recap of Eigen values and Eigen vectors, problems on Singular value decomposition. *Experiential Learning component: Problems on QR-factorization and singular value decomposition*	**L : 04** **T : 04**	**Apply**

Course Outcomes: After completing the course, the students will be able to

CO 1: Apply Fourier series & transform concepts in data visualization and cryptography.

CO 2: Communicate the basic concepts of logic and their relevance for computer science engineering.

CO 3: Apply the knowledge of vector spaces for solving problems in arising in engineering field

CO 4: Apply the knowledge of linear transform for solving problems in arising in image processing

CO 5: Compute orthogonal and orthonormal bases vectors and decomposition of a symmetric matrix using standard technique.

Reference Books:

1. E. Kreyszig: "Advanced Engineering Mathematics", John Wiley & Sons,10th Edition(Reprint), 2016.
2. B. S. Grewal: "Higher Engineering Mathematics", Khanna Publishers, 44th Ed., 2017.
3. H. K. Dass, " Advanced Engineering Mathematics" S. Chand publication.
4. C. Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6" Edition, 2. McGraw-Hill Book Co., New York, 1995.
5. James Stewart : "Calculus —Early Transcendentals", Cengage Learning India Private Ltd., 2017.
6. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
7. Srimanta Pal & Subodh C Bhunia: "Engineering Mathematics", Oxford University Press, 3"Reprint, 2016.
8. David C. Lay, Steven R. Lay and J. J. McDonald "Linear Algebra and its applications", 3rd Edition, Pearson Education Ltd., 2017.
9. Ralph P. Grimaldi, " Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education 2004.

Web links and Video Lectures:

1. https://nptel.ac.in/courses/111106111
2. https://youtu.be/OynpZwylau8
3. https://archive.nptel.ac.in/courses/111/106/111106051/
4. https://www.youtube.com/watch?v=zvRdbPMEMUI
5. https://www.youtube.com/watch?v=cHNmT1-qurk
6. https://www.youtube.com/watch?v=ATqV_I8DCh0

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Mathematics

Semester: IV

Course: Statistics, Probability and Graph theory

Course Code: 23MAC141 (Common to CSE, ISE, AIML)

L:T:P:J	**2:2:0:0**	**CIA: 50**
Credits:	**03**	**SEA: 50**
Hours:	**40**	**SEA Duration: 03 Hours**

Course Learning Objectives: The students will be able to

1	Provide an insight into applications of Graph Theory, Curve fitting & Statistical methods.
2	Develop the knowledge of probability, joint probability distribution and Queuing theory occurring in digital signal processing, design engineering and micro wave engineering.

Module-1: Curve fitting & Statistical methods	No. of hours	BLL, CO
Examples from Engineering that require curve fitting and statistical methods. **Curve Fitting:** Curve fitting by the method of least squares-fitting the curves of the form: $y = ax+b$, $y = ax^b$ *and* $y = ax^2 + bx + c$. **Statistical methods:** Introduction to Moments, Skewness, Kurtosis and problems. Karl Pearson's coefficient of correlation and lines of regression. *Experiential Learning component: Problems on curve fitting and statistical methods*	**L: 04** **T: 04**	**Apply**
Module-2: Probability distributions & Joint probability distribution		
Examples from Engineering that require Probability and Joint probability distribution **Probability distributions:** Review of basic probability theory. Discrete and continuous Random variables, probability mass/density functions (definitions only). Binomial, Poisson, exponential and normal distributions (without proof). **Joint probability distribution:** Joint Probability distribution for two discrete random variables, expectation, covariance and correlation. *Experiential Learning component: Problems on Binomial, Poisson, Exponential and Normal distributions*	**L: 04** **T: 04**	**Apply**
Module-3:Markov chain & Sampling theory		
Examples from Engineering that require Markov Chain and Sampling Theory **Markov chain**: Introduction to Stochastic process, Probability vectors, Stochastic matrices, Regular stochastic matrices, Markov Chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states, Markovian processes. **Sampling theory**: Introduction to sampling theory, testing of hypothesis, level of significance, confidence limits, test of significance of mean and difference of means for large samples-z-test, test of significance of small Samples-Student's t- distribution, Goodness of fit-Chi-Square test. *Experiential Learning component: Problems on Markovian processes and, Sampling Theory*	**L: 04** **T: 04**	**Apply**
Module-4: Queuing theory		
Examples from Engineering that require queueing theory Introduction, birth and death process, Kendall's Notation, Symbolic representation of a queuing model, single server Poisson queuing model with infinite capacity (M/M/1: ∞/FCFS), when $\lambda_n = \lambda$ and $\mu_n = \mu(\lambda < \mu)$, Multiple server Poisson queuing model with infinite capacity (M/M/S: ∞/ FCFS), when $\lambda_n = \lambda$ for all $n, (\lambda > S\mu)$, *Experiential Learning component: Problems on* (M/M/1: ∞/FCFS) and (M/M/S: ∞/ FCFS) queuing models	**L: 04** **T: 04**	**Apply**
Module-5: Graph theory		
Examples from Engineering that require graph theory Basic concepts, types of graphs, order and size of a graph, in-degree and out-degree, bipartite-graphs, connected and disconnected graphs, Eulerian graph, Hamiltonian graphs, sub-graphs, isomorphic graphs. Matrix representation of graphs, adjacency matrix, incidence matrix.	**L: 04** **T: 04**	**Apply**

