Department of
COMPUTER SCIENCE & ENGINEERING

4 Years B.Tech Degree Programme

REGULATION & SYLLABUS 2017

Choice Based Credit System
Outcome Based Assessment

SEMESTER- III & IV

AUTONOMOUS

Accredited by NBA

Accredited by NAAC with 'A' Grade (3.28 out of 4.00 CGPA)

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

Affiliated to UGC New Delhi & Biju Patnaik University of Technology, Odisha

GUNUPUR – 765022, Odisha, India

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: To render best platform for adequate training and opportunities to work as teams on projects with effective communication skills and leadership qualities and understand professional ethics, social awareness and organizational context in which their engineering skills are utilized.

PEO2: To endow the students with sound knowledge in the field of mathematics, basic science and engineering fundamentals to solve and inculcate the ability to utilize their skills to prepare them for higher studies, research and analyze engineering problems.

PEO3: To extend an ability to analyze the need of the society by providing innovative solutions, leading to their personal cum professional growth as an entrepreneur.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: To provide students an understanding of the expectations of industry and practical competence with a broad range of programming language and open source platforms through value added courses.

PSO 2: The ability to analyze and develop computer programs in the areas related to artificial intelligence, big data analytics and cyber security for efficient design of computer-based systems of varying complexity.

PROGRAMME OUTCOMES (POs)

PO-1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO-2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3. Design / Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO- 4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO- 5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO- 6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO- 7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO- 8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO- 9. Individual and team work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11. Project management and finance: Demonstrate knowledge and understanding of the engineering management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

COURSE STRUCTURE

SEMESTER-III

Sl.	Course Category	Course Code	Course Title	L	T	P	C	QP
THEORY								
1	PC	BCSPC3010	Design & Analysis of Algorithms	3	1	-	4	A
2	PC	BCSPC3020	Operating System	3	-	-	3	A
3	PC	BCSPC3030	OOPS through JAVA	3	-	-	3	A
4	ES	BCSES3040	Digital Logic Design	3	-	-	3	A
5	BS	BBSBS3050	Discrete Mathematical Structures	3	1	-	4	A
6	HS	BMGHS3061	Engineering Economics & Costing	3	-	-	3	A
		BBSHS3062	Environmental Engineering & Safety					
PRACTICAL								
7	PC	BCSPC3110	Design & Analysis of Algorithms Lab	-	-	2	1	A
8	PC	BCSPC3120	Operating System Lab	-	-	2	1	A
9	PC	BCSPC3130	Java Programming Lab	-	-	2	1	A
10	ES	BCSES3140	Digital Logic Design Lab	-	-	2	1	A
Total				**18**	**2**	**8**	**24**	

Course Code	Course Title	L	T	P	C	QP
BCSPC3010	**DESIGN AND ANALYSIS OF ALGORITHMS**	**3**	**1**	**0**	**4**	**A**

PRE - REQUISITE:

Course Educational Objective	
CEO1: Analyze the asymptotic performance of algorithms	
CEO2: Demonstrate a familiarity with major algorithms	
CEO3: Apply important algorithmic design paradigms and methods of analysis	
CEO4: Synthesize efficient algorithms in common engineering design situations	
Course Outcome: At the end of the course, the student will be capable of	
CO1	Analyze worst-case running times of algorithms using asymptotic analysis
CO2	Apply the algorithms and design techniques to solve problems.
CO3	Apply the algorithms and design techniques to find the optimal solution.
CO4	Apply the approximation algorithm for time consuming problem.

UNIT:1 Introduction **(15 Hours)**

Definition, Chrematistics of algorithm, Growth of Functions, Asymptotic analysis, Amortized analysis, standard notations and common functions, limit theorem, String's formula. Recurrences: solution of recurrences by substitution, recursion tree and Master methods, Extension Master Methods. Algorithm design techniques.

UNIT:2 **(15 Hours)**

Divide-and- conquer Approach: Binary search, Quick sort, Merge sort, Heap Sort, Priority Queue, Lower bounds for sorting. Worst case analysis of Quick sort.

Dynamic programming methodology: Elements of dynamic programming, Matrix-chain multiplication, Longest common subsequence, Assembly-line scheduling.

Greedy Algorithms: Elements of Greedy strategy, Activity selection Problem, Fractional knapsack problem, Huffman codes.

UNIT:3 **(10 Hours)**

Graph Algorithms: Data structure for disjoint sets, Disjoint set operations, Linked list representation, path compression, Disjoint set forests. Graph Algorithms and their characteristics, Breadth first search and depth-first search, Minimum Spanning Trees, Kruskal algorithm and Prim's algorithms, single- source shortest paths (Bellman-ford Algorithm and Dijkstra's algorithms), All-pairs shortest paths (Floyd–Warshall Algorithm).

UNIT:4 **(10 Hours)**

Back tracking, Branch and Bound, Eight Queen problem, Sub Set Sum Problem. String matching algorithms, naïve string matching algorithm, Rabin-Karp algorithm, Knuth–Morris–Pratt algorithm, NP - Completeness (Polynomial time, Polynomial time verification, NP - Completeness and reducibility, NP-Complete problems (without Proofs), Approximation algorithms characteristics, Traveling Salesman Problem, vertex Cover Problem.

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, 2nd Edition, PHI Learning Pvt. Ltd.
2. Fundamentals of Algorithm, Horowitz & Sahani:, 2nd Edition, Universities Press.

Ref. Books :

1. Algorithms, Design and Analysis, H. Bhasin, First Edition, Oxford University
2. Design and Analysis of Algorithim, S. Sridhar, Oxford University Press
3. Algorithms, Sanjay Dasgupta, Umesh Vazirani , McGraw-Hill Education.

DESIGN & ANALYSIS OF ALGORITHMS LAB.

(Sub. Code: BCSPC3110)

Course Educational Objective	
CEO1: To design and implement algorithms of Brute Force Technique.	
CEO2: To design and implement algorithms with Divide and Conquer technique.	
Course Outcome:	
CO1	Design and implement algorithms of Brute Force Technique.
CO2	Design and implement algorithms with Divide and Conquer technique.
CO3	Design and implement algorithms with Space and Time Tradeoff.
CO4	Design and implement optimization algorithms using Dynamic Programming and Greedy technique.

1. Using a stack of characters, convert an infix string to postfix string (1 class)
2. Implement insertion, deletion, searching of a BST. (1 class)
3. (a) Implement binary search and linear search in a program
   (b)Implement a heap sort using a max heap.
4. (a) Implement DFS/ BFS for a connected graph.
   (b)Implement Dijkstra's shortest path algorithm using BFS.
5. (a) Write a program to implement Huffman's algorithm.
   (b) Implement MST using Kruskal /Prim algorithm.
6. (a) Write a program on Quick sort algorithm.
   (b) Write a program on merge sort algorithm.

Take different input instances for both the algorithm and show the running time.

7. Implement Strassen's matrix multiplication algorithm.
8. Write down a program to find out a solution for 0 / 1 Knapsack problem.
9. Using dynamic programming implement LCS.
10. (a) Find out the solution to the N-Queen problem.
    (b) Implement back tracking using game trees.

Course Code	Course Title	L	T	P	C	QP
BCSPC3020	**Operating System**	**3**	**0**	**0**	**3**	**A**

Pre -Requisite:

Course Educational Objective	
CEO1: Provide an insight into the basic organization of computer systems.	
CEO2: Familiarize students with various components of an Operating System	
CEO3: Focus on fundamental problems and optimal solutions for resource managements in operating systems such as process and disk scheduling and memory management	
CEO4: understand the different possible solutions for handling deadlock situation.	
CEO5: Introduce the design of a file system on secondary storage	
CEO6: Discuss system I/O in depth	
Course Outcome: At the end of the course, the student will be capable of	
CO1	Correlate the different components of an operating system and analyze techniques for system resource management.
CO2	Compare and contrast various scheduling algorithms and their performance tradeoffs.
CO3	Analyze algorithmic solutions for process synchronization problems and handling deadlocks.
CO4	Analyze performance issues associated with I/O devices and learn the algorithms and structures used for storage management.

UNIT:1 **(10 Hours)**

Operating System Introduction: Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special - Purpose Systems, Operating System services, user OS Interface, System Calls, Types of System Calls, System Programs, Opening System Design and Implementation, OS Structure, Virtual machines.

UNIT:2 **(12 Hours)**

Process and CPU Scheduling - Process concepts - The Process, Process State, Process Control Block, Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies: Linux, Windows.Process Coordination - Process Synchronization, The Critical section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT:3 **(14 Hours)**

Memory Management and Virtual Memory - Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table. Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames, Thrashing.

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

UNIT:4 **(14 Hours)**

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance.Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk

Management, Swap space Management.
Protection - System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition.
2. Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

Ref. Books :

1. Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.
2. Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhere, TMH.
3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
4. Operating Systems, A. S. Godbole, 2nd Edition, TMH
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems, S, Haldar and A. A. Arvind, Pearson Education.
7. Operating Systems, R. Elmasri, A. G. Carrick and D. Levine, Mc Graw Hill.
8. Operating Systems in depth, T. W. Doeppner, Wiley.

OPERATING SYSTEM LAB.
(Sub. Code: BCSPC3120)

Course Educational Objective

CEO1: Enable students to learn the problems in inter process communication and the possible solutions

CEO2: Imbibe students with disk scheduling concepts and techniques

Course Outcome: At the end of the course, the student will be capable of

CO1	Write programs to implement the basic functionality of an operating system and its components using UNIX Commands and shell programming.
CO2	Write programs to implement the various scheduling algorithms and analyze their performance tradeoffs
CO3	Implement algorithmic solutions to process synchronization problems
CO4	Implement algorithmic solutions to handle deadlocks

1. Basic UNIX Commands.
2. Linux Administrative commands.
3. UNIX Shell Programming.
4. Programs on process creation and synchronization, inter process communication including shared memory, pipes and messages. (Dinning Philosopher problem / Cigarette Smoker problem / Sleeping barber problem)
5. Programs on UNIX System calls.
6. Simulation of CPU Scheduling Algorithms. (FCFS, RR, SJF, Priority, Multilevel Queuing)
7. Simulation of Banker's Algorithm for Deadlock Avoidance, Prevention
8. Program for FIFO, LRU, and OPTIMAL page replacement algorithm.
9. Android Programming for mobile application.

Course Code	Course Title	L	T	P	C	QP
BCSPC3030	**OOPS Through JAVA**	**3**	**0**	**0**	**3**	**A**

Course Educational Objective

CEO1: The model of object oriented programming: abstract data types, encapsulation, inheritance and polymorphism

CEO2: Fundamental features of an object oriented language like Java: object classes and interfaces, exceptions and libraries of object collections

CEO3: How to take the statement of a business problem and from this determine suitable logic for solving the problem; then be able to proceed to code that logic as a program written in Java.

CEO4: How to test, document and prepare a professional looking package for each business project using java doc.

Course Outcome: At the end of the course, the student will be capable of

CO1	Ability to analyze ,formulate and model problems using concepts of object oriented analysis and design and implement using Java and Implement object oriented principles for reusability.
CO2	Students will be able to write programs using basic data types and strings, using loops, Array.
CO3	Analyze the problems and resolve run-time errors with Multithreading and Exception Handling techniques
CO4	Realize the power of generics and Collections Framework and Java.io package

UNIT:1 **(12 Hours)**

An introduction Object Oriented Programming, Features of Object Oriented Programming Introduction to Java. Difference between C/C++ and Java, Features of Java, First Java Program, Writing the java program, Compiling the program, JVM and its significance in executing a program?, Architecture of JVM. Understanding, Java Tokens, Datatypes, Operators, Control Structures and Arrays, Conditional Statements, Loops/ Iterators, Jumping Statements, Java Arrays, Multidimensional Arrays, Taking Input from keyboard, Command Line Arguments, Using Scanner Class, Using Buffered Reader class.