Planar graphs: definition, characterization of planar graphs, Kuratowski's theorem, Euler's formula and consequences. *Experiential Learning component: Problems on detection of planar and non-planar graphs*		

Course Outcomes: After completing the course, the students will be able to

CO 1:	Make use of correlation and regression analysis to fit a suitable mathematical model for the statistical data.
CO 2:	Apply discrete and continuous probability and joint probability distributions in analyzing the probability models arising in engineering field.
CO 3:	Use Markov chain in prediction of future events and demonstrate the validity of testing the hypothesis.
CO 4:	Acquire skills in analyzing queuing models.
CO 5:	Apply the knowledge of Graph Theory in Network modeling, electrical network and computational algorithms.

Reference Books:

1. E. Kreyszig: "Advanced Engineering Mathematics", John Wiley & Sons, 10th Edition(Reprint), 2016.
2. B. S. Grewal: "Higher Engineering Mathematics", Khanna Publishers, 44th Edition, 2017.
3. S. D. Sharma : "Operations Research", Kedar Nath Ram Nath & Co. Meerut, 2014.
4. T. Veerarajan : Probability, Statistics and Random processes, McGraw Hill Education(India) Private Limited, Third edition, Nineteenth reprint 2017.
5. C. Ray Wylie, Louis C. Barrett : "Advanced Engineering Mathematics", 6th Edition, McGraw-Hill Book Co., New York, 1995.
6. James Stewart : Calculus-Early Transcendental, Cengage Learning India Private Ltd., 2017.
7. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
8. Srimanta Pal & Subodh C. Bhunia: "Engineering Mathematics", Oxford University Press, 3rd Reprint, 2016.
9. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice Hall of India, 2000.

Web links and Video Lectures:

1. https://nptel.ac.in/courses/111104098
2. https://www.youtube.com/watch?v=1YkfeR05YXY
3. https://archive.nptel.ac.in/courses/111/104/111104079/
4. https://www.youtube.com/watch?v=xGkpXk-AnWU
5. https://archive.nptel.ac.in/courses/106/104/106104170/

B. N. M. Institute of Technology

An Autonomous Institute Under VTU

Dept. of Computer Science and Engineering

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

Semester: IV

Course Name: Microcontroller and Embedded Systems
Course Code: 23CSE142

L: T: P: J	**2:1:1:0**	**CIA Marks:** 50
Credits:	**3**	**SEA Marks:** 50
Hours/Week (Total)	**4 (40)**	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Understand the fundamentals of ARM-based systems, including programming modules with registers and the CPSR.
2	Use the various instructions to program the ARM controller.
3	Program various embedded components using the embedded C program.
4	Identify various components, their purpose, and their application to the embedded system's applicability.

Module1: Microprocessors versus Microcontrollers	**No. of Hours**	**BLL, CO**
ARM Embedded Systems: Microprocessors versus Microcontrollers, The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software. **Embedded system Development Environment:** Block diagram, Disassembler/decompiler, simulator, emulator and debugging techniques, target hardware debugging, boundary scan.	**6**	**Understand CO1**
Laboratory Component: Using Keil software, observe the various registers, dump, CPSR	**2**	
Module2: ARM Processor Fundamentals and ARM Instruction Set		
ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions **Introduction to the ARM Instruction Set:** Introduction to the ARM Instruction Set: Data Processing Instructions, Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants, Simple programming exercises.	**6**	**Apply CO2**
Laboratory Component: 1. Write a program to multiply two l6-bit binary numbers. 2. Write a program to find the sum of first I0 integer numbers. 3. Write a program to find factorial of a number. 4. Write a program to find the square of a number (1 to l0) using look up table.	**2**	
Module-3: Introduction to the Thumb Instruction Set		
Introduction to the Thumb Instruction Set: Thumb Register Usage, ARM-Thumb Interworking, Other Branch Instructions, Data Processing Instructions, Single-Register Load-Store Instructions, Multiple-Register Load-Store Instructions, Stack Instructions, Software Interrupt Instruction	**6**	

Laboratory Component: 1. Write a program to add an array of 16-bit numbers and store the 32-bitresult in internal RAM. 2. Write a program to find the largest/smallest number in an array of 16 numbers. 3. Write a program to count the number of ones and zeros in two consecutive memory locations.	**2**	**Apply CO3**

Module4: Embedded System Components		
Embedded System Components: Embedded Vs General computing system, Classification of Embedded systems, Major applications, and purpose of ES. Core of an Embedded System including all types of processor/controllers, Memory, Sensors, Actuators, LED, 7 segment LED display, Optocoupler, Relay, Piezo buzzer, Push button switch, Communication Interface (onboard and external types)	**4**	**Apply CO4**
Laboratory Component: 1. Interface and Control a DC Motor. 2. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction. 3. Interface DAC to generate triangular & square waves. 4. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between.	**4**	
Module-5: RTOS for Embedded System Design:		
RTOS for Embedded System Design: Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread preemption, Preemptive Task scheduling techniques, Task Communication, how to choose an RTOS	**8**	**Understand CO5**

Course Outcomes: After completing the course, the students will be able to	
23CSE142.1	Understand the fundamentals of ARM-based systems and Embedded system Development Environment,
23CSE142.2	Apply the knowledge to write an ALP and observe the status of registers, CPSR.
23CSE142.3	Apply the knowledge gained for Programming in ARM Thumb Instruction Set.
23CSE142.4	Apply the knowledge to interface external devices and I/O with ARM microcontroller.
23CSE142.5	Identify the importance of RTOS for Embedded Systems in real time.

Text Books

1. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers, 2008.
2. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2nd Edition.

Reference Books

1. Raghunandan. G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication,2019
2. The Insider's Guide to the ARM7 Based Microcontrollers, Hitex Ltd.,1st edition, 2005.
3. Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson, 2015.
4. Raj Kamal, Embedded System, Tata McGraw-Hill Publishers, 2nd Edition, 2008.

Marks Distribution for Assessment:

CIA	**Component**	**Description**	**Marks**
50	Test	Total Number of Test: 3 Each Theory test will be conducted for 30 Marks Average of 3 tests = 30 Marks	30
	Weekly assessment	10 Marks	10
	AAT	10 Marks	10
Total Marks			**50**
SEA	**Component**	**Description**	**Marks**
50	Theory Exam	5 Questions to answer of 20 Marks 2 Questions from each module with internal choice. Student should answer one full question from each module.	100 Reduced to 50
		Total marks for the Course	**50**

B. N. M. Institute of Technology

An Autonomous Institute Under VTU

Dept. of Computer Science & Engineering

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

Semester: IV

Course Name: Database Management System		**Course Code: 23CSE143**
L: T: P: J	3: 0 :2 :0	**CIA Marks:** 50
Credits:	4	**SEA Marks:** 50
Hours/Week (Total)	5	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Understand fundamental concepts, terminology and application of databases, SQL and NoSQL
2	Design concepts and creation of relational databases using relation algebra.
3	Practice SQL programming through a variety of database problems.
4	Demonstrate the use of Normalization, concurrency and transactions in database.