UNIT:2 **(14 Hours)**

Introduction to Classes and Objects. Constructors, static Keyword , this Keyword, Array of Objects, Access Modifiers (Public, Private, Protected, Default). Inheritance ,Types of Inheritance and Java supported Inheritance, super, Polymorphism, Method Overloading, Constructor Overloading, Method Overriding, Dynamic Method Dispatching. String Manipulations. Wrapper classes, Auto boxing and unboxing. Abstract classes, Interfaces, Multiple Inheritance Using Interfaces,

Java API Packages, User-Defined Packages, Accessing Packages, Error and Exception Handling, Types of exceptions Hierarchy of Exception classes, try, catch, finally, throw, throws, Commonly used Exceptions and their details ,User defined exception classes.

UNIT:3 (**14 Hours**)

Multithreading, Thread in Java, Thread execution prevention methods. (yield(), join(), sleep()), Concept of Synchronization, Inter Thread Communication, Basics of Deadlock, Demon Thread, Improvement in Multithreading, Inner Classes, Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class.

IO Streams (java.io package) , Byte Stream and Character Stream, Files and Random Access Files, Serialization, Collection Frame Work (java.util), Util Package interfaces, List, Set, Map.

UNIT:4 **(14 Hours)**
Applet Introduction, Life Cycle of an Applet, GUI with an Applet, Abstract Window Toolkit (AWT), Introduction to GUI, Description of Components and Containers, Component/Container hierarchy, Understanding different Components/Container classes and their constructors, Event Handling, Different mechanisms of Event Handling, Listener Interfaces, Adapter classes.

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS
Text Books: 1 Programming in Java. Second Edition. Oxford Higher Education. (Sachin Malhotra/ Saurav Choudhary) 2 Core Java For Beginners. (Rashmi Kanta Das), Vikas Publication
Ref. Books 1. JAVA Complete Reference (9th Edition) Herbalt Schelidt

JAVA PROGRAMMING LAB.
(Sub. Code: BCSPC3130)

Course Educational Objective	
CEO1: To introduce the pure object-oriented concepts through Java programming.	
CEO2: To enable a detailed insight into the Java programming concepts such as creating classes, Methods, Interfaces, Packages, Multithreaded Environment, String handling, Enumerations, Creating small Swing application.	
Course Outcome: At the end of the course, the student will be capable of	
CO1	Apply the object-oriented concepts through Java language.
CO2	Demonstrate the concepts of polymorphism and inheritance.
CO3	Write Java programs to implement error handling techniques using exception handling
CO4	Develop solution for a real problem using Java programming.

JAVA programs on:

1. Introduction, Compiling & executing a java program.
2. Data types & variables, decision control structures: if, nested if etc.
3. Loop control structures: do, while, for etc.
4. Classes and objects.
5. Data abstraction & data hiding, inheritance, polymorphism.
6. Threads, exception handlings and applet programs
7. Interfaces and inner classes, wrapper classes, generics

Subject Code	Course Title	L	T	P	C	QP
BCSES3040	**DIGITAL LOGIC DESIGN**	**3**	**1**	**0**	**4**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: To acquire the basic knowledge of digital logic levels and implements it in digital electronics.

CEO2: Prepare the students to perform the analysis and design of various digital electronic circuits.

Course Outcome: At the end of the course, the student will be capable of

CO1	Understand working of logic families and logic gates.
CO2	Recognize and study various number systems and their application in digital design.
CO3	Design and implement combinational logic circuits.
CO4	Design and analyze sequential logic circuits.
CO5	Employ PLDs to execute the given logical problem.
CO6	Establish the process of analog to digital conversion and digital to analog conversion.

UNIT: I **8 Hours**

Number Systems and Codes: Binary, Octal, Hexadecimal and Decimal Number System and their Conversion; Representation of Signed Binary and Floating Point Number; Binary Arithmetic using 1's and 2's Complements, Binary Codes - BCD Code, Gray Code, ASCII Character Code.

Boolean Algebra and Logic Gates: Axioms and Laws of Boolean Algebra; Reducing Boolean Expressions; Logic levels and Pulse Waveforms; Logic Gates; Boolean Expressions and Logic Diagrams.

UNIT: II **(9 Hours)**

Gate-level Minimization: Canonical and Standard Forms; K-maps - Two, Three and Four Variable K-maps, Don't-Care Conditions; NAND and NOR Implementation; Other Two-Level Implementations, Exclusive-OR Function.

Combinational Logic: Combinational Circuits; Analysis Procedure; Design Procedure; Adders; Subtractors; Parallel Binary Adders; Binary Adder-Subtractor; Binary Multiplier; Magnitude Comparator; Decoders; Encoders, Multiplexers; De-multiplexers.

UNIT: III **(14 Hours)**

Synchronous Sequential Logic: Sequential Circuits; Latches, Flip-Flops; Master-Slave Flip-Flop; Conversion of Flip-Flops; Analysis of Clocked Sequential Circuits; Mealy and Moore Models of Finite State Machines.

Registers and Counters: Shift Registers; Data Transmission in Shift Registers; SISO, SIPO, PISO and PIPO Shift Registers; Counters; Asynchronous Counters; Design of Asynchronous Counters; Synchronous Counters; Design of Synchronous Counters; Ring Counter.

UNIT: IV **(14 Hours)**

Memory and Programmable Logic: Introduction; Random-Access Memory; Memory Decoding; Error Detection and Correction; Read-Only Memory; Programmable Logic Array; Programmable Array Logic; Sequential Programmable Devices.

Analog-to-Digital and Digital-to-Analog Converters: Digital-to-Analog Converters - R-2R Ladder D/A Converter, Weighted Resistor D/A Converter; Analog-to-Digital Converters - Counter-type A/D Converter, Parallel Comparator A/D Converter, Dual-Slope A/D Converter, Successive-Approximation A/D Converter, A/D Converter using Voltage-to-Frequency.

IC Logic Families: Special Characteristics; RTL, DTL, TTL, ECL, IIL, MOS and CMOS Logic Circuits.

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Books:

1. Digital Design, 3rd Edition, M. Morris Mano, Pearson Education.
2. Digital Fundamentals, 5th Edition, T. L. Floyd and R. P. Jain, Pearson Education, New Delhi.
3. Fundamentals of Digital Circuits, 8th Edition, A. Anand Kumar, PHI.