Module-1: Database System Concepts, Data Modeling	**No. of Hours**	**BLL, CO**
Databases and Databases Users: Characteristics of database Approach, Advantages of using the DBMS Approach. Database System Concepts and Architecture: Data Models-Schemas, Three-Schema Architecture and Data Independence, Database Languages, and Interfaces. Data Modeling Using the Entity-Relationship (ER) Model: Entity Types-Entity sets- Attributes and Keys, Relationship types, structural Constraints, Weak Entity Types. converting the database specification in E/R notation to the relational schema **Practical component:** Draw ER Diagram for the following Databases using GitMind software. Order Database Library Database Bank Database	10	**Understand CO1**
Module-2: Relational Data Model and Relational Algebra		
Concepts of relations, keys, referential integrity and foreign keys, relational algebra operators: selection, projection, cross product, various types of joins, division, example queries **Practical component:** Create Schema, insert at least 5 records in each table and add appropriate constraints for the following Library Database using ORACLE or MySQL DBMS under LINUX/Windows environment 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, Br_id, Card_No, Date_Out, Due_Date) LIBRARY_BRANCH (Branch_id, Branch_Name, Address) Write SQL queries to 1. Retrieve details of all books in the library – id, title, name of publisher,authors, number of copies in each branch, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books,but from Jan 2020 to Jun 2022.	**10**	**Apply CO2**

3. Delete a book in BOOK table. Update the contents of other tables toreflect this data manipulation operation.		
Module-3: SQL		
Basic SQL: SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT – DELETE and UPDATE Statements in SQL, Additional features in SQL **More SQL: Complex Queries, Triggers, Views:** Complex SQL Retrieval Queries, Specifying Constraints as Assertions and actions as Triggers, Views (Virtual Tables) in SQL. **Practical component:** Create Schema, insert at least 5 records for each table and add appropriate constraints for the following Order Database using ORACLE or MySQL DBMS under LINUX/Windows environment. SALESMAN (Salesman_id, Name, City, Commission) CUSTOMER (C_id, Cust_Name, City, Grade, Salesman_id) ORDERS (Ord_No, Purchase_Amt, Ord_Date, C_id, S_id) Write SQL queries to 1. Count the customers with grades above Bangalore's average. 2. Find the name and numbers of all salesman who had more than onecustomer. 3. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.) 4. Create a view that finds the salesman who has the customer with the highest order.	**10**	**Apply CO3**
Module-4: Functional Dependencies and Normalization		
Basics of Functional Dependencies and Normalization for Relational Database: Functional Dependencies, Armstrong's axioms for FD's, Equivalent Decompositions, closure of a set of FDs, minimal covers, Normal forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce- Codd Normal Forms [BCNF] **Practical component:** Create Schema, insert at least 5 records for each table and add appropriate constraints for the following Company Database using ORACLE or MySQL DBMS under LINUX/Windows environment. EMPLOYEE (SSN, Name, Address, Sex, Salary, Super SSN, D No) DEPARTMENT (D No, D Name, Mgr. SSN, Mgr. Start Date) DLOCATION(D No,D Loc) PROJECT (P No, P Name, P Location, D No) WORKS_ON(SSN, P No, Hours) Write SQL queries to 1. 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. 2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. **3.** Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary and the average salary in this department.	**10**	**Analyze CO4**

Module-5: Transaction Processing, Concurrency Control, NoSQL		
Introduction to Transaction Processing –Introduction to Transaction Processing, Desirable Properties on Transactions (ACID) **Concurrency Control Techniques:** Transactions and Schedules, Serializability, Precedence Graphs, Concurrency, Lock Based Protocols: 2PL, Strict 2PL Protocols, Deadlocks - Detection and Prevention **NoSQL:** SQL v/s NoSQL, The Emergence of NoSQL, BASE Properties, Data Models: Relationships, Graph Database, Schema less Database.	**10**	**Analyze CO5**

Course Outcomes: After completing the course, the students will be able to	
23CSE143.1	Understand the Database System Concepts along with Data Modeling Using the Entity-Relationship (ER) Model
23CSE143.2	Apply the concepts of relations on RDBMS, constraints, joints using relational algebra operators.
23CSE143.3	Apply Structured Query Language for database manipulation.
23CSE143.4	Analyze functional dependencies to normalize relations of relational database
23CSE143.5	Analyze transactions processing, schedules protocols, serializability issues, deadlocks in DBMS and concepts of NoSQL with its advantages