Reference Books:

1. Digital Systems – Principles and Applications, 10th Edition, Ronald J. Tocci, Neal S. Widemer and Gregory L. Moss, Pearson Education.
2. A First Course in Digital System Design: An Integrated Approach, India Edition, John P. Uyemura, PWS Publishing Company, a division of Thomson Learning Inc.
3. Digital Systems – Principles and Applications, 10th Edition, Ronald J. Tocci, Neal S. Widemer and Gregory L. Moss, Pearson Education.
4. Digital Design, Robert K. Dueck, CENGAGE Learning.
5. Digital Principles and Applications, 6th Edition, Donald P. Leach, Albert Paul Malvino and Goutam Saha, Tata McGraw Hill Publishing Company Ltd., New Delhi.

DIGITAL LOGIC DESIGN LAB.
(Sub. Code: BCSES3140)

Course Educational Objective

CEO1: To Develop assembly language programs and basic concepts of the microprocessor and microcontroller

CEO2: To provide solid foundation on interfacing the external devices to the microprocessor & microcontroller according to the user requirements in order to create novel products and solutions for the real time problems

CEO3: To Familiar and Design of any type of embedded systems related to industrial and real time applications by knowing the concepts of Microprocessor and Microcontrollers.

CEO4: To assist the students with an academic environment aware of excellence guidelines and lifelong learning needed for a successful professional carrier in the field embedded systems.

1. Digital Logic Gates: Investigate logic behavior of AND, OR, NAND, NOR, EX-OR, EX-NOR, Invert and Buffer gates, use of Universal NAND Gate .
2. Gate-level minimization: Two level and multi level implementation of Boolean functions.
3. Combinational Circuits: design, assemble and test: adders and subtractors , comparators.
4. Design and Implementation of code converters, gray code to binary and BCD to seven Segment display.
5. Design and Implementation of a function using MUX/ DEMUX.
6. Design of functions using encoder, decoder.
7. Flip-Flop: assemble, test and investigate operation of SR, D & J-Kflip-flops.
8. Shift Registers: Design and investigate the operation of all types of shift registers with parallel load.
9. Counters: Design, assemble and test various ripple and synchronous counters - decimal counter, Binary counter with parallel load.
10. Design of Binary Multiplier.
11. Verilog/VHDL simulation and implementation of Experiments listed at Sl. No. 1 to 10.
12. C/C++ implementation of Experiments listed at Sl. No. 1 to 10.

Course Code	Course Title	L	T	P	C	QP
BBSBS3050	**Discrete Mathematical Structures**	**3**	**1**	**0**	**4**	**A**

Course Educational Objective

CEO1:

CEO2:

Course Outcome: At the end of the course, the student will be capable of

CO1	
CO2	
CO3	
CO4	

UNIT:1 MATHEMATICAL LOGIC & SET THEORY (15 Hours)

Propositional logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Proof methods and Strategies, Sequences and Summations, Mathematical Induction, Recursive definition and structural induction.

UNIT:2 RECURRENCE RELATION (10 Hours)

Recurrence relation, Solution to recurrence relation, Generating functions, Inclusion and exclusion, Relation and their properties, Closure of relations, Equivalence relations, Partial orderings.

UNIT:3 BOOLEAN ALGEBRA & ALGEBRAIC SYSTEMS (15 Hours)

Algebraic systems, Lattices, Distributive and Complemented Lattices, Sub-lattices, Boolean Lattices and Boolean Algebra, Boolean Functions and Boolean Expressions.

Semi groups, Monoids, Groups, Subgrorups, Cosets, Lagrange theorem, Permutation groups, isomorphism, Homomorphism, Normal subgroups

UNIT:4 - GRAPH THEORY (12 Hours)

Basic Definitions – Some Special Graphs – Matrix , Representation of Graphs ---– Paths and circuits - Eulerian and Hamiltonian Graphs – connected graphs, Planar graph, Graph coloring ,Trees - Spanning Trees - Rooted trees – Binary Trees, Minimum Spanning tree - Kruskal's algorithm , Prim's algorithm , Tree Traversal.

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. L. Liu and D. Mohaptra, "Elements of Discrete Mathematics", Third Edition, 2008, Tata McGraw Hill Education, New Delhi

Ref. Books

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Sixth Edition, 2008, Tata McGraw Hill Education , New Delhi
2. N. Deo, Graph Theory and Applications to Engineering and Computer Science, Prentice Hall of India
3. Discrete Mathematics by Schaum's Outlines(Second Edition)
4. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", Fifth Edition, 2005,Pearon Education, New Delhi

Subject Code	Course Title	L	T	P	C	QP
BMGHS3061	**Engineering Economics and Costing**	**3**	**0**	**0**	**3**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: To teach students the basic concepts of economics, managerial and decision making ability towards cost estimation of engineering projects.

Course Outcome: At the end of the course, the student will be capable of

CO1	explain the basic concepts of the economy, supply and demand and carry out elementary economic analysis
CO2	demonstrate various types of interests involved in cost estimation
CO3	explain various types of cash flow diagrams and estimate worth of a product using different methods
CO4	Demonstrate estimations using rate of return method.

UNIT:1 **(10 Hours)**

Engineering Economics- Nature, Scope, Basic problems of an economy, Micro Economics and Macro Economics. Demand- Meaning of demand, Demand function, Law of Demand and its exceptions ,Determinants of demand, Elasticity of demand & its measurement (Simple numerical problems to be solved), Supply-Meaning of supply, Law of supply and its exception, Determinants of supply, Elasticity of supply, Determination of market equilibrium (Simple numerical problems to be solved).

UNIT:2 **(8 Hours)**

Production-Production function, Laws of returns: Law of variable proportion, Law of returns to scale Cost and revenue concepts, Basic understanding of different market structures, Determination of equilibrium price under perfect competition (Simple numerical problems to be solved), Break Even Analysis-linear approach (Simple numerical problems to be solved).