Text Books

1. Ramez Elmasari, Shamkant B Navathe "Fundamentals of Database Systems", Pearson, Seventh Edition 2017.
2. "Database System Concepts", Silberschatz, H Korth, S Sudarshan, 6th Edition, McGraw-Hill, 2010
3. Pramod J Sadalage, Martin Fowler, "NOSQL Distilled", Pearson, 2013

Marks Distribution for Assessment:

CIA (50)	**Component**	**Description**	**Marks**
	Written Test	Total Number of Test: 3 Each Theory test will be conducted for 30 marks Average of 3 tests = 30 Marks (Scaled down to 15 marks)	15
	Lab Test		15
	Weekly Assessment		10
	Assignment / AAT		10
Total Marks			**50**
SEA (50)	**Component**	**Description**	**Marks**
	Written Exam	5 Questions to answer, each of 20 marks. 2 Questions from each module with internal choice. Student should answer one full question from each module.	20*5=100 Scale down to **50**
		Total marks for the Course	**100**

B. N. M. Institute of Technology

An Autonomous Institute Under VTU

Dept. of Computer Science & Engineering

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

Semester: IV

Course Name: Design and Analysis of Algorithms **Course Code: 23CSE144**

L: T: P: J	**3 :0:2:0**	**CIA Marks:** 50
Credits:	**4**	**SEA Marks:** 50
Hours/Week (Total)	**5 (50)**	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Analyze the asymptotic performance of algorithms.
2	To understand the concept of designing an algorithm.
3	Synthesize efficient algorithms in common engineering design situations.
4	Analyze the efficiency of programs based on time complexity

Module-1:	No. of Hours	BLL, CO
Introduction: Notion of algorithm, Fundamentals of Algorithmic ProblemSolving, Fundamentals of the Analysis of Algorithmic Efficiency: Analysis frame work, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysisof Non-recursive and Recursive Algorithms. **Practical Programs** 1 Implement Java program to find Factorial of a given number. 2 Implement Java program to print Fibonacci series of a given number. 3 Implement Java program to check whether elements in an array is unique or not. 4 Implement a Java program for Tower of Hanoi problem. 5 Implement a Java program to generate list of prime numbers using Sieve of Eratosthenes.	**6 hours (Theory) 4 hours (Practical)**	**Analyze CO1**
Module-2:		
Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching **Divide and Conquer**: General method, Binary search, Recurrence equation for divide and conquer, Finding the Maximum and Minimum,Merge sort, Quick sort, Strassen's matrix multiplication. Decrease and Conquer Approach: Topological Sort.	**6 hours (Theory) 4 hours (Practical)**	**Analyze CO2**
Practical Programs: 1 Implement Java program for Linear search and find the time required to search the key element. 2 Implement Java program to sort the elements using Selection sort and find the time required to sort the elements. 3 Implement Java program to sort the elements using Bubble sort and find the time required to sort the elements. 4 Develop a Java program to sort a given set of elements using Merge sort method and find the time required to sort the elements. 5 Develop a Java program to sort a given set of elements using Quick sort method and find the time required to sort the elements. 6 Develop a Java program to search a key in a given set of elements using Binary search method and find the time required to find the key. 7 Develop a Java program to find Maximum and Minimum using divide and conquer technique and find the time required to find the elements.		

Module-3:		
Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines. Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm. Single source shortest paths: Dijkstra's Algorithm. Optimal Tree problem: Huffman Trees and Codes. **Practical Programs:** 1 Develop a Java program to find maximum profit using Knapsack technique. 2 Implement Java program for Job Sequence problem using Greedy method. 3 Implement a Java program to construct a minimum cost spanning tree using Prim's algorithm. 4 Implement a Java program to construct a minimum cost spanning tree using Kruskal's algorithm. 5 Implement a Java program to find a single source shortest path using Dijkstra's algorithm.	**6 hours (Theory) 4 hours (Practical)**	**Apply CO3**
Module-4:		
Dynamic Programming: General method with Examples, Multistage Graphs. Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem, Bellman-Ford Algorithm, Travelling Sales Person problem, Reliability design. **Practical Programs:** 1 Implement a Java program to find all-pairs shortest path using Floyd's algorithm. 2 Implement a Java program to find a transitive closure of directed graph using Warshall's algorithm. 3 Develop a Java program to implement 0/1 knapsack using Dynamic Programming. 4 Develop a Java program to find a single source shortest path using Bellman Ford algorithm. 5 Develop a Java program to implement travelling sales man problem using Dynamic Programming.	**6 hours (Theory) 4 hours (Practical)**	**Analyze CO4**
Module-5:		
General method (T2:7.1), N-Queens problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles. Programme and Bound: Assignment Problem, Travelling Sales Person problem, NP- Complete and NP-Hard problems: Basic concepts, nondeterministic algorithms, P, NP, NP-Complete and NP-Hard classes **Practical Programs:** 1 Develop a Java program to implement N-Queen problem using Backtracking technique. 2 Design and implement a Java program for Sum-Subset problem. 3 Design and implement Java program to find all Hamiltonian Cycles in a connected undirected graph (G) of n vertices.	**6 hours (Theory) 4 hours (Practical)**	**Analyze CO5**

Course Outcomes: After completing the course, the students will be able to	
23CSE144.1	Analyze the asymptotic runtime complexity of algorithms by using mathematical relations that help to identify them in specific instances.
23CSE144.2	Analyze time complexities of algorithms using brute force and divide and conquer technique.

23CSE144.3	Apply various problem solving methodologies such as greedy, decrease and conquer to solve a given problem.
23CSE144.4	Analyze the dynamic programming strategy to estimate the computational complexity of different algorithms.
23CSE144.5	Analyze Backtracking and Branch and Bound algorithm design approaches to find best possible solution.

Text Books
1. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2rd Edition, 2009. Pearson. 2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
Reference Books
1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, CliffordStein, 3rd Edition, PHI. 2. Data Structures and Algorithms using C, R.S.Salaria, 5th Edition, Khanna Publication.

Marks Distribution for Assessment:

<table>
<tr><th rowspan="2">PCI</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th colspan="2">SEA
Conduction: 100 MReduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th><th>PART A</th><th>PART B</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="2">IA Test</td><td>30</td><td>30</td><td rowspan="2">30 Marks</td><td rowspan="2">70 Marks</td></tr>
<tr><td colspan="2">Average of two tests – 30 M</td></tr>
<tr><td>Continuous Assessment</td><td colspan="2">Weekly Assessment -20 marks</td></tr>
<tr><td colspan="3">Total – 50 Marks</td><td colspan="2">Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

IA Test: 2 IA tests - each of 30 Marks	Average of 2 tests – 30 M
Practical Lab record – 10 Marks Performance – 05 Marks Viva – 05 Marks	20 Marks
	Total 50 Marks

ii) SEA : 50%

Question Paper:

Theory part	5 questions to answer each of 6 Marks questions from each module with internal choice Student should answer one full question from each module	6 M x 5 = 30 Marks
Execution part	Write up - 20 Marks Conduction - 40 Marks Viva-Voce - 10 Marks	70 Marks
Total		**100 Marks reduced to 50 M**

Note:

- No Assignment and AAT
- **Minimum 40% passing marks in all divisions**

B. N. M. Institute of Technology

An Autonomous Institute Under VTU

Dept. of Computer Science and Engineering

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

Semester: IV

Course Name: Introduction to Augmented reality and Virtual reality **Course Code: 23CSE145**

L: T: P: J	**0 : 2 : 0 : 2**	**CIA Marks:** 50
Credits:	**2**	**SEA Marks:** 50
Hours/Week (Total)	**4**	**SEA Duration:** 03 Hours

Pre-Requisites: Basic mathematics, and Computer aided design

Course Learning Objectives: The students will be able to

1	Describe how VR systems work and list the applications of VR.
2	Understand the design and implementation of the hardware that enables VR systems tobe built.
3	Understand the system of human vision and its implication on perception and rendering.
4	Explain the concepts of motion and tracking in VR systems