UNIT:3 **(8 Hours)**

Banking -Commercial bank, Functions of commercial bank, Central bank, Functions of Central Bank. Inflation-Meaning of inflation, types, causes, measures to control inflation. National Income-Definition, Concepts of national income, Method of measuring national income. Time value of money- Interest - Simple and compound, nominal and effective rate of interest, Cash flow diagrams, Principles of economic equivalence.

UNIT:4 **(8 Hours)**

Evaluation of engineering projects-Present worth method, Future worth method, Annual worth method, Internal rate of return method, Cost benefit analysis for public projects .Depreciation- Depreciation of capital assert, Causes of depreciation, Methods of calculating depreciation (Straight line method, Declining balance method), After tax comparison of project.

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books

1. Riggs, Bedworth and Randhwa, “Engineering Economics”, McGraw Hill Education India
2. Principles of Economics, Deviga Vengedasalam; Karunagaran Madhavan, Oxford University Press.
3. Engineering Economy by William G.Sullivan, Elin M.Wicks, C. Patric Koelling, Pearson
4. R.Paneer Seelvan, “Engineering Economics”, PHI
5. Ahuja,H.L., “Principles of Micro Economics” , S.Chand & Company Ltd
6. Jhingan,M.L., “Macro Economic Theory”
7. Macro Economics by S.P.Gupta, TMH

Ref. Books 1

Subject Code	Course Title	L	T	P	C	QP
BBSHS3062	**Environmental Engineering & Safety**	**3**	**0**	**0**	**3**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: This course introduces the students to the environmental consequences of industries, developmental actions, etc., and the methods of minimizing their impact through technology and legal systems.

Course Outcome: At the end of the course, the student will be capable of

CO1	
CO2	
CO3	
CO4	

UNIT :1 **(10 Hours)**

Ecological Concepts: Biotic components, Ecosystem Process: Energy, Food Chain, Water cycle, Oxygen cycle, Nitrogen cycle etc., Environmental gradients, Tolerance levels of environment factors, Indian Environmental Law. Chemistry in Environmental Engineering: Atmospheric chemistry, Soil chemistry. Biodiversity and its conservation

UNIT:2 **(12 Hours)**

Water Treatment: water quality standards and parameters, DO and BOD of water. Water treatment processes: Pre-treatment of water, Conventional process, Advanced water treatment process.

Waste Water Treatment: pretreatment, primary and secondary treatment of waste water, Activated sludge treatment: Anaerobic digestion, Reactor configurations and methane production.

Air Pollution: Air pollution and pollutants, criteria pollutants, Acid deposition, Global climate change –greenhouse gases, non-criteria pollutants, Air pollution meteorology, Atmospheric dispersion. Industrial Air Emission and Control. Flue gas desulphurization, NOx removal, Fugitive emissions. Noise pollution- Noise standards, measurement and control.

UNIT:3 **(10 Hours)**

Solid Waste Management: Source, classification and composition of MSW, Separation, storage and transportation, Reuse and recycling, Waste Minimization Techniques. Hazardous Waste Management: Hazardous waste and their generation, Treatment: Incinerators, Inorganic waste treatment. E.I.A., Environmental auditing.

UNIT:4 **(12 Hours)**

Occupational Safety and Health Acts, Safety procedures, Type of Accidents, Chemical and Heat Burns, Prevention of Accidents involving Hazardous substances, Human error. Hazard Control Measures in steel industry, Petroleum Refinery, Pharmaceutical industry. Fire Prevention -Detection, Extinguishing Fire safety, Handling and Storage of Hazardous Materials. Personal Protective Equipments.

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Books

1. Environmental Engineering, Irwin/ McGraw Hill International Edition, 1997, G. Kiely,
2. Environmental Engineering by Prof B.K. Mohapatra, Seven Seas Publication, Cuttack
3. Environmental Engineering and Safety , Raut & Sen Scientific Publishers.
4. Industrial Safety ,Desmukh

Reference Books

1. Environmental Engineering by Arcadio P. Sincero & Gergoria A. Sincero PHI Publication
2. Hill International Edition, 2004
3. Environmental Science, Curringham & Saigo, TMH,
4. Man and Environment by Dash & Mishra
5. Industrial Safety Management and Technology, Colling. D A – Prentice Hall, New Delhi.

COURSE STRUCTURE

SEMESTER-IV

Sl.	Course Category	Course Code	Course Title	L	T	P	C	QP
THEORY								
1	PC	BCSPC4010	Fundamentals of Python Programming	3	1		4	A
2	PC	BCSPC4020	Advanced JAVA Programming	3			3	A
3	PC	BCSPC4030	Database Management System	3			3	A
4	ES	BECES4040	Microprocessors and Microcontrollers	3			3	A
5	PC	BCSPC4050	Theory of Computation	3	1		4	A
6	HS	BMGHS3061	Engineering Economics & Costing	3			3	A
		BBSHS3062	Environmental Engineering & Safety					
PRACTICAL								
7	PC	BCSPC4110	Python Programming Lab			2	1	A
8	PC	BCSPC4120	Advanced Java Programming Lab			2	1	A
9	PC	BCSPC4130	Database Management System Lab			2	1	A
10	ES	BECES4140	Microprocessors and Microcontrollers Lab			2	1	A
			Total	**18**	**02**	**08**	**24**	

Subject Code	Course Title	L	T	P	C	QP
BCSPC4010	**Fundamental of Python Programming**	**3**	**1**	**0**	**4**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: To understand the basics of programming using Python.

CEO2: To Construct and execute basic programs in Python.

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

CO1	Demonstrate the understanding and usage of core python scripting elements.
CO2	Interpret python especially the object-oriented concepts and the built-in objects of Python
CO3	Understand and apply the concepts of file and exception handling
CO4	Create practical and contemporary applications such as web applications and discrete-event simulations

UNIT:1 **(08 Hours)**

Introduction: Installation, First Python Program: Interactive Mode Programming, Script Mode Programming; Identifiers, Reserved Words, Lines and Indentation, Multi-Line Statements, Quotation & Comments; Assigning Values to Variables, Multiple Assignment.

UNIT:2 **(10 Hours)**

Standard Data Types: Numbers, Strings, Lists, Tuples, Dictionary; Data Type Conversion; Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Operators: Logical, Membership, Identity; Operators Precedence; Python Numbers & Mathematical functions.