Module-1: Introduction	No. of Hours	BLL, CO
Defining Virtual Reality, History of VR, Human Physiology and Perception, Key Elements of Virtual Reality Experience, Virtual Reality System, Interface to the Virtual World-Input & output- Visual, Aural & Haptic Displays, Applications of Virtual Reality **Case Studies:** Study the use of Virtual Reality at NASA	**6**	**Apply CO1**
Module-2:		
Representation of the Virtual World, Visual Representation in VR, Aural Representation in VR and Haptic Representation in VR **Case Studies** GHOST (General Haptics Open Software Toolkit) software development toolkit.	**6**	**Apply CO2**
Module-3:		
Geometric Models, Changing Position and Orientation, Axis-Angle Representations of Rotation, Viewing Transformations, Chaining the Transformations, Human Eye, eye movements & implications for VR. **Case Studies:** Sweeping coverage of eye movements	**6**	**Apply CO3**
Module-4:		
Visual Perception - Perception of Depth, Perception of Motion, Perception of Color, Combining Sources of Information Visual Rendering -Ray Tracing and Shading Models, Rasterization, Correcting Optical Distortions, Improving Latency and Frame Rates **Case Studies:** Automatic stitching of panoramas in Virtual Reality	**6**	**Apply CO4**
Module-5:		
Motion in Real and Virtual Worlds- Velocities and Accelerations, The Vestibular System, Physics in the Virtual World, Mismatched Motion and Vection Tracking- Tracking 2D & 3D Orientation, Tracking Position and Orientation, Tracking Attached Bodies **Case Studies:** A virtual Study Use Case- NICE, An Educational Experience	**6**	**Apply CO4**

Laboratory Component:

1) Create a 3D object and Apply different geometric Transformations using Mouse/Keyboard
2) Create animation for a 3D object (transformation, color, texture, etc.)

3) Bouncing ball on multiple 2D/3D platforms
4) Develop First Person Controller to a Scene
5) Create a 3D Character movement
6) Create a menu driven interface for adding and removing objects from a Scene
7) Build a cubic room, whose sides are made out of six planes. The room should be 15x15x15 Unity units. At the center of the roof of the room, place a point source of light. This light should change color by pressing the Tab key.
8) Finding target using 2D Ray-caster
9)Create a loading bar (health bar, progress bar, start bar)
10)Create and show motion effect using time scale and scripts for 2D images.

Course Outcomes: After completing the course, the students will be able to	
23CSE145.1	Apply the concepts of VR systems work and list the applications of VR.
23CSE145.2	Design and implementation of the hardware that enables VR systems tobe built.
23CSE145.3	Implement the system of human vision and its implication on perception and rendering.
23CSE145.4	Apply the concepts of motion and tracking in VR systems

Text Books

1. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016
2. Understanding Virtual Reality: Interface, Application and Design, William R Sherman and Alan B Craig, (The Morgan Kaufmann Series in Computer Graphics)". Morgan Kaufmann Publishers, San Francisco, CA, 2002
3. Developing Virtual Reality Applications: Foundations of Effective Design, Alan B Craig, William R Sherman and Jeffrey D Will, Morgan Kaufmann, 2009.

Reference Books

1. Gerard Jounghyun Kim, "Designing Virtual Systems: The Structured Approach", 2005.
2. Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, "3D User Interfaces, Theory and Practice", Addison Wesley, USA, 2005.
3. Oliver Bimber and Ramesh Raskar, "Spatial Augmented Reality: Meging Real and Virtual Worlds", 2005. 4. Burdea, Grigore C and Philippe Coiffet, "Virtual Reality Technology", Wiley Interscience, India, 2003.

e-Books:
http://lavalle.pl/vr/book.html

MOOC Courses:
https://nptel.ac.in/courses/106/106/106106138/
https://www.coursera.org/learn/introduction-virtual-reality

Marks Marks Distribution for Assessment:

CIA (50)	**Component**	**Description**	**Marks**
	Written Test	• Total Number of Test: 2 • Each test will be conducted for 40 marks out of which 15 marks for theory and 25 marks for lab test. • Average of 2 tests to 40 Marks	40
	AAT	Presentation / Demonstration of mini project and weekly assessment.	10
Total Marks			**50**
SEA (50)	**Component**	**Description**	**Marks**
	Written Exam	External lab exam will be conducted for 100 marks and scaled down to 50 Marks	**50**
		Total marks for the Course	**100**

B. N. M. Institute of Technology

An Autonomous Institute Under VTU

Dept. of Computer Science and Engineering

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

Semester: IV

Course Name: Advanced IoT		**Course Code: 23CSE146**
L: T: P: J	**0:0:2:2**	**CIA Marks:** 50
Credits:	**2**	**SEA Marks:** 50
Hours/Week (Total)	**4**	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Grasp the fundamental concepts of IoT, including architecture, protocols, and applications.
2	Learn to interface various sensors and actuators with microcontrollers.
3	Understand networking protocols used in IoT applications.
4	Explore methods for data collection, storage, and processing in the cloud.
5	Design and develop a simple IoT application using relevant tools and platforms.

Module-1: Introduction to IoT and Networking Basics	**No. of Hours**	**BLL, CO**
Definition and scope of IoT, IoT architecture and components IoT applications in various domains (smart homes, healthcare, agriculture, etc.),Overview of computer networks,OSI and TCP/IP models,IP addressing and subnetting, Introduction to networking protocols (HTTP, MQTT, CoAP)	**6**	**Understand CO1**

Hands On:

- Setting up an IoT development environment (installing software like Arduino IDE, Node-RED)
- Basic IoT project: LED control using Arduino and a smartphone app
- Setting up a local network using routers and switches
- Configuring static and dynamic IP addresses
- Basic networking project: Creating a small network with Raspberry Pi

Module-2: Device Interfacing: Sensors and Actuators		
Types of sensors and actuators used in IoT Working principles and applications	**6**	**Apply CO2**

Hands On:

- Interfacing sensors (temperature, humidity, motion) with Arduino/Raspberry Pi
- Creating a simple weather station project that reads and displays sensor data

Module-3: Data Communication in IoT		
Communication models: Device-to-Device, Device-to-Cloud Introduction to wireless communication technologies (Wi-Fi, Bluetooth, Zigbee, LoRa)	**6**	**Apply CO3**

Hands On:

- Implementing data transmission between two devices using MQTT
- Setting up a simple home automation system using ESP8266 and MQTT

Module-4: Cloud Computing for IoT		
Cloud computing concepts and models (IaaS, PaaS, SaaS),IoT cloud platforms overview (AWS IoT, Google Cloud IoT, Azure IoT)	**6**	**Apply CO4**

Hands On:

- Creating a cloud account and connecting IoT devices to a cloud platform

• Building a project that sends sensor data to the cloud and visualizes it (using Grafana or similar)		
Module-5: Data Analytics and Security in IoT		
Introduction to data analytics in IoT Basic data processing techniques Security challenges and solutions in IoT	**6**	**Apply CO5**
Hands On: • Analyzing sensor data using basic statistical methods or Python libraries (Pandas, Matplotlib) • Implementing basic security measures (data encryption, secure MQTT connections)		

Course Outcomes: After completing the course, the students will be able to	
21CSE146.1	Understand the basic concepts and platforms of IoT and Networks
21CSE146.2	Experiment with Sensors and Actuators and build simple projects.
21CSE146.3	Construct Activities by using IoT communication proctocols.
21CSE146.4	Apply Data analytics for the data stored in cloud.
21CSE146.5	Build Projects which demands the concept of IoT.

Textbooks
1. "Internet of Things: A Comprehensive Approach" by Ranjeet 2. "Arduino Cookbook" by Michael Margolis
Reference Resource
1. Coursera, edX for IoT and networking courses 2. GitHub for open-source IoT projects
Tools & Technologies: 1. Hardware: Arduino, Raspberry Pi, ESP8266, various sensors and actuators 2. Software: Arduino IDE, Node-RED, MQTT brokers (Mosquitto), cloud platforms (AWS, Azure)

Marks Distribution for Assessment:

CIA	**Component**	**Description**	**Marks**
50	Test	Total Number of Test: 2 Each Theory test will be conducted for 30 Marks Average of 2 tests = 25 Marks	25
	Weekly Assignment	Lab Record/Project	10
		Lab IA Test	15
Total Marks			**50**
SEA	**Component**	**Description**	**Marks**
50	Execution Part	Write-up – 10 Marks Project Report – 25 Marks Presentation and Demonstration—50M Viva Voce – 15 Marks	100 Marks reduced to 50 Marks
		Total marks for the Course	**100**