Data Type Conversion: Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Basic Operators, Python Numbers & Mathematical functions; Python Strings.

UNIT:3 **(12 Hours)**

Python statements and Loops: if, if-else, While, for loops, break, continue, pass, Python Function; Files I/O.

Functions: Definition, call, positional and keyword parameter. Default parameters, variable number of arguments. Modules - import mechanisms. Functional programming - map, filter, reduce, max, min. lambda function - list comprehension.

UNIT:4 **(15 Hours)**

Object Oriented Programming: classes and objects - Inheritance – Polymorphism overloading; Error handling & Exceptions - try, except and raise - exception propagation

File Processing: reading and writing files

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Reference books:

1. “Learning Python”, Mark Lutz, O’Reilly Media, Inc., Fifth Edition, 2013.
2. “Introduction to Computer Science Using Python”, Charles Dierbach, Wiley Publication, Second Edition, 2012.

PYTHON PROGRAMMING LAB
(Sub. Code: BCSPC4110)

Exercise 1 - Basics

a) Running instructions in Interactive interpreter and a Python Script
b) Write a program to purposefully raise Indentation Error and Correct it

Exercise 2 - Operations

a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

Exercise - 3 Control Flow

a) Write a Program for checking whether the given number is a even number or not.
b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/10
c) Write a program using a for loop that loops over a sequence. What is sequence?
d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

Exercise 4 - Control Flow - Continued

a) Find the sum of all the primes below two million.
Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:
1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Exercise - 5 - DS

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Exercise - 6 DS - Continued

a) Write a program combine_lists that combines these lists into a dictionary.
b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

Exercise - 7 Files

a) Write a program to print each line of a file in reverse order.
b) Write a program to compute the number of characters, words and lines in a file.

Exercise - 8 Functions

a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.
Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius
If (distance between two balls centers) <= (sum of their radii) then (they are colliding)
b) Find mean, median, mode for the given set of numbers in a list.

Exercise - 9 Functions - Continued

a) Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
b) Write a function dups to find all duplicates in the list.

c) Write a function unique to find all the unique elements of a list.

Exercise - 10 - Functions - Problem Solving

a) Write a function cumulative_product to compute cumulative product of a list of numbers.

b) Write a function reverse to reverse a list. Without using the reverse function.

c) Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.

Exercise 11 - Multi-D Lists

a) Write a program that defines a matrix and prints

b) Write a program to perform addition of two square matrices

c) Write a program to perform multiplication of two square matrices

Exercise - 12 - Modules

a) Install packages requests, flask and explore them. using (pip)

b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)

c) Write a simple script that serves a simple HTTP Response and a simple HTML Page

Exercise - 13 OOP

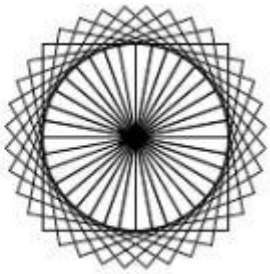
a) Class variables and instance variable

i) Robot

ii) ATM Machine

Exercise - 14 GUI, Graphics

1. Write a GUI for an Expression Calculator using tk.
2. Write a program to implement following figures using turtle



Exercise - 15 - Testing

a) Write a test-case to check the even numbers function even_numbers which return True on passing a list of all even numbers.

b) Write a test-case to check the function reverse_string which returns the reversed.

Exercise - 16 - Advanced

a) Build any one classical data structure.

b) Write a program to solve knapsack problem.

Subject Code	Course Title	L	T	P	C	QP
BCSPC4020	**Advanced Java Programming**	**3**	**0**	**0**	**3**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: Objective of this course is to provide the ability to design console based, GUI based and web based applications. Students will also be able to understand integrated development environment to create, debug and run multi-tier and enterprise-level applications.

Course Outcome: At the end of the course, the student will be capable of

CO1	Explore various programming paradigms as well as principles of building object-oriented software.
CO2	Identification of different technology and Framework for network programming and web development.
CO3	Use of JDBC, Servlet and JSP, Update and retrieve the data from the databases like Oracle, SQL Server.
CO4	Demonstrate the ability to work on larger, more complex projects by collaboratively designing and then individually implementing applications

UNIT:1 **(12 Hours)**

An introduction to Network Programming: Basics of Networking, Introduction to Socket Programming, Remote Method Invocation, Java Mail API, A small chatting pplication using Network Programming.

Introduction to Web Application and its programming: Description about Web application, Client, Server (Apache Tomcat/ WebLogic/GlassFish), An Introduction to client side programming (HTML5/CSS3/JavaScript/JQuery), An Introduction to XML/JSON.

UNIT:2 **(12 Hours)**

Basics of JDBC: Introduction to JDBC, Need of JDBC, JDBC Drivers (4 types), Architecture of JDBC, Components of JDBC (Classes and Interfaces).

Programming with JDBC: Creating a DATABASE (MS- ACCESS/ORACLE/MySQL (for Type-3 and Type-4 connection), First Program to connect to the DATABASE created, Loading the Driver, Establishing the Connection, Creating Statements (Statement/PreparedStatement/CallableStatement), Executing a SQL Query, Different types of SQL Queries, Simple Statement, Atomic Statement, Pre-Compiled Statement, SQL Statements for stored Procedures.

JDBC Program to retrieve data from DATABASE: Introduction to Result Set, Result Set with Statement Interface, Result Set with Prepared Statment Interface, Bidirectional Result Set, Result Set Scroll ability Type, Result Set Updatabiltiy Type, Updating data to the database using Result Set, Result Set Metadata, Executing Stored Procedures Using Callable Statement.

UNIT:3 **(12 Hours)**

Introduction to Servlets: What is Servlet, Advantage of Servlet Over Applets and CGI, Strengths of Servlet, Architecture of Web Application, Web Servers and its Containers, Role of servlet in Web application development, Understanding servlet-api, Understating HTTP protocol and communication between HTML-SERVLET.

Getting Deep to Servlets: Types of Servlet, Difference between HttpServlet and GenericServlet, Life cycle of Servlets and different life cycle methods, Difference between doGet() and doPost(), Servlet Generating Html output, Collecting Client submitted data in a Servlet.

Servlet communications: Servlet to DBMS communication using type-4 connection, Servlet to DBMC communication using JDBC connection pooling, Servlet communication with other servlets (Servlet Chaining), Servlet communication with JSP or HTML page (sendRedirect(), Difference between sendRedirect() and RequestDispatcher forward(), Understanding ServletConfig.

Conclusion to Servlets: Servlet Filters and wrappers, Servlet Listeners, Session Tracking, Cookies, HttpSession, HTML hidden form filed element, URL rewriting, Annotation based

servlet programs, Web Security with servlets, Servlet code for file uploading and downloading, Servlet code for mailing.

UNIT:4 **(12 Hours)**

Java Server Pages: Introduction to JSP, Scope of JSP, Anatomy of a JSP program, xecution of a JSP program, Significance of JSP Engine, Built in objects of JSP, Significance of JSP Elements, Scripting Elements, Scriptlets, Declaration, Expression, Directives and Action Elements, Page Directive, Include Directive, Taglib Directive, Forward action element, Include, Param, useBean with introduction to beans, setProperty, getProperty

Miscellaneous: Introduction to JNDI, Introduction to web services (SOAP/SOA), Rest API, An introduction to JSTL, CORBA Architecture, Facelets, JSF, AJAX Programming, Struts/Springs, Hibernates.

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books 1 Advanced Java Programming, Uttam K. Roy, Oxford University Press.

Reference Book:-

1. Black book, Kogent Learning Solution Inc.
2. Java 2: The Complete Reference by Herbert Schildt, Fifth Edition Paperback

ADVANCED JAVA PROGRAMMING LAB
(Sub. Code: BCSPC4120)

Course Outcome	
CO1	Objective of this course is to provide the ability to design console based, GUI based
CO2	Database connectivity to different rdbms Packages
CO3	The student will be able to develop distributed business applications, develop web pages using advanced server-side programming through servlets and Java server pages.
CO4	Students will also be able to understand integrated development environment to create, debug and run multi-tier and enterprise-level applications

Syllabus

- Write a program to prompt the user for a hostname and then looks up the IP address for the hostname and displays the results.
- Write a program to read the webpage from a website and display the contents of the webpage.
- Write programs for TCP server and Client interaction as per given below.
  i. A program to create TCP server to send a message to client.
  ii. A program to create TCP client to receive the message sent by the server.
- Write programs for Datagram server and Client interaction as per given below.
  i. A program to create Datagram server to send a message to client.
  ii. A program to create Datagram client to receive the message sent by the server.
- Write a program by using JDBC to execute a SQL query for a database and display the results.
- Write a program by using JDBC to execute an update query without using PreparedStatement and display the
  results.
- Write a program by using JDBC to execute an update query by using PreparedStatement and display the results.
- Write a program to execute a stored procedure in the database by using CallableStatement and display the results.
- Write a program to display a greeting message in the browser by using HttpServlet.
- Write a program to receive two numbers from a HTML form and display their sum in the browser by using HttpServlet.
- Write a program to display a list of five websites in a HTML form and visit to the selected website by using Response redirection.
- Write a program to store the user information into Cookies. Write another program to display the above stored information by retrieving from Cookies.
- Write a program in Java Beans to add a Button to the Bean and display the number of times the button has been clicked.
- Write a program for Java Bean with Simple property by using SimpleBeanInfo class.
- Write a program for Java Bean with Indexed Property by using SimpleBeanInfo class.
- Write a program to develop a Enterprise Java Bean of “Session Bean” type.
- Write a program to develop a Enterprise Java Bean of “Entity Session Bean” type.
- Write a program to develop a Enterprise Java Bean of “Message Driven Bean” type

Subject Code	Course Title	L	T	P	C	QP
BCSPC4030	**Database Management System**	**3**	**0**	**0**	**3**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: Gain a good understanding of the architecture and functioning of Database Management Systems as well as associated tools and techniques.

CEO2: Understand and apply the principles of data modeling using Entity Relationship and develop a good database design.

CEO3: Understand the use of Structured Query Language (SQL) and its syntax

CEO4: Apply Normalization techniques to normalize a database.

CEO5: Understand the need of Database processing and learn techniques for controlling the consequences of concurrent data access.

Course Outcome: At the end of the course, the student will be capable of

CO1	***Identify*** and ***Classify*** the concepts of Database Management system, Data models and architecture of database, ER to Relational mapping concepts.
CO2	***Applying*** the constraints in database using different query languages like:- relational algebra and calculus, SQL and QBE for the ***implementing*** the Data definition and data manipulate languages in Database.
CO3	***Compare*** the different normal forms to ***Apply*** normalization process to construct the consistent Database.
CO4	***Design and Develop*** the Database by inspecting concurrency control and recovery strategies to make complete Database without confliction and anomalies in concurrent access environment.

UNIT:1 **(15 Hours)**

Introduction to database Systems, advantages of database system over traditional file system, Basic concepts & Definitions, Database users, Database Language, Database System Architecture, Schemas, Sub Schemas, & Instances, database constraints, 3-level database architecture, Data Abstraction, Data Independence, Mappings, Structure, Components & functions of DBMS, Data models.

UNIT:2 **(13 Hours)**

Entity relationship model, Components of ER model, Mapping E-R model to Relational schema, Relational Algebra, Tuple & Domain Relational Calculus, Relational Query Languages: SQL and QBE. Database Design :-Database development life cycle (DDLC), Automated design tools, Functional dependency and Decomposition, Join strategies, Dependency Preservation & lossless Design, Normalization, Normal forms:1NF, 2NF,3NF, and BCNF, Multi-valued Dependencies, 4NF & 5NF. Query processing and optimization: Evaluation of Relational Algebra Expressions, Query optimization, Query cost estimation.

UNIT:3 **(10 Hours)**

Network and Object Oriented Data models, Storage Strategies: Detailed Storage Architecture, Storing Data, Magnetic Disk, RAID, Other Disks, Magnetic Tape, Storage Access, File & Record Organization, File Organizations & Indexes, Order Indices, B+ Tree Index Files, Hashing Data Dictionary.

UNIT:4 **(12 Hours)**

Transaction processing and concurrency control: Transaction concepts, properties of transaction, concurrency control, locking and Timestamp methods for concurrency control schemes. Database Recovery System, Types of Data Base failure & Types of Database Recovery, Recovery techniques. fundamental concepts on Object-Oriented Database, Object relational database, distributed database, Parallel Database, introduction to Data warehousing & Data Mining.

Teaching Methods: Chalk& Board/ PPT/Video Lectures

Text Books:

1 Sudarshan, Korth: **Database System Concepts**, 6th edition, McGraw-Hill Education
2. Elmasari &Navathe: **Fundamentals of Database System**, Pearson Education.

References Books:

1. Elmasari &Navathe: **Fundamentals of Database System**, Pearson Education.
2. Ramakrishnan: **Database Management Systems**, McGraw-Hill Education.
3. Andrew S. Tanenbaum: **Modern Operating Systems**, 3rd Edition, Pearson Education.
4. Terry Dawson, Olaf Kirch: **Linux Network Administrator's Guide**, 3rd Edition O'Reilly Media

DATABASE MANAGEMENT SYSTEM LAB.
(Sub. Code: BCSPC4130)

Course Educational Objective	
CEO1: Design and create a ERD (Entity Relationship Diagram) using software tool.	
CEO2: Learn how to design and create and use a relational database system.	
Course Outcome	
CO1	***Implement*** the concept of Entity-Relationship (E-R) model from specified information and to transform into to relational model.
CO2	***Apply*** the different types of Constraints in relational database and defines the database.
CO3	***Compares*** the different types of manipulation and access methods of data from database.
CO4	***Analyze*** and simple database application that demonstrates understanding of all the above, working as a team.

1. Use of SQL syntax: insertion, deletion, join, updation using SQL.
2. Programs on join statements and SQL queries including where clause.
3. Programs on procedures and functions.
4. Programs on database triggers.
5. Programs on packages.
6. Programs on data recovery using check point technique.
7. Concurrency control problem using lock operations.
8. Programs on ODBC using either VC++.
9. Programs on JDBC.
10. Programs on embedded SQL using C / C++ as host language.

Additional Assignments

1. Use of NoSQL database like MongoDB.
2. Programs on connectivity to MongoDB using MEAN.
3. Programs on connectivity to Mongo-DB using Python.
4. Programs on connectivity to MongoDB using PHP.

Subject Code	Course Title	L	T	P	C	QP
BECES4040	**Microprocessors and Microcontrollers**	**3**	**0**	**0**	**3**	**A**

Pre -Requisite:

Course Educational Objective

CEO1: To Develop assembly language programs and basic concepts of the microprocessor and microcontroller

CEO2: To provide solid foundation on interfacing the external devices to the microprocessor & microcontroller according to the user requirements in order to create novel products and solutions for the real time problems

CEO3: To Familiar and Design of any type of embedded systems related to industrial and real time applications by knowing the concepts of Microprocessor and Microcontrollers.

CEO4: To assist the students with an academic environment aware of excellence guidelines and lifelong learning needed for a successful professional carrier in the field embedded systems.

Course Outcome: At the end of the course, the student will be capable of

CO1	Understand the architecture of 8085 ,8086 &8051
CO2	Impart the knowledge about the instruction set
CO3	To use the basic idea about the data transfer schemes and its applications
CO4	Develop skill in simple program writing for 8085 , 8086 and 8051 applications
CO5	To design circuits for various applications using microcontrollers.
CO6	To introduce the need & use of Interrupt structure 8085, 8086 & 8051.

UNIT:I **(10 Hours)**

Introduction to 8 bit Microprocessors

Introduction to 8085 microprocessor,Architecture,Signal Descriptions,Buses-Address bus,data bus and control Bus ,Instruction format ,Instruction sets ,addressing Modes,Assembly Language Programming,Timing diagram,stack and sub routine,Data Transfer Schemes,Memory Interfacing and 8085 interrupts.

UNIT:2 **(12 Hours)**

Advanced Microprocessor

Introduction to 8086 microprocessor, 8086 Architecture, Register Organization, signal descriptions, Memory Segmentation. Physical memory organization. Addressing Modes , instruction Set .Minimum and Maximum mode operation, Bus Cycle of minimum mode and maximum mode. Interrupts of 8086, Memory interfacing & Assembly Language Program.

UNIT:3 **(10 Hours)**

Peripheral Devices

Programmable Peripheral Interface (8255),Programmable Interval Timer (8254)Programmable Interrupt Controller (8259A) - Programmable DMA Controller(8257),Programmable Communication Interface (8251A) – Programmable Keyboard and Display Controller (8279).

UNIT:4 **(12 Hours)**

8051 Microcontroller

Overview of 8051 microcontroller. Architecture. I/O Ports. Memory organization, Addressing modes, data transfer instructions, Logical instructions, Arithmetic instructions, Branching (Jump & Call) instructions, Bit addressable instructions and special instructions, Interrupts and interrupt handler sub routines (Interrupt Service Routines).Assembly language program.

Teaching Methods: Chalk& Board/ PPT/Video Lectures/Lecture by Industry Expert/MOOCS

Text Book(s)

1 Microprocessor Architecture, Programming and application with 8085,R.S. Gaonkar, PRI Penram International publishing PVT. Ltd., 5thEdition

2. Advanced Microprocessors and Peripherals - A. K. Ray and K.M. Bhurchandani, TMH, 2nd edition 2006.

3. The 8051 Microcontroller and Embedded Systems, Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D.M C Kinlay, Pearson Education, Second Edition, 2008.

Reference Book(s)

1. Microprocessors and Microcontrollers Architecture, programming and system design using 8085, 8086, 8051 and 8096, Krishna Kant, PHI Publication, 2007.
2. Microprocessors and Interfacing, Programming and Hardware,Douglas V Hall, TMH Publication, 2006.
3. Microprocessors and Interfacing, N. Senthil Kumar, M. Saravanan, S. Jeevananthan and S.K. Shah, Oxford University Press